Factory Automation

Factory Automation

Edited by

Javier Silvestre-Blanes

In-Tech
intechweb.org

Published by In-Teh

In-Teh
Olajnica 19/2, 32000 Vukovar, Croatia

www.intechweb.org
Additional copies can be obtained from:
publication@intechweb.org

First published March 2010
Printed in India

Technical Editor: Sonja Mujacic
Cover designed by Dino Smrekar

Factory Automation,
Edited by Javier Silvestre-Blanes

p. cm.
ISBN 978-953-307-024-7

Preface

Factory automation has evolved significantly in the last few decades, and is today, a complex, interdisciplinary scientific area. In this book a selection of papers on themes related to factory automation are presented, covering a broad spectrum so that the reader may become familiar with the various fields, and also study more deeply where required.

In the various chapters in this book, special attention is given to distributed applications and their use of networks, since it is one of the most relevant subjects in the evolution of factory automation. Different Medium Access Control and networks are analyzed, while Ethernet and Wireless networks are looked at in more detail, since they are among the hottest topics in recent research. Another important subject is everything concerned with the increase in the complexity of factory automation, and the need for flexibility and interoperability. Finally the use of multi-agent systems, advanced control, formal methods, or the application in this field of RFID, are more examples of the ideas and disciplines that experts around the world have analyzed in their work.

Finally, the editors would like to thank the authors for their effort and support in the publication of this book.

Javier Silvestre-Blanes

Departamento Informática de Sistemas y computadores (DISCA)

Universidad Politécnica de Valencia (UPV)

jsilves@disca.upv.es

Contents

1

Engineering Processes for Decentralized Factory Automation Systems

Thomas Wagner[1], Carolin Haußner[1], Jürgen Elger[1],
Ulrich Löwen[1] and Arndt Lüder[2]
[1]*Siemens AG, Corporate Technology, Department of Systems Engineering, Germany*
[2]*Otto-von-Guericke-University Magdeburg, Institute for Ergonomics, Manufacturing Systems, and Automation, Center for Distributed Systems, Germany*

1. Introduction

The necessity to improve the current automation concepts for cost reduction in factory automation represents a widely discussed problem. Research has developed solutions for this problem area for quite some time based on decentralized automation concepts (e.g., holonic systems, agent systems). For an overview of agent-oriented systems, please refer to (Jennings, 2000; Weiß, 2002); for agent-oriented or holonic automation systems, see (Parunak 1998; Shen et. al, 2006; Barata, 2001; Wagner et. al., 2003).
However, industrial companies have shown reluctance concerning a broad application in practice. The reasons reside mainly in lacking engineering methods for systematic implementation in industrial businesses (Hall et. al., 2005) and lacking reliable evaluation of the consequences for application domains over the entire life cycle (Lu & Jafari, 2007).
The main goal for the development of decentralized automation concepts and systems is to achieve flexibility in factory automation systems, driven by ever more rapidly changing production conditions, such as order variations, changing products, load variations, or plug&produce capabilities of machines (ElMaraghy, 2005; Wagner & Goehner, 2006). Hence, the value promised by decentralized automation concepts mainly resides in the improvement of operative parameters of a plant, for instance, through more flexibility of usage, higher efficiency, or availability. Such parameters were evaluated based on prototypes, as well as by means of simulation, and were verified with a relatively good validity (Sundermeyer & Bussmann, 2001; Thramboulidis, 2008).
However, the costs for introducing and applying decentralized automation concepts and systems – in terms of total cost of ownership (TCO) – have not been properly investigated yet, neither in science nor in industrial application. The reason is that we know much about the operation phase behaviour of production facilities using decentralized automation systems, but there exists hardly any explanation of the impact, in terms of benefits and risks, on the entire industrial life cycle (Habib 2007). In addition to operating costs, the second most important aspect of TCO lies in the activities and processes for engineering (design, realization, and commissioning) of production facilities. The related cost potential is certainly significant. In 2005, automotive manufacturers identified the portion of

the cost for engineering tasks in automotive manufacturing to be 20 to 25% of the total investment for the production line and 55% for the control system – with an upward trend (Alonso Garcia & Drath, 2007). Hence, the requirements for engineering of such facilities must be considered (Achatz & Loewen, 2005). The main objective here is to identify the impact of automation concepts on engineering activities, to create suitable and effective engineering methods for decentralized automation, and to render them applicable for industrial applications.

Based on our industrial project experience with production lines in the automotive industry and with intralogistics systems such as automatic warehouses, and our experience in decentralized automation systems from two research projects PABADIS'PROMISE (PABADIS'PROMISE, 2008) and "Internet of Things" (Internet of Things, 2009), we derived engineering methods and processes for two use cases:

- Engineering of decentralized production line control and
- Engineering of decentralized material flow control

For these two use cases, we will also elaborate the consequences for the engineering contributors, like suppliers or system integrators, as well as the evolving benefits and risks. For this purpose, we apply a systematic evaluation methodology that we have developed in earlier projects (Wagner et. al., 2008). The evaluation is performed through qualitative comparison of the different engineering processes. Based on the evaluation results, we will also present an iterative concept for a stepwise introduction of and migration to decentralized automation systems in industrial applications without disruptive technology changes.

2. Basic Approach of (Intelligent) Decentralized Automation

The factory automation is structured hierarchically in form of an "automation pyramid," in which each level performs special automation functions. Individual automation devices/systems, such as field devices, PLC, HMI, and control systems, are specifically attributed to individual levels (Table 1). Data generated in industrial production are strictly attributed to the different levels, transferred via interfaces, and aggregated or abstracted.

Level acc. IEC62264	Label	Typical Systems	Tasks (excerpt)
4	Enterprise level	Enterprise Resource Planning Systems	• Rough production planning • Order processing, logistics
3	Production level	Manufacturing Execution, or Warehouse Management Systems	• Order management: detailed planning, control, and monitoring of production or material flow • Management of plant information, reports, alarms • Management of resources, personnel, and quality • Maintenance and material inventory management
2	Process control level	Process control, HMI, and SCADA systems	• Superordinate process control and monitoring (for continuous, batch, or discrete processes) • Operation, monitoring, measurement archiving • Recipe administration and implementation
1	Automation level	PLCs, motion controllers	• Control, Monitoring and diagnostics for the equipment and machines to be automated
	Field level	I/O-modules, field devices, field bus	• Collection and processing of information from the technological process through sensors • Active process intervention through actuators

Table 1. Automation Levels and Systems

Initially, approaches for the decentralization of systems within factory automation occurred primarily at the hardware level based on PLC and field bus systems with process oriented automation functions (e.g., systems for decentralized periphery such as SIMATIC ET 200, or for DCS, such as PCS7). An essential objective here was the reduction of cabling and thus hardware cost. The consistent further development of these concepts led to intelligent field devices and the distribution of process oriented process control functions. Such decentralization, however, applied mainly to Level 1 Systems with **process-oriented** functions. In the meantime, numerous research projects develop decentralized concepts that include also the higher levels of the classic automation pyramid, such as production control systems and material flow computers reaching level 2 & 3 systems. The basic idea of these approaches is to modularize "central" process control tasks and to decentralize them to basic control unit types:

- Independent functional automation units representing and controlling the resources of a factory and
- Independent production process units representing and controlling the processed objects within the factory (e.g., a manufactured workpieces, products or transported goods)

Decentralized process control is achieved by situation-based interaction of controlling units of resources, as well as those of processed objects, resulting in a dynamic online setting for the process. For implementation of such approaches, mainly agent-systems are applied.

3. Use Cases of Decentralized Automation Systems in Factory Automation

There are various reasons for research and development of decentralized automation technologies. However, decentralized automation systems do not have an advantage per se, but must rather be regarded as means to achieve certain benefits and objectives. Examples for the benefits are:

- Increase of the flexibility and improvement of the adaptability of the automation system,
- Higher efficiency (throughput, use of resources, degree of automation),
- Reduction of communication effort,
- Increased robustness and reliability by local troubleshooting,
- Reduction of complexity for integration of resources/machines, and
- Faster setup and reconfiguration.

The following two applications of decentralized automation were developed in the framework of the research projects PABADIS'PROMISE and Internet of Things. They are representative for approaches in decentralized factory automation and can be found in a similar manner in other approaches (Shen et. al, 2006; Bussmann, 1998; Tönshoff & Woelk, 2001; Hall, 2005).

3.1 Flexible, Decentralized Production Systems

Accelerated innovation, shorter product life cycles, and more variants combined with small batch sizes represent new tasks for production automation. In the future, production plant suppliers must provide the following functionalities in addition to the actual automation:

- Production flexibility through individual production planning, and optimization,
- Vertical integration of information and functions at control, MES, and ERP levels, and
- Fast reconfiguration and scalable capacity through dynamic resource integration ("Plug'n'Produce").

In the European research project PABADIS'PROMISE, an architectural basis for flexible and vertically seamless production automation was developed. Key features of the approach are:

- Product driven definition of control processes instead of machine centered definition,
- Merging of the previously hierarchically separated tasks of control and MES systems,
- Decentralization of the automation functions into independent, cooperating units.

In this approach, the product to be manufactured represents the set point, while the control system is based on an explicit description of product data and production processes (machine independent working processes and parameters). Machine interfaces are described accordingly in a behavior-oriented manner. The integrated description model serving as basis represents an expansion of the IEC/ISO 62264 Standard (Diep et. al., 2007). Based on these descriptions, the product-specific, on-the-fly configuration of the specific production processes of the production system and the dynamic integration of additional modules at runtime are implemented by means of software agents and RFID technology. Machines and production orders are represented by a logically and physically decentralized agent network (see Fig. 1), which coordinates the production process dynamically according to machine functions, degree of utilization, and error status, as well as allows the seamless online access from the ERP level to the field level (Lüder, et al., 2007). Fig. 1 depicts an example of decentralized control of an automotive production line where the approach was prototyped.

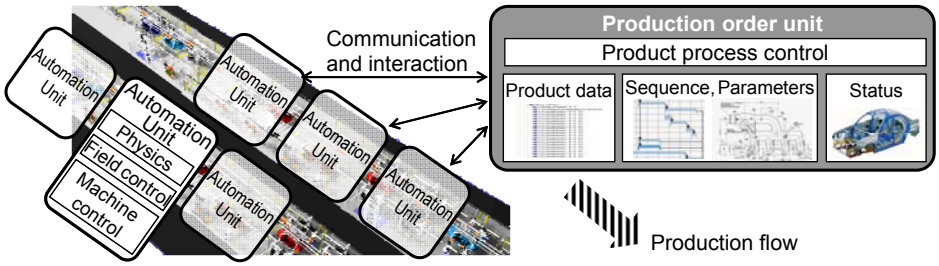


Fig. 1. Decentralized automotive production line acc. to the PABADIS'PROMISE approach

3.2 Flexible, Decentralized Intralogistics Systems

The term intralogistics refers to the logistical flow of goods and data "inside the four walls" of a plant. A typical example of an intralogistics system is an automatic high bay warehouse for storage and retrieval of goods in combination with an automated transport system that moves these goods within the plant. Currently, such systems are designed to fulfill a specialized and more or less static workflow. In the future, there will be a stronger demand to support or quickly adapt to different workflows to increase flexibility and adaptability both in the warehouse and in the transport area.

A practical application example is the increasing third-party business in the intralogistics area in which logistics companies perform storage and shipping services for one or more

customers. Because of the mostly short contract relationship, the necessity results to adapt such a facility quickly and in an optimized manner to new customers and their requirements. Considering contemporary facility designs, this would result in frequent reconstructions. Due to high efforts, such a reconstruction is often only done in a limited manner, for which reason a part of the benefit potential of the facility will not be achieved. By means of a cost effective provision of a higher flexibility through a convertible facility, third-party logistics suppliers may achieve a significant cost advantage.
Requirements on increased flexibility and adaptability of automation solutions are for example providing additional storage capacity, supporting the storage and transport of new types of boxes, introducing new storage and picking strategies, including new sources and destinations in the transportation system. These adaptions must be supported effectively by a more efficient engineering of the related disciplines, such as mechanical, field control, and process control engineering. There is the notion that higher flexibility can be achieved through an efficient modularization of the intralogistics system, along with suitable engineering processes. The "Internet of Things" research project addresses these demands by developing a new modularization concept.
Up to now, modularization is aiming mostly at the creation of modules *within* each of the abovementioned disciplines, i.e., modularization is done horizontally as shown in Fig. 2 (left), where every discipline creates its "optimum" modules and standards that help to improve the level of integration between those discipline-specific modules. The problem here is that the integration *between* the systems of the different disciplines – and, therefore, the integration of the entire facility – is not addressed through such horizontal approach and, therefore, must be implemented specifically for each project. This fact leads, according to experience, to substantial additional costs, particularly during the commissioning phase.

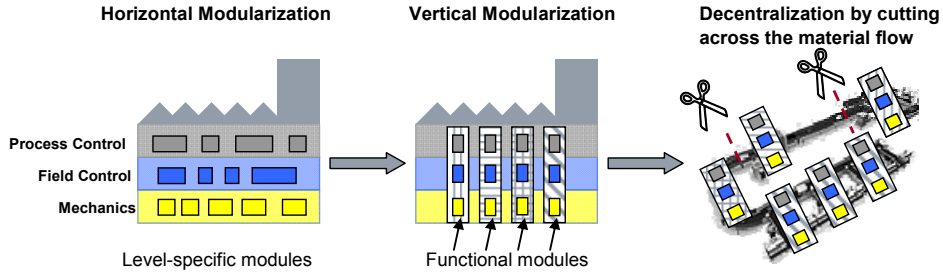


Fig. 2. Modularization and decentralization

The "Internet of Things" project introduces the concept of mechatronic modularization, stating that modularization is oriented towards the physical structure of intralogistics systems, and it is designed vertically with each module containing parts of the mechanical, field control, and process control engineering disciplines; see Fig. 2 (mid). This modularization concept is applied in a consistent and uniform manner to all engineering tasks that have to be performed during the lifecycle of an intralogistics system, in the design phase as well as in the commissioning and the reconfiguration phases.
An example of such a mechatronic module within the transport system could be a divert module that sends a box either straight on or causes it to turn left or right. The mechanical

part of the module is the divert itself. The field control part comprises, among others, the control logic required for switching the divert to straight or left/right along with the sensors and actuators. The process control level contains routing capabilities that analyze the transport destination of the box and decide if it should move straight on or not, see Fig. 3.

Fig. 3. Divert in baggage handling system

In contemporary intralogistics systems, the functionality of the process control level is implemented in a Material Flow Control (MFC) System. Once this functionality is also integrated into the concept of the mechatronic modularization, it means the dissection of the previously centralized MFC System into autonomous modules. This procedure is necessary since the modules must be easily combinable with each other and, therefore, must not have interdependences with each other. Existing, possibly complex interdependences should remain encapsulated within the modules and not appear outside of the modules (cohesion). The total functionality is dissected into independent partial functionalities, which are integrated again at runtime. For this purpose, the automation functions of the system must be dissected at all levels; see Fig. 2 (right), which leads to the decentralization of the system as shown in (Elger, 2007).
By doing this, the automation pyramid in the classical sense is largely dissolved. Decisions previously taken centrally are now taken at runtime by interaction between the individual modules. The concept of the Internet of Things research project (Internet of Things, 2009) is that modules provide mechanisms for two-way identification, for coordination among each other, and with the environs, for determination of decisions, or additionally for the dynamic adaptation to changed conditions. An example is a box to be conveyed in the intralogistics system that communicates its identification at each divert and identifies its target location by RFID to the divert module. The divert module now communicates with other material handling modules in the intralogistics system, such as other diverts, conveyors, and merges, in order to determine which path the box must take to reach the target location.
The use of pre-integrated modules, the encapsulation of the functionalities, as well as the ability of the modules for identification and coordination among each other, lead to the fact, that modules can, referring to automation engineering, be integrated with each other to an entire facility on a “Plug&Play” basis. This allows supporting the mentioned requirements for flexibility regarding construction and reconfiguration of facilities.

4. Essential Aspects in the Life Cycle of Factory Automation Systems

For a comprehensive analysis of a decentralized automation approach, such as the one mentioned above, we need to map it into the context where it is applied – the industrial life cycle. We especially focused our work on the design, realization, installation, and commissioning phases for plants. In the following, the essential aspects of these phases are explained.

Factory automation systems – like all other industrial facilities – are complex mechatronic systems according to (VDI 2206), consisting of the synergetic integration of different technical components and partial systems, which together fulfill a specific function. By doing so, the number of identical systems is substantially reduced for larger, more complex solutions; while devices and their components are mostly sold as ready products, factories are generally a one of a kind installation constructed under contract based on specific customer requirements. For the installation of a factory, specialists of many technical disciplines participate, such as mechanical engineering, process engineering, energy and automation engineering, informatics, etc., whose extensive knowledge and activities have to be integrated in the project.

Within the context of our work, the term *Engineering* is understood to represent the entire technical working process within a plant project, starting with the concept, and including Detailed Design, Realization, Installation, and Commissioning up to the transfer of the factory Fig. 4). This process contains, particularly, the *selection, design, and integration* of the components or partial systems of the subsections to an integrated mechatronic system. The objectives of the Engineering are (**1)** to secure an error-free interaction of the integrated components relating to the overall function, and (**2)** the implementation in a predictable and efficient project development process.

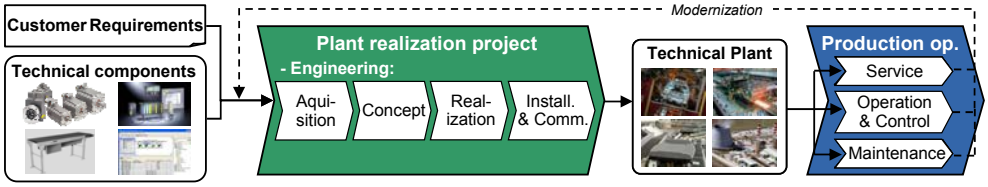


Fig. 4. Engineering of plants – integration of individual components to an entire plant

The factory and solution business encompasses the business and technical cooperation of many participants through an often fragmented value-added chain in connection with a high share of purchased components and services that must all be integrated in the context of a project. Table 2 shows schematically the most important stakeholders in the industrial life cycle, their particular deliverables/assets, and their different business goals. The effects of an innovative automation concept must be considered separately for each player.

It is important to notice that decentralized automation does not only affect the technology, but also changes the respective planning processes and models. The value-adding parameters in the engineering phases cannot be found in the technology itself, but in their application in plant design and implementation. The most important challenges in plant engineering are (Löwen & Wagner, 2009):

- More efficient development and faster time-to-operate by efficient work share, interdisciplinary cooperation, and continuity in the life cycle,
- Reduction of project risk and safeguarding of the project through systematic engineering and continuous validation throughout the project, as well as through cooperation and exchange of information in the supply chain, and
- Effect of quantity and increase of quality through repetitive use of modular solutions

Consequently, we identified the need to develop appropriate engineering methods and processes for decentralized factory automation systems and to show how they address the challenges listed above. This is the subject of the following chapters.

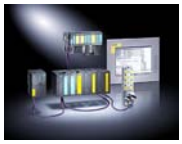
Stakeholder	Component Supplier	Equipment/ Machine Supplier	System Integrator	Plant Operator/ Owner
Description	Provides automation technology, devices, & components	Builds machines /plant equipment by using components	Builds plants, integrates equipment and components	Designs products, defines production requirements & workflows
Deliverables/ Assets	Sensors, drives, PLC, panels, control and MES systems	Robots, CNC or assembly machines, conveyor systems	Technical plant & automation solution	Product model, production process, rough material flow
Example Depiction of Deliverables /Assets				
Business Goals	Value added by flexible, integrable components and systems	Value added by integration of components to marketable machines	Value added by specific integration of systems, equipment & components	Flexible, optimized production process, availability and capacity utilization
Scope	Make-to-Stock, large numbers → Wide application scope	Make-to-Stock/ Adapt-to-Order → Domain scope	Make-to-Order → Project-oriented	Production scope (ERP to production execution level)

Table 2. Stakeholders in the Industrial Life Cycle

5. Engineering of Decentralized Factory Automation Systems

5.1 Engineering of a Production Line

When working on decentralized solutions for automated factory systems, the question was how to plan, design and implement such decentralized automation solutions. For the solution of this problem, an engineering model was developed that describes the possible processes for the engineering of decentralized production automation systems under consideration of mechatronic modules. In order to provide a reference, existing engineering

projects were analyzed first and a generalized engineering process for factory automation was documented (Fig. 5) with basic tasks described in Table 3.

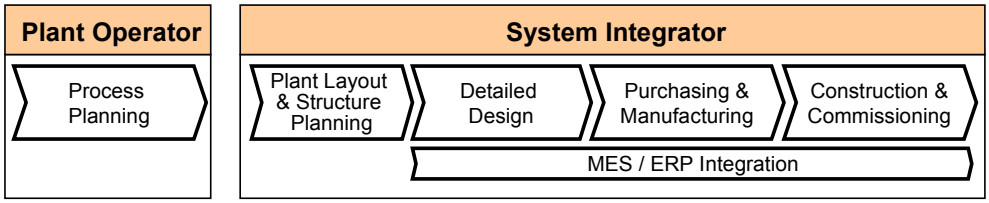


Fig. 5. Generalized engineering process for factory automation

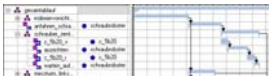
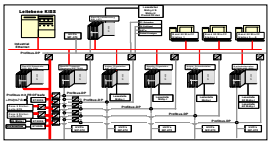
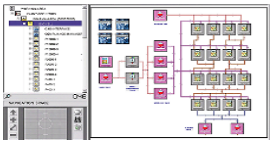
Phase	Basic Engineering Tasks	Example Results
Process Planning	▪ Define manufacturing steps and their physical and logical order	
Plant Layout & Structure Planning	▪ Derive plant layout from the process description ▪ Specify machine types by means of required manufacturing capabilities ▪ Specify material transport	
Detailed Design	▪ Specify technical details of plant equipment, instrumentation, and automation (parallel proceeding of all involved technical disciplines) ▪ Automation engineering: extend layout by control devices, detail production steps, implement control	
Purchasing & Manufacturing	▪ Purchase /manufacture devices, equipment, and machines ▪ Construct and implement toward functional units (all involved technical disciplines)	
Construction & Commissioning	▪ Integrate functional units toward complete plant ▪ Verify and optimize machine and process capability	
MES/ERP Integration	▪ MES: specify the interfaces between control system and MES system; configure MES system based on the layout and the manufacturing process ▪ ERP: specify the data for exchange; implement the necessary interfaces and communication	

Table 3. Phases and Tasks of the Basic Process for Manufacturing System Engineering

5.2 Engineering of a Decentralized Production Line

For the case of the development of a facility with decentralized automation, the previously connected automation functionality, in the course of engineering, must be distributed to the individual technical resources (machines, etc.). This corresponds to the principle of vertical (mechatronic) modules as described in Chapter 3.2. For an efficient design of the

engineering process, these modules must integrate all system specific parts of a facility resource, as well as must be used consistently. In order to avoid the effort for the specific dissection of the facility into individual mechatronic modules for each individual project, these modules must be specified in advance and provided as comprehensively as possible. The project-specific engineering process based on these principles is presented as an overview in Table 4 and explained thereafter.

Plant Operator	System Integrator		
Process Planning	Plant Layout and Structure Planning	Detailed Design	Construction & Commissioning
▪ Use of pre-defined generic production activities ▪ Define hierarchies and dependencies	▪ Use of a pool of predefined mechatronic modules ▪ Match available modules with required production activities	▪ Control engineering: **configure** module functionality ▪ Program additional control functionality	▪ Control integration via **Plug-and-Play** ▪ Virtual commissioning

Table 4. Engineering of a Decentralized Production Line – Overview of Phases

Process Planning
As previously, the production process for the product to be produced is described by a specification of the production steps by the operator of the production. In addition, the production process has to be structured hierarchically in production steps, and the predecessor-successor relation, as well as parallel processing between production steps, must be described. Likewise, the machine functionalities for the performance of a production step has to be characterized. This concerns both the control functionality and the kinematic behavior of machines. All mentioned additional steps serve for the preparation of the efficient implementation of the subsequent steps.

Plant Layout and Structure Planning
Based on these informations and the available mechatronic modules, the system integrator derives the facility parts to be installed and their technical interrelationship. Mechatronic modules that offer the desired functionalities are attributed to each production step. For this purpose, the mechatronic modules will be selected from a library that may also contain complete process step realizing production cells. For selection and matching, the predefined production functions of the mechatronic modules are used. If the production process was specified comprehensively as described in Process Planning, the Plant Layout and Structure Planing may be performed in a supported or automated manner.
In addition to the pre-specified, mechanical and electronic data, the mechatronic module also contains predefined control applications for the realisation of the production functions. In addition, the specific data of the utilized mechatronic modules for the geometrical positioning in the facility, their kinematic behavior, and their geometries in the 3D factory layout has to be substantiated. Furthermore, the interdependence is considered for the dimensioning of mechanics and electrics, e.g., for the layout of the power of motors.

Detailed Design
In the next step, the configuration of the predefined control functionalities of the utilized mechatronic modules is done in accordance with the specific process parameters, and, if necessary, the programming of auxiliary functions is performed.
In addition, the connections between mechatronic modules and the corresponding parameterizations of these connections have to be specified:

- Connection of electrical, pneumatic, and hydraulic systems,
- Connection of the communication devices to communication networks,
- Connection of material flow relations, and
- Attribution of I/O signals between sensors/actuators and control applications.

The connections are generated automatically or manually, depending on to what extent the connections have been specified in advance.
PLC and HMI programs are usually generated automatically from the control applications of mechatronic modules. In this planning phase, the design or the adaptation of the facility mechanics (CAD) and the electrical design (CAE) are also done unless large extends have been already predefined in the mechatronic modules.

MES/ERP Integration
In this planning phase, initially, the interfaces between the predefined process control and MES functions of the mechatronic modules on the one side and the facility MES level on the other side are designed. Engineering data that have to be considered in the design of the MES modules and the facility MES functions are

- the production process with its specified production steps,
- the facility layout, and
- the technical relations of the mechatronic modules in the facility.

Once this information is available in a comprehensive manner, the essential tasks can be partially automated. Additionally, control applications have to be adapted to specific MES functionalities – for instance, the scheduling of production – if support still exists. Once the MES functionalities are designed, the parameterization of the interfaces to the ERP system and the synchronization of relevant data between the MES and the ERP system are performed.

Construction & Commissioning
As the real used facility resources correspond to pre-integrated mechatronic modules, they can be provided in a manner in which they can be integrated into the automation system via plug and play. After corresponding adaptation, these facility parts can be used immediately.

Predevelopment of Mechatronic Modules
The essential precondition for the feasibility of the engineering of production facilities according to the described procedure is that, in the run-up, mechatronic modules are prepared in order to be available as a template in a library for the facility planner. This task requires an additional *project independent* phase and corresponds to a predevelopment (Fig. 6).

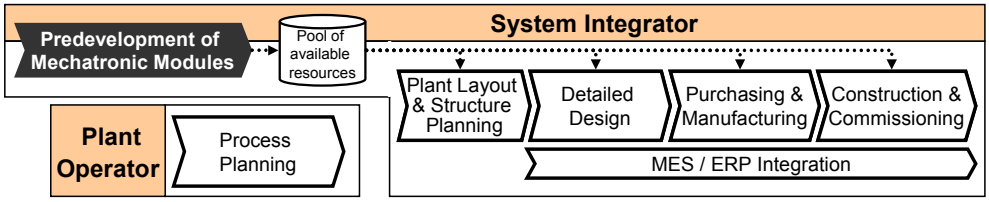


Fig. 6. Engineering process for decentralized production automation systems based on mechatronic modules

During the design of these facility and project independent mechatronic modules, the following information must be specified and provided in an integrated manner:

- Description of the mechanical parts, e.g., in CAD layout.
- Description of the behaviors of the facility resource, such as kinematic behavior.
- Description of the machine functionalities that the resource offers for the design of production steps (function oriented view).
- Description of the automatisation portions: control and communication interfaces, required connections for hydraulics, pneumatics, and electrics, as well as models or implementations of baseline control, HMI, and communication building blocks.
- Description of combination possibilities or limitations with other resources in order to combine individual functionalities into one overall functionality.

Fig. 7 shows the information flow for two of the abovementioned phases, which converges in an orderly fashion using mechatronic modules.

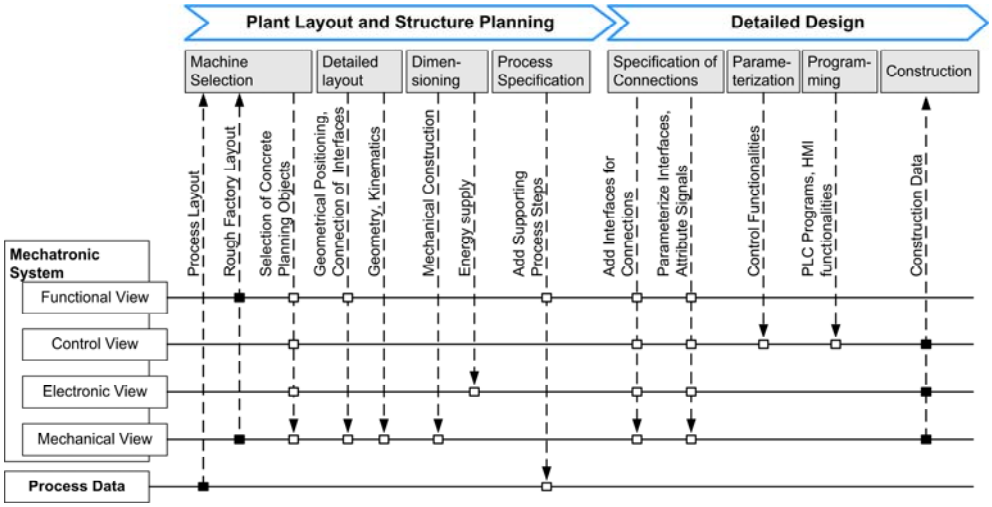


Fig. 7. Information flow between engineering tasks and modules

5.3 Engineering of a Material Flow System

As for the description of the production line scenario, for the engineering of an intralogistics material flow system a general engineering process is described; see Table 5, before a

model is presented that offers a possibility for the engineering of the decentralized material flow system. The phases refer to the partitioning of the processes as presented in Fig. 8.

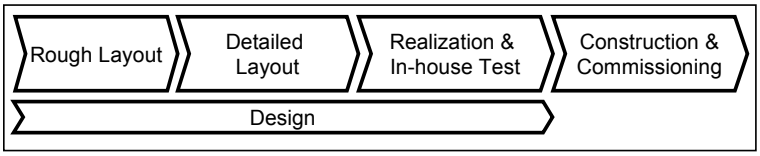


Fig. 8. Generalized engineering process for intralogistics systems

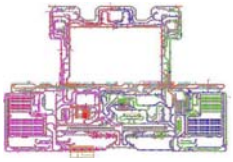
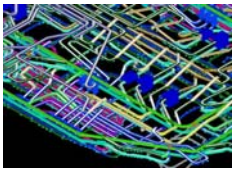
Phase	Basic Engineering Tasks	Example Results
Plant Topology and Rough Layout	Deduction of the rough layout based on the required material flow (requirement, particularly in view of transport functionalities, storage, handling, throughput of the factory), rough simulation for the detection of bottlenecks	
Detailed Layout and Specifications	Refinement and detailed specification of the requirements, establishment of detailed layout, adaptations, optimization of parts lists, concept for electrics, determination of the IT concept, strategies for control algorithms, definition of the interfaces between disciplines, refined simulation based on control algorithms	
Realization and In-House Test	Procurement, partial installation, PLC, and IT side configuration and programming – related individual tests/interface tests automation/IT	
Construction/ Commissioning	Installation of the systems, commissioning of control system, test of IT interfaces with other processes	

Table 5. Phases and Tasks of Basic Process for Engineering of Material Flow System

5.4 Engineering of a Decentralized Material Flow System

As shown in Chapter 3.2, the request for flexible integration and reconfiguration of an intralogistics system could be realized by the creation and continuous use of mechatronic modules. For this purpose, previously centralized automation functionalities are distributed to individual modules, which leads to the decentralization of the intralogistics system.

As explained there, this approach has implications on the architecture of the automation system and, therefore, on the handling of the modules. This is reflected by the introduction of an engineering process for decentralized systems that supports the engineering in the design, commissioning, and reconfiguration phases. This will be presented hereafter.

Plant Topology and Rough Layout

As common, a rough layout is created, based on the customer requirements and the specifications of the factory planners, that describes the general topology of the system and all source/target relations. From this layout, the system integrator identifies the factory parts to be installed and their technical interrelations as depicted in Fig. 9. These factory parts may consist of one or more mechatronic modules. One example for a factory part consisting of several modules is a buffer storage unit for intermediate storage of boxes, consisting of the material handling modules, a rack, and an automated storage and retrieval system (AS/RS).

In this phase, not only the factory components but also the mechatronic modules are still generic. For example, it is determined where the buffer storage unit is located in the topology, how many spaces it must have, which throughput it performs, and which functions are generally required. But it is not yet defined which particular conveyor types and which AS/RS will be used. At this time, a rough simulation of the process is performed for the verification of the throughput.

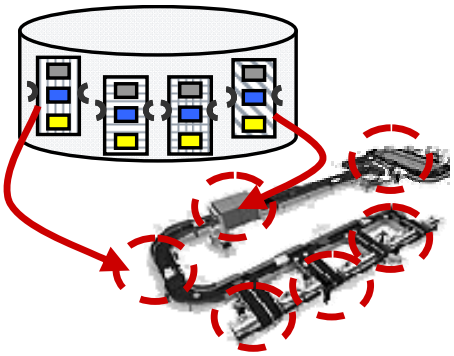


Fig. 9. Identification of appropriate modules

Detailed Layout and Specification

Now, a successive refinement of the developed requirements and definitions is done in a top-down approach. Initially, the requirements for the mechatronic modules are further detailed. Then, it is determined which mechatronic modules will be used in reality, e.g., the conveyor of Company A, that can move a certain maximum load, and a rack feeder of Company B. At this time, the identification of components to be developed specifically for the project is also made, such as, for example, a special gripping device for the rack feeder for the nonstandard boxes of the customer, including the corresponding automation system.

In this phase, the control and process concepts are specified. In opposite to central automation, in which the process logic can be explicitly specified, the procedures may only be defined through the rules of interrelations. As an example, the control of a transport process by driverless transport systems (Automated Guided Vehicles, AGV) can be considered. This control can be implemented by a framework of auctions. A bid for the box transport can be requested by the box while the AGVs will provide it based on availability of the corresponding AGV, its actual location, its transport capacity, and other parameters. The box transport than shall be assigned to the AGV with the most “beneficial” bid.

That means that the engineering process is performed at a more abstract level; the possibilities of interactions of the modules must be very familiar to the developer. Then, a refined simulation of the processes is performed. This simulation exploits functionalities of the mechatronic modules used later on in the actual operation in suitable granularity introducing them into the simulation by Plug&Play. This simulation allows the rough testing of the effects of different control concepts (i.e., different system configurations). This also allows identifying components and rules to be developed specifically for the project.
The interfaces between the modules are defined using standardized interface descriptions. The interfaces of the mechatronic material flow system to superimposed processes (e.g., ERP system or Warehouse Management System) and neighboring processes (e.g., the production control system) are defined. In addition, the connections of the mechatronic modules to the central infrastructure components are defined, such as, e.g., electrical power supply, compressed air supply, and connection to the IT network.

Realization and In-House Test
The individual mechatronic modules are configured and parameterized for their particular task in accordance with the requirements of the established specifications. For this purpose, the specialists of the different technical disciplines continue to work on their specific tasks. However, its work is based on consistent mechatronic modules that provide the corresponding views. The necessary project-specific designs are performed. Based on the control concept and the defined possibilities for communication and coordination between the modules, the concrete control mechanisms are configured and project-specifically extended if necessary. The functionality per se is generated at runtime by using defined rules.
An in-house test of the system is performed during which a pretest of the facility is done to the possible extent. The vertically modularized structure and the "Plug&Play" functionality support virtual commissioning by subjecting readily configured or installed facility parts to the test and recreate others that are not completed by simulation or emulation.

Construction/Commissioning
During the commissioning phase, the "Plug&Play" functionality, i.e., the use of pre-integrated modules with the capability to identify each other and coordinate among each other, is applied. It allows a bottom-up installation of the factory according to the factory layout. The physical modules possess mechanisms in order to recognize at which position of the factory topology they are located and perform their configuration and parameterization based on their specifications. They connect themselves with their neighboring modules in terms of automation and communicate with each other as illustrated in Fig. 10.
The use of pre-integrated mechatronic modules allows a functionality oriented installation of the factory and, therefore, an early testing of interconnected partial areas and, specifically, an early start of the commissioning of the automation. It is not necessary to integrate completely mechanics and electrics before commissioning the automation, but it is possible to combine and test small functional units such as, for instance, a material handling circuit that is able to move boxes while additional integration still occurs around it.

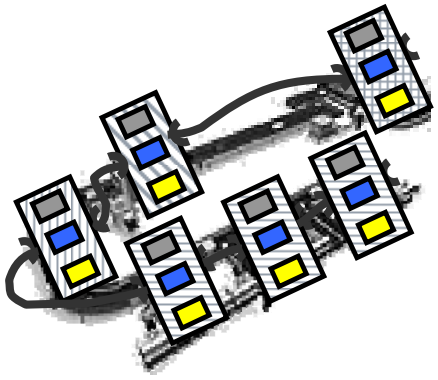


Fig. 10. Simplified Integration

Reconfiguration
Even in existing plants, reconfiguration procedures occur. They generally require extensive engineering processes. If, for instance, additional diverts shall be included in the automatic transport system mentioned above in order to be able to move to different locations, an entire engineering process needs to be performed for such a change. Then, the mechanics of the transport system must be separated at the plant locations in the context of the commissioning, the diverts must be mechanically installed, and subsequently, the electrics and the automation system must be installed and newly commissioned.

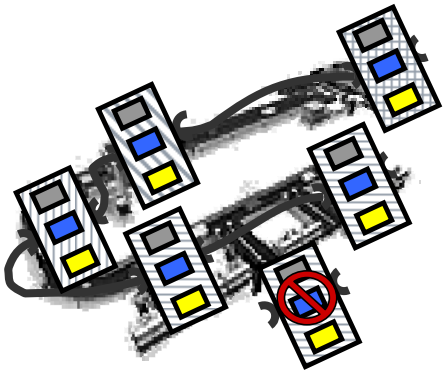


Fig. 11. Simplified Adaption to Changes

In the case of the decentralized mechatronic system, these activities are supported by the "Plug&Play" functionality of the modules. Modules can be removed from the overall system without causing far-reaching changes to the automation system (dynamic adaptation to changed topology) as shown in Fig. 11. In terms of automation, new modules, to a large extent, are autonomously integrated into an existing plant. The required extent of the engineering depends essentially, in this example, on the requirements for actuality

and consistency in the plant documentation, such as, for example, the updating of the plant layout.

Predevelopment of Mechatronic Modules
In order to keep the project specific effort for the use of these modules as low as possible, they must be defined and specified as far as possible in advance, also for the intralogistics system as described in Chapter 5.2. This means the creation of mechatronic module building blocks that are defined independent of specific projects. These modules provide all involved disciplines with their particular views using a consistent model. The building block approach supports the top down proceeding during the design phase by the provisioning of module descriptions based on which the initially still rough plant layout can be refined successively. Therefore, standardized means of descriptions must be applied. They contain required module information related to the different disciplines or to the proposed (transport) services, for instance.
In an intralogistics system there are functionalities that require comprehensive information, such as, the routing of a box to its target destination or the allocation of plant resources to tasks based on the current load of the system (e.g., selection of automated storage/retrieval system, AS/RS). They were previously performed through central systems. Such functionalities have to be replaced for decentralized plants by other mechanisms. This includes concepts for the substitution and distribution of such control logics, capabilities for communication, as well as for coordination of the modules, such as auctions. For instance, the decision finding for allocating a storage order to an AS/RS, is done rule-based in the decentralized case. Formerly, a central system maintained information about the load of the AS/RS and made decisions for a suitable device. Now, whether an AS/RS may offer its services or not has to be determined by rules. In addition, it must be determined according to which rules the devices coordinate among each other about the allocation of an order and, lastly, select one of them for the execution of the storage order.

6. Methodical Evaluation of Benefits and Risks

Changes, such as the introduction of mechatronic modules, a mechatronic approach in engineering, distributed control logic, and function-oriented commissioning, are only supportable if their impact, in terms of benefits and risks, on the entire industrial life cycle, has first been worked out in detail (Löwen, et al., 2005).
For the planning, realization, and commissioning phases, there is no well known procedure through which the added value of an automation concept can be determined. Regarding the introduction of new concepts, it is of high importance to evaluate benefits and risks before the actual practical deployment. Although an effort-and-cost based evaluation would be desirable, the lack of completed projects and reliable figures prevents the use of such an evaluation. In order to close this gap, an evaluation strategy was developed within the PABADIS'PROMISE and "Internet of Things" research projects. The basic idea of the method is the creation of reference models for engineering processes and activities along the life cycle of an automation solution. These models refer to the corresponding automation concepts, for instance, the classically developed system and the decentralized automation system developed mechatronically, to compare both.

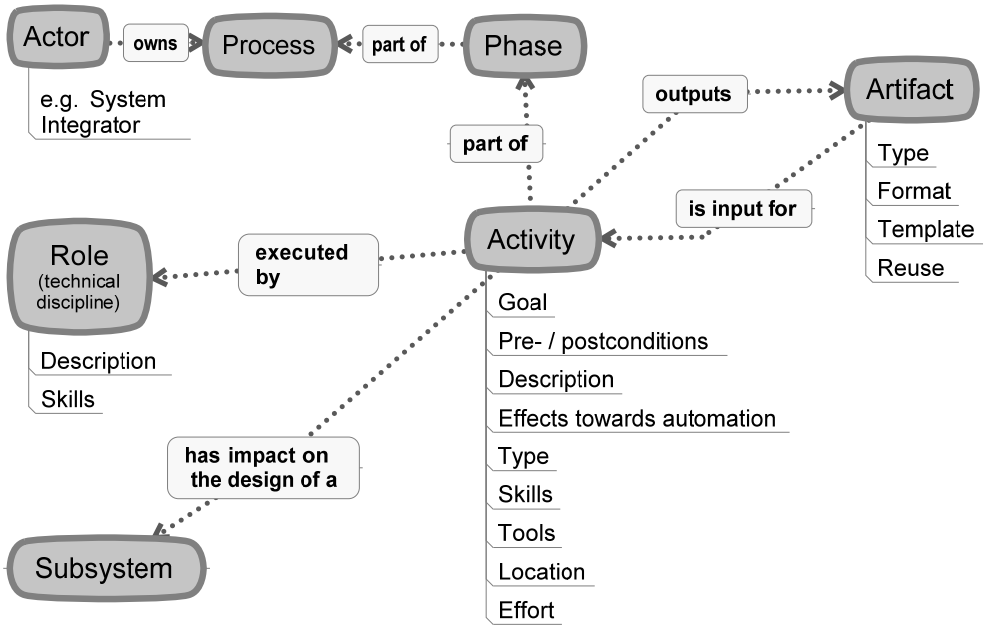


Fig. 12. Meta-model for structural and attributive features of engineering task

The creation of such a model is done by dissecting the life cycle of the plant in its phases, by the dissection of the processes (e.g., layout planning) within the individual phases into individual activities, and by the identification of their relation with the automation system.

For the creation of a neutral reference basis for the later comparison of different automation concepts, a meta-model was developed; see Fig. 12, which defines all required features of the corresponding reference models for engineering and commissioning in a uniform manner. As an auxiliary tool for building the reference models, corresponding templates were created to record activities/processes.

The evaluation is done by a comparative assessment. As one of the basics depicted in Table 6, generally applicable evaluation criteria were defined, derived from central levers in factory engineering, such as modularization, re-use, integration, and seamlessness (Löwen et. al., 2005), as well as under the viewpoint of applicability. They serve as guiding issues for the identification of critical factors in engineering and address specifically the challenges concerning the phases mentioned in Chapter 4. In addition, the standard ISO/IEC 9126, which regards the assurance of software quality, was evaluated, and the quality characteristics included therein were transferred, adapted, and amended to meet the requirements of plant business. The criteria catalogue thusly generated serves as the basis for the evaluation; an excerpt of questions can be found in Table 6.

Topic	Criteria	Question	Details
Suitability	Customer requirements	Which customer requirements are relevant for the system integrator in order to design and realize the plant?	▪ Typical customer requirements today → aims of system ▪ Control architecture specific requirements
	Appropriateness	Are the requirements considered in a suitable way?	e.g., highly redundant (and costly) automation system in case of non-time-critical processes is not appropriate
Interoperability	Content-related dependencies between activities	Which dependencies during the engineering process do exist between automation engineering and other disciplines?	▪ required information from mechanical engineering, electrical engineering and mechanical plant layout ▪ processes to be controlled/monitored, list of process signals ▪ sensors/variables to measure the process ▪ actuators/variables to influence the process
	View integration	How is cooperation of technical disciplines supported?	▪ e.g., integrated information models such as multi-discipline information objects

Table 6. Excerpt of Criteria Catalogue

For the automation concepts to be compared, according to the evaluation proceeding as shown in Fig. 13, a reference model is initially established for the processes and activities along the life cycle. The information in Chapter 5 serve as a basis for the creation of these models for the presented automation solutions. Subsequently, the different models are analyzed based on the evaluation criteria and compared to each other related to benefits and disadvantages. In the following, the results of the comparison of centralized to decentralized automation systems shall be presented.

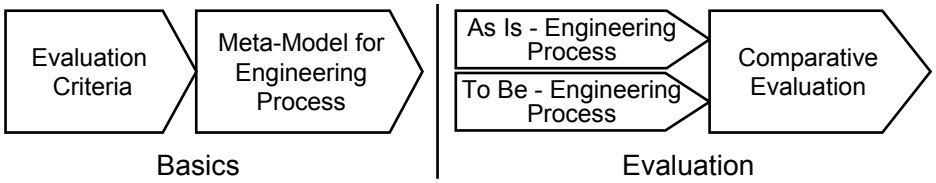


Fig. 13. Basics and evaluation proceeding of reference model method

7. Implications for the Engineering Processes

The detailed gathering and analysis of the individual processes and activities for both decentralized production systems and, also, decentralized material flow systems showed very similar results as compared to previous engineering processes. To begin with, the following findings are related to the activities to be performed and the resulting (expected) efforts.

Predevelopment: For the creation of mechatronic blocks, a correspondingly higher effort has to be undertaken for predevelopment. This includes:

- Adequate definition of standard modules with as few as possible technical and functional dependencies by systematic dissection of a plant,
- Coordination across the involved disciplines for the generation of mechatronic modules
- Standardized interface descriptions (functional, technical, information technical),
- Standardized description of process functions or partial processes provided by a module,
- Description of sequences of processes and the invocation of process functions,
- Provision of configuration possibilities and interfaces for project specific adaptations,
- Provision of module-specific automation control functionality and its implementation,
- Provision of application guidelines for modules (e.g., application areas, limitations, dependency on configuration possibilities, and variants).

The creation process for mechatronic module building blocks has to be done in cooperation and coordination with all participating stakeholders (system integrator, equipment/machine supplier, and component supplier).

Layout, Detailed Design, and Realization: By means of the project-independent creation of mechatronic modules, a shifting of engineering processes to earlier phases results:

- Control engineering is not an individual task any more, but is derived from models of the production process and of mechatronic modules.
- Depending on the reuse concept and the grade of standardization appropriate in this domain, the engineering process is essentially reduced to the selection and adaptation of the utilized modules through configuration.
- By means of the Plug&Play characteristics (predefined functionalities and coordinated interfaces) and the mechatronic building blocks, simulation or emulation can be used in a more realistic way for the mapping of the plant (use of the same software modules in both the automation system and the simulation).

The mechatronic approach supports parallel work and coordination across the involved disciplines for project specific engineering tasks.

Installation/Commissioning: Here, mainly the Plug&Play characteristics and the pre-integrated mechatronic modules are used:

- The effort for installation and commissioning can be reduced since essential functional and technical aspects in the modules have already been pre-integrated. This includes, particularly, the integration of the control system and MES functionalities.
- The existing mechatronic information and the detailed emulation possibilities facilitate virtual commissioning.
- Through standardized interface definitions and the use of proven modules, the integration risks can be reduced.
- By using mechatronic modules, a stronger focusing on rapid availability/completion of functionalities is possible during installation (installation according to functional areas and not according to diciplines). This also allows a faster completion of tests.

In summary, regarding the engineering efforts in projects, efforts are shifted from design and commissioning to process planning and plant layout. This is valid for both engineer-

ing of production lines and material flow systems. I.e., efforts are moved towards earlier phases in the project; see Fig. 14. Additional efforts (project independent) will be required for the development and maintenance of module libraries. The benefits occur in plant realization, commissioning, and reconfiguration. Standardization and reuse in terms of generic utilizable resources will reduce total costs in the long term.

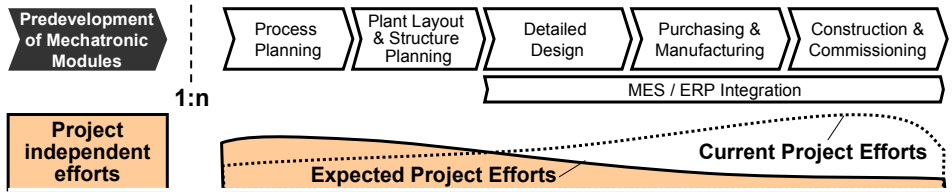


Fig. 14. Expected shifting of efforts, exemplary for engineering for production lines

8. Comparison of the Engineering Approaches and Conclusions

The collected information related to the activities along the life cycle of an automation solution serves as basis for the comparative evaluation between a classically developed system and a mechatronicly developed decentralized system. The following results have been worked out with the characteristics from decentralized automation use cases as introduced in Section 3 in combination with the described mechatronic engineering approaches. The general advantages of engineering with mechatronic modules, such as, for instance, cross-discipline- work and the improved reuse or standardization of partial solutions, were already discussed in other papers and shall not be described here any further (Thramboulidis 2008, Löwen & Wagner 2009, VDI 2206).

Application Effort and Complexity: The applicability and manageability of a modular, decentralized automation approach – particularly regarding the modularization of automation and control functionalities – must still be ensured. For this purpose, one must differentiate between the creation of the mechatronically modules in a predevelopment project and their application in the particular engineering projects. Overall, it can be expected that the specification and implementation effort to be performed in advance and the complexity for the creation of mechatronic modules with integrated automation and control functions, which are autonomous and can be integrated independently of each other, will increase substantially. For this purpose, bases for modules have to be created that are usable for different projects and configurable for the specific applications. This requires detailed knowledge of the domain, extensive technological expertise, and an understanding of the core functionality of the modules. One example for a starting point for this approach is the INTEGRA standard specifying first consistent mechatronic viewpoints for automotive production lines.
These pre-produced modules facilitate the engineering for project-specific applications. The more plug&play modules are available the less effort is required for the engineering. In addition, project risks can be reduced by early validation of fulfillment of customer requirements due to availability of module descriptions and models. This, however, only applies if module adaptations occur which can be done by configuration, selection of variants, or extensions, and in which do not require any changes to the predefined core

functionalities of the modules. In that case, there is no knowledge of the inner structure of the modules required, but only an understanding of the configurable functionality, interfaces, and external interdependencies. The modules themselves can be regarded as black boxes. For required project-specific adaptations at the core functionalities of modules, a deep technical understanding is required, for instance, relating to the internal design and structure or the interaction between the modules. For this activity, an increased complexity must be considered in analogy to the creation of the modules. The objective here is to find the optimum working point by properly cutting and sizing both plant structure and control tasks.
For the operating phase, it is important that the inner complexity of the system is hidden from the user. Previously existing interaction and visualization possibilities must continue to exist, which means that previously centralized information must now be collected in a decentralized manner. For this purpose, suitable mechanisms and utilities must be created.

Task Distribution/Automation Pyramid: The hierarchical structure of the automation pyramid has the objective of being able to manage the complexity of the overall system. This is obtained with the described approach by stronger functionally and modularly oriented structuring. Now each module integrates the needed automation functionalities for several automation levels. Decisions by the previous control level are shifting toward the field level, which means that decisions are made closer to the process location of activities. However, there still needs to be a layering within the automation software. This is since the direct control of field devices, for instance, by a decentralized control system based on an agent framework, appears improbable because of insufficient real time characteristics; see Fig. 15.

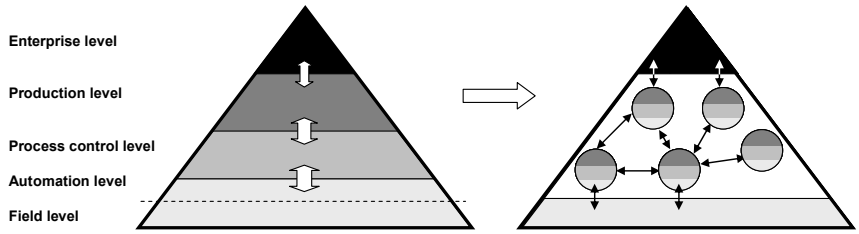


Fig. 15. Modifications of the automation pyramid in the decentralized approach

It became obvious that distribution is advantageous particularly in those cases where a plant can be structured in functionally separated areas and where information about the overall status is not frequently required for decisions. In such cases, a very high communication/coordination effort would be required and information would have to be available redundantly. Therefore, in plants, there will exist automation functions that still must be deployed centrally, like the plant visualization and archiving (SCADA system). Suitable methods are required for the easy adaptation of such systems to the application case in order to avoid additional separate engineering.
Determinism: Still today and particularly in intralogistics, very complex automation systems cannot be currently designated as strictly deterministic. Within a baggage handling system, for example, the routing is dependent on so many input values that decisions are

difficult to predict and to repeat. This development is even amplified by process control through distributed intelligence since the way in which such systems work is not determined by central algorithms but is rather based on more or less generic rules. For the engineering and configuration of the system, the utilized mechanisms and the interaction possibilities of the components among each other must be well known and manageable. This causes the simulation to become substantially more important in order to ensure the behavior pattern of the system.

Adaptation/Change Ability, Dynamic Behavior: The flexible, adaptable behavior of distributed intelligence is particularly recognized as an advantage during operation. Rule-based coordination mechanisms allow the reaction to events that are not precisely planned and the flexible adaptation to changed processes. These advantages pay off during the failure or repair of individual components, as well as during different load situations or for changed processes. As above, in this case an increased use of simulation during engineering will be required for the verification of the desired behavior.
Additionally, there is a discrepancy between the detailed project-specific requirements related to automation functionality and the necessary generalization, which is required for the provision of project-independent, decentralized components or modules. It is an open question, by which means an optimal flexibility of such modules can be predefined without increasing internal complexity, and how these predefined flexibility can be adapted to particular use cases without causing high complexity in engineering (also regarding the interaction between the modules).

Traceability: Precondition for the timing related correct retracing of events in the plant is a corresponding synchronization mechanism that sorts the (partial) events that are stored in the individual decentralized modules by time and that is able to create an overall picture. This capability is mandatory for troubleshooting in the framework of engineering activities, particularly during commissioning and acceptance, and is indispensable for customer acceptance. The corresponding required mechanisms, for instance, tools to support the troubleshooting for decentralized systems, have to be still further developed.

Robustness: By eliminating the central control computer and, consequently, a Single Point of Failure, the impact of a failure of decentralized automation components tends to be locally limited. Opposed to this are questions still to be analyzed, such as the consideration of data management and data security. Key words in this connection are: transaction security, data distribution/persistency, redundancies, backup/recovery concepts, etc. Adequate measures must be taken during engineering and lead to a corresponding higher engineering effort, as well as, possibly, to higher cost for more intelligent automation components and additional mechanical redundancies.

Formalization, Overlapping Models, and Standards: High project-independent investments are required for the development of modular mechatronic units that are standardized and usable in multiple projects. This modularization is subject to the planning process and needs to incorporate all stakeholders in the manufacturing life cycle. The need to introduce common models, interface and descriptions standards, as well as business-spanning standardized processes, arises. In general, standardization entails a struc-

tured, repeatable proceeding that has a positive effect on the quality of the overall system. However, a strong formalization and consistent information model through the engineering phases and automation levels are required for this purpose. Furthermore, new technologies are required for the provision of modular, configurable automation functions.
Additionally, it is subject to further discussion who will be the driver and responsible for module library development. Machine suppliers and system integrators (as well as plant owners) could be drivers, but in all cases, common standards are required that do not exist today. In addition, know-how protection will become an important topic in this scenario.
The creation of extensive module building blocks does not make economic sense in every case. It has to be individually verified which effort is appropriate for the module building blocks and how much effort is to be exerted for each specific project.

9. Migration Concepts

The evaluation of the benefits and risks of the decentralized systems has shown that their advantages can be utilized in different phases of the plant life cycle. During layout, detailed design, and realization such systems offer the chance to break up the previous discipline-oriented procedure with a mechatronic approach. A joint mechatronic data model, offers the opportunity to achieve continuity of data and provide the participating disciplines with data tailored to their required view of that data. This reduces the danger of inconsistent data, particularly for later technical changes. The construction and the commissioning of a plant assembled of decentralized components can be supported by a Plug&Play concept that allows the integration of individual components in a more simple manner, as well as a semi-automatic configuration of the modules. The advantages of decentralized automation during operation are generated by a higher flexibility with respect to changes in the workload or of production steps, and with respect to failure, repair or extension of a component. This allows dynamic reaction on unforeseen events.
Concerning the actual system architecture, the question arises regarding the Plug&Play concept as to whether it makes technical or economic sense to equip each module with its own control. This becomes even more difficult in functions that have previously been centrally available, for instance, a SCADA or a visualization system. It is not sufficient here to simply distribute these functions logically, as new concepts are required that lead to a principle change in the technical infrastructure of previous automation and control levels. Such decentralized systems, therefore, work at a different abstraction level; instead of concrete functionality, rules for coordination are established that then, for example, are worked off by an agent framework in the form of auctions or other mechanisms. Much further work has to be done before such systems are ready for the market. The actual implementation of agent platforms, for instance, are widely based on developments in the university environment and must be adapted to the industrial application, particularly in the automation environment, or may even need to be newly developed.
The domain-specific requirements of the customers are an important aspect of the economic feasibility of decentralized systems. For instance, for production lines as well as for baggage handling systems, customers (automotive manufacturers or airport operators) have extensive, specific and very concrete requirements as to how the automation architecture of such facilities shall be designed. A system without a central control computer

with multiple redundancies that would be controlled by a multitude of small, decentralized controllers is currently (still) hardly imaginable, e.g., for large airports.
In order to consider these findings and still be able to use the advantages of a decentralized approach, the procedure described hereafter is proposed. A modularly, decentrally designed plant can, by all means, still be realized by a centralized automation system. If continuity of modularity can be ensured from design to realization, this approach offers itself as a practicable compromise. Such a procedure is also supported by (VDI 2206), which describes the development methods for mechatronic systems. It is distinguished between the functional and spatial view of mechatronic modules: "The functional integration of mechanical and electrical/ electronic components takes place by connecting them by means of material, energy and information flows. The components may in this case be spatially separate from one another. In the case of spatial integration, the mechanical and electrical/electronic components form a structural unit in the sense of a common entity."
Even if the focus of this guideline is more oriented toward the construction of mechatronic products, such as a breaking unit, and less toward plant engineering, the findings described in it can be well applied. A very promising approach in this direction consists of, on one hand, performing the engineering mechatronicly, i.e., by using building blocks of independent mechatronic components in a logically decentralized manner, and, on the other hand, designing the plant per se physically/technically in a central manner (e.g., with a centralized control system).
By doing so, it is possible to build upon proven automation technologies and, at the same time, use the advantages of mechatronically -based engineering. New developed approaches supporting mechatronic based engineering processes can become an essential driver for this migration path (Drath et. al. 2008)
On the other hand, a migration concept can be developed resulting in actually decentralized configured plants. For instance, modular control code for an PLC program can be handled as a logically integrated part of the mechatronic component during plant design. In the framework of plant commissioning, one then omits spatial (i.e., completely decentralized as in Fig. 16 upper half) integration, and completes, instead, a mapping to a centralized automation system; see Fig. 16 lower half.
The abovementioned control code of a mechatronic component would not run on a decentralized controller that is spatially attached to the mechatronic component, but be attributed, rather, to an PLC on which it runs easily integrated since it is pretested and equipped with the corresponding mechanisms. The term PLC stands here representative for a classic PLC, a soft-PLC, a plug-in module, or other solutions, for instance, on a PC basis.
The plant created in this manner has "from the outside," essentially, the same appearance as the previous centralized plant but came into being in a different manner and is constructed differently on the inside. If desired, further reaching spatial decentralization may be done in subsequent steps.

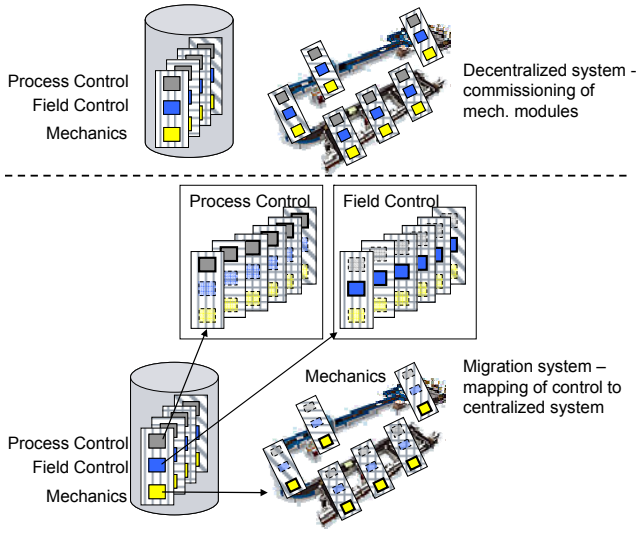


Fig. 16. Migration concepts for commissioning

10. Summary and Outlook

In this chapter, the impact of the currently much discussed decentralized automation on the total life cycle of factory automation systems has been examined. It became apparent that, while decentralized automation increases the flexibility of a production facility, it becomes unreasonably more expensive without corresponding adapted engineering. Essential to the introduction of such approaches are, therefore, adapted engineering processes through which solutions can be provided that are optimally prepared for this challenge.

For this purpose, the basic idea of decentralized automation approaches for factory automation systems – the distribution of production intelligence to individual autonomous automation units – was transferred to the entire engineering of the automation solution. The consequent involvement of all disciplines participating in the development of a facility and its assets leads to a mechatronic approach in engineering using pre-developed and pre-integrated mechatronic modules. Corresponding engineering processes for the use cases production systems and intralogistic systems were presented, and the most important differences were exposed. In addition, the impact on the individual participants in the facility life cycle, as well as the benefits and disadvantages, were discussed.

It became clear that modularization and decentralization are closely interrelated and profit from each other: Independent, standardized modules in plant engineering, which contain automation functionality, can only be meaningfully designed independent of projects if the control strategy is also distributed. How promising and cost and work intensive such disassembly/decentralization is correlates heavily with the ability of the plants in an area to be modularized, i.e., with the variance of physical, organizational, and control conditions. In addition, it became clear that implementation requires extensive run-up work and investment, as well as intensive cooperation between all stakeholders partici-

pating directly or indirectly in the value-creating chain for the construction of a facility. In each individual case, there is always a specific cost/benefit consideration required for the corresponding domain and the business environment.
Remaining challenges are to develop common models and standards (e.g., for module description, interfaces, and integration technologies) for cross-discipline and cross-value-chain modularization. In addition, business-spanning processes and business models need to be found to preserve the value-additions for each stakeholder. Finally, an obviously clear but largely unsolved challenge must be overcome: the establishment of modularization and mechatronic procedures consequently and consistently in practical applications.

11. References

Achatz, R. & Loewen, U. (2005). Industrieautomation, In: *Software Engineering eingebetteter Systeme: Grundlagen, Methodik, Anwendungen,* P. Liggesmeyer, D. Rombach (Ed.), pp. 497–525, Elsevier, Spektrum Akademischer Verlag, München

Alonso Garcia, A. & Drath, R. (2007). AutomationML - Die Motivation. *Computer und Automation*, Vol. 10/2007, pp. 28-32 WEKA Zeitschriftenverlag, Poing

Barata, J. et. al. (2001). Integrated and Distributed Manufacturing, a Multi-Agent Perspective. *Proceedings of 3rd Workshop on European Scientific and Industrial Collaboration*, pp. 145-156, Entschede, 2001

Bullinger, H.-J. & ten Hompel, M. (2007). *Internet der Dinge*, VDI-Verlag, Düsseldorf

Bussmann, S. (1998). Autonome und kooperative Produktionssysteme. In: Informationstechnik und Technische Informatik it+ti 40 (1998), Vol. 4, pp. 34-39.

Drath, R.; Lüder, A.; Peschke, J. & Hundt, L. (2008) An evolutionary approach for the industrial introduction of virtual commissioning, *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, Germany

Diep, D.; Alexakos, C. & Wagner T. (2007). An Ontology-based Interoperability Framework for Distributed Manufacturing Control. *Proceedings of International Conference of Emerging Technologies und Factory Automation (ETFA 2007)*, pp. 855-862

Elger, J. (2007). Das Internet der Dinge – Der Weg zur modularen Automatisierung. *25. Dortmunder Gespräche*, Bundesvereinigung Logistik in Kooperation/Fraunhofer Institut für Materialfluss und Logistik 2007

ElMaraghy, Hoda A. (2005). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing, Vol. 17 (2005)*, pp. 261-276

Habib, M. K. (2007). Mechatronics – A Unifying Interdisciplinary and Intelligent Engineering science Paradigm, *IEEE Industrial Electronics Magazine*, Vol. 1, I. 2, 2007, pp. 12-24, IEEE Computer Society Press, Los Alamitos

Hall, K. et al. (2005). Experience with Holonic and Agent Based Control Systems and their Adoption by Industry, *Proceedings of HoloMAS 2005*, pp. 1–10

IEC 62264 (2008). *Enterprise-control system integration – Part 1: Models and terminology, Part 2: Object model attributes, Part 3: Activity models of manufacturing operations management*, International Electrotechnical Commission, 2003-2008.

Internet of Things. Wandelbare Echtzeit-Logistiksysteme auf Basis intelligenter Agenten für den produktionsnahen Bereich, www.internet-der-dinge.de

Jennings, N. (2000). On agent-based software engineering. *Artificial Intelligence,* Vol. 117 (2000), pp. 277-296, Elsevier Press

Löwen U. & Wagner T. (2009). Modeling of complex technical systems – challenges and experiences in industrial plant business, *Proceedings of Mechatronic 2009*, pp. 27-34, Wiesloch, May 2009, VDE-Verlag, Düsseldorf

Löwen, U. et al. (2005). Systematization of industrial plant engineering. *atp – Automatisierungstechnische Praxis*, Vol. 47, I. 4, pp. 54-61

Lu, Y. & Jafari, M. A. (2007). Distributed intelligent industrial automation – issues and challenges, *atp – Automation Technology in Practice*, Vol. 5, I. 1 (2007), pp. 18-24

Lüder, A. et al. (2007). The Pabadis'Promise Architecture. *Journal Automazione e Strumentazione*, November 2007, pp. 93-101

PABADIS'PROMISE (2008). Plant Automation based on Distributed Systems - Product Oriented Manufacturing Systems for Re-Configurable Enterprises. FP6-IST016649, www.pabadis-promise.org

Parunak, H. v. D. (1998). *Practical and industrial applications of agent-based systems*. Environmental Research Institute of Michigan (ERIM), 1998.

Shen, W.; Hao, Q.; Yoon, H. & Norrie, D.H. (2006). Applications of Agent Systems in Intelligent Manufacturing: An Update Review, *International Journal of Advanced Engineering Informatics*, Vol. 20 (4), pp. 415-431

Sundermeyer, K. & Bussmann, S. (2001). Einführung der Agententechnologie in einem produzierenden Unternehmen. *Wirtschaftsinformatik*, Vol. 43, I. 2 (2001), pp. 135-142.

Thramboulidis, K. (2008). Challenges in the Development of Mechatronic Systems: The Mechatronic Component. *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, Germany

Tönshoff, H. & Woelk, P.-O. (2001). Flexible Process Planning and Production Control Using Co-operative Agent Systems. *Proceedings of International Conference on Competitive Manufacturing* (COMA 2001), Stellenbosch.

VDI 2206 (2204). Design methodology for mechatronic systems. *VDI-Guideline 2206*, Verein Deutscher Ingenieure, Gesellschaft Entwicklung Konstruktion Vertrieb, Juni 2004

Wagner, T. & Göhner, P. (2006). Flexible Automatisierungssysteme mit Agenten. *Proceedings of VDE Kongress 2006,* pp. 291-296, Aachen, 2006, VDE-Verlag, Berlin

Wagner, T.; Goehner, P. & Urbano, P. de A. (2004). Softwareagenten - Einführung und Überblick über eine alternative Art der Softwareentwicklung. Teile I - III. *atp - Automatisierungstechnische Praxis,* Vol. 45 (2003), I. 10, pp. 48-57, I. 11, pp. 57-65, and Vol. 46 (2004), I. 2, pp. 42-51

Wagner, T.; Schertl, A.; Elger, J. & Vollmar J. (2008). Evaluation of Effectiveness and Impact of Decentralized Automation. *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, Germany

Weiß, G. (2002). Agent Orientation in Software Engineering. *Knowledge Engineering Review* Vol. 16, I. 4 (2002), pp. 349-373

2

Wireless Technologies in Factory Automation

Aurel Buda, Volker Schuermann and Joerg F. Wollert
University of Applied Sciences Bochum
Germany

1. Introduction

Wireless technologies are utilised in consumer applications for several years, now. However, the operation of radio based communication systems in automation applications was considered doubtful for a long time. Primarily, the highly fluctuating quality of wireless transmission channels, compared to wired ones, was responsible for this fact. Transmitted electromagnetic waves experience reflexion, scattering, and diffraction, which may cause constructive or destructive interferences of the different signal copies arriving at the receiver. The direct consequences are packet errors and losses, resulting in higher transmission delays. Especially industrial propagation environments, with a lot of metallic surfaces and moving objects, are classified as demanding for wireless transmission. While fluctuations in latency and short losses of connections may be tolerated in consumer applications, exceeding given timelines in automation applications implies intolerable errors. The results are low plant availabilities and decreasing productivity.
By means of the development of diverse wireless standards and the adaption of well-suited protocols for industrial applications, the end-users doubts could be reduced dramatically over the last decay. The advantages of wireless solutions are obvious. In harsh environments, mobile and rotating scenarios, or at positions, difficult to access, cable connections and sliding contacts represent a main source of error. In this context the error probability can be decreased and the maintenance intervals increased by the utilisation of wireless technologies. In addition to that, there is a great potential on saving time and money during planning, installation, and commissioning of plant sections. Many domains of the industrial automation already profit from the deployment of wireless solutions. With KNX RF (Konnex, 2006) and ZigBee (ZigBee Standards Organization, 2007) the first standards for building automation have been introduced. Within the scope of the HART 7 specifications (HART Communication Foundation, 2008) the first standard, WirelessHART, for the process automation was released in late 2007. Further standards, especially for the domain of factory automation, are expected to get published in 2010. In order to reduce costs and time during the development of wireless solutions, unlicensed frequency bands are typically used for operation. Moreover an almost worldwide harmonised operation is guaranteed. This tendency is very pronounced for the 2.45 GHz ISM (Industrial, Scientific, and Medical) frequency band. Because of the high availability, transceiver chips of commercial standards are

often applied on the physical layer. Good examples are the technologies of IEEE 802.11 (LAN/MAN Standards Committee of the IEEE Computer Society, 2007a), IEEE 802.15.1 (LAN/MAN Standards Committee of the IEEE Computer Society, 2005)/Bluetooth or IEEE 802.15.4 (LAN/MAN Standards Committee of the IEEE Computer Society, 2006). Previous analysis (Willig et al., 2005; Vedral et al., 2006; Vedral & Wollert, 2006) attested the capabilities of these technologies for industrial applications. In order to comply with the requirements in automation with respect to determinism, reliability, and availability, above the physical layer adapted protocols are implemented. With latency requirements of 100 ms … < 1 ms, the domain of factory automation is one of the most demanding. With state of the art technologies, wireless solutions may serve applications with update times of 10 ms…20 ms. Below these timelines the reserve for packet retransmissions and error corrections of the current technologies is insufficient to guarantee a reliable communication. The upcoming ultra wideband (UWB) technologies are considered potential candidates for faster wireless solutions of the next generation. Within the scope of the chapter "Wireless Technologies for Factory Automation" the current state of the art and research of wireless technologies in factory automation gets introduced.

The rest of the chapter is organised as follows. Section 2 gives a short overview of the requirements of wireless technologies in factory automation and its possible applications. In Section 3 the properties of wireless transmission channels with respect to industrial environments are depicted. In addition to that, performance increasing techniques are stated, as well. Section 4 describes the state of the art of wireless base technologies in fabric automation, their utilisation and the regulation for the 2.45 GHz ISM frequency band. Furthermore the impact of coexisting wireless technologies and the necessity for coexistence management are discussed. Section 5 gives an overview of upcoming wireless technologies and the frequency regulation for UWB devices. Section 6 gives a conclusion of the most important information.

Due to the large span of the topic, this chapter can only give an overview of the recent properties and technologies. For further insights the most important references of each subject are listed.

2. Performance Requirements of Communication Systems in Factory Automation

Figure 1 shows the hierarchical model of automation. With respect to the current state of the art, wireless technologies may be applied, as a supplement to existing wired communication networks, across all levels of the model. The major part of automation devices is equipped on the control and sensor/actuator level.

Basically, the control level consists of programmable logic controllers (PLCs) and (intelligent) decentralised I/O-Terminals, which are interconnected by one or more field buses (IEC, 2007a). Depending on the plant, the number of devices varies and their distances may range from a few to some hundreds of meters. Classical field buses for example are Profibus or Interbus (IEC, 2007b). Current field buses aim at convergence to the well established Ethernet IT-networks with data rates of up to 100 Mbps. Popular examples are Profinet, EtherCat, Ethernet-Powerlink, Sercos III and Ethernet/IP (IEC, 2007c). Depending on the area of application typical cycle times of 10 ms…100 ms (PLC to PLC, Visualization),

1 ms…10 ms (Communication to decentralized peripherals), and < 1 ms (Motion Control) are required (Jasperneite, 2005).

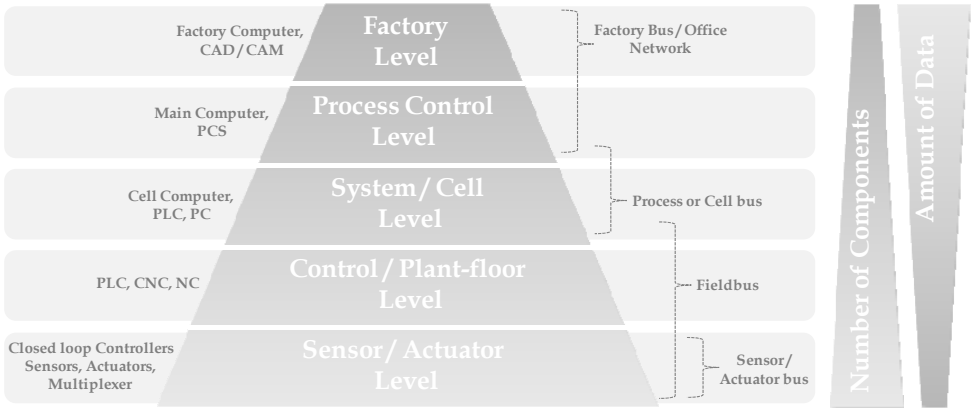


Fig. 1. The hierarchical automation pyramid.

The transition from control level to the sensor/actuator level is fluent. At this level sensors, actuators, drives, and manufacturing robots are equipped. The number of I/O-points is very high and densities of 2/m³ to 10/m³ can be expected in typical production plants (Scheible et al., 2007). The major part consists of sensors like proximity switches. The connection of sensors and actuators to PLCs can be realised in different ways. A differentiation is made between parallel and serial wiring. The classical parallel wiring uses analogue or digital (remote) I/O-Terminals to connect the sensors and actuators with the field bus system. New concepts, like IO-Link (IO-Link, 2009), use a 2/3-wire-standard-parallel-wiring of the sensors and actuators, as well. However, on the basis of a master/slave protocol, additional information, like diagnosis or parameters for configuration, can be exchanged with a PLC besides the process data. IO-Link specifies data rates of at least 4.8 kBaud and 38.4 kBaud. Corresponding to the amount of process data, cycle times of 2 ms can be achieved at data rates of 38.4 kBaud. The serial wiring is usually realised by means of a sub bus system, which serially interconnects the sensors and actuators. The most representative technology is the AS-Interface (AS-Interface, 2009). At a maximal amount of 31 (62) devices, the cycle times are < 5 ms (10 ms). Because of the high requirements concerning latency, drives and robots are interfaced directly to the field bus via appropriate application profiles or special drive buses, i.e. Sercos.

The application of wireless technologies is useful on the control as well as the sensor/actuator level. However, besides the stringent requirements concerning latency, determinism, reliability, and data integrity, the requirements with respect to costs, energy consumption, distances, bandwidth, node density, and network topologies differ.

Depending on the application, the requirements with respect to costs, energy consumption and node density are of minor importance on the control level. Because of the wide spreading of PLCs and decentralised peripherals, the ranges between devices may vary from a few meters up to some hundreds of meters. The process data of a whole plant is exchanged via the field bus system. In this conjunction the amount of process data of decentralised periph-

erals may vary between < 10 Bytes…> 100 Bytes. Hence, the bandwidth and data rates are of major importance. The size of the actual data packets depends on the structure of the field bus system and whether it uses multi-slave or single-slave frames. The network topologies for wireless solutions range from simple cable replacement point-to-point and point-to-multipoint connections up to cellular networks with roaming capabilities (production lines, automated guided vehicles).
Because of the high quantities of devices, the costs for acquisition, installation, commissioning, and operation are of major importance on the sensor/actuator level. The sphere of action is restricted to small production cells (10 m^3…100 m^3) with high node densities. The amount of process data of a single sensor or actuator typically ranges from 1 Byte…10 Bytes. Hence, lower data rates and bandwidth are sufficient. In its simplest form, wireless solutions operate as cable replacements in point-to-point topology, as well. However, the development is focused on high speed wireless sensor/actuator networks (WSANs), supporting large numbers of devices. These networks are usually arranged in star topology and consist of wireless sensors/actuators, wireless I/O-concentrators, and a master base station, which acts as the interface to a super ordinate control system. Due to the increasing latencies, multihop topologies are currently not considered for WSANs in factory automation.

3. Industrial Wireless Communication Channels

Communication systems have to comply with the stringent requirements concerning reliability, availability, and determinism in order to serve automation applications. In contrast to that, the quality of a wireless transmission channel experiences random time and frequency variant fluctuations. Hence, the development of wireless communication systems, for the extreme time critical area of factory automation, is a big challenge.
Industrial environments are often characterised by a high degree of metallic surfaces and time-varying influences. Besides the movement of the radio systems itself the movements of materials/tools, rotating machines and persons are responsible for this time variant properties. In principle industrial radio channels are akin to mobile radio channels. Thus, most phenomena of industrial radio channels comply with the ones of mobile radio channels. The occurring physical phenomena of transmitted electromagnetic (EM) waves are illustrated in figure 2:

- Reflexions occur, when EM-waves encounter reflecting objects, whose dimensions are much larger than the wavelength.
- Scattering appears, either when the dimensions of the encountered object are much smaller than the wavelength of the EM-wave, or when the surface structure is classified very rough in comparison to the wavelength.
- Diffraction occurs when EM-waves encounter sharp edges.
- Shadowing is caused by obstacles, which completely block the propagation paths of EM-waves.
- Doppler effects arise, either when there is a relative movement between transmitter and receiver, or a mobile obstacle in the propagation field reflects, scatters, diffracts, or shadows the EM-wave.

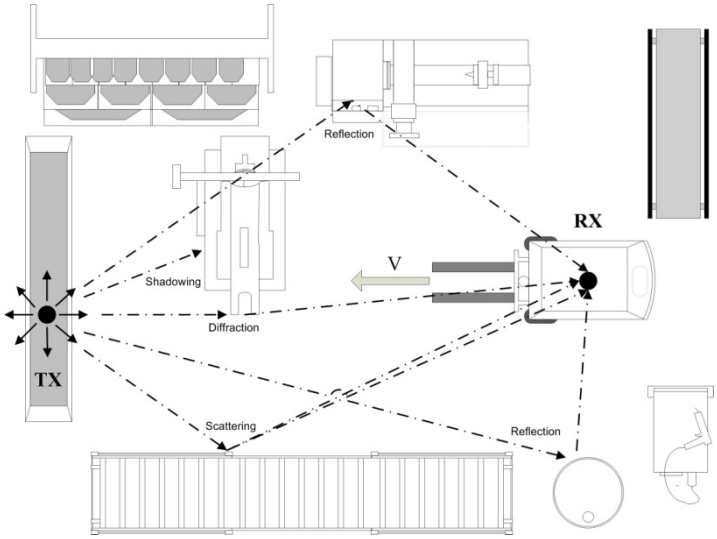


Fig. 2. The classical propagation of electromagnetic waves in a typical industrial environment.

Because of these wave phenomena a received signal is a composition of different attenuated and phase shifted versions of the original transmitted signal. Depending on the phase of these versions, a constructive or destructive overlapping occurs at the receiver. This effect is called multipath scattering. The absence of a direct non reflected version of the transmitted signal is typical for industrial radio channels. Does a relative proper motion between the transmitter and receiver take additionally place, or does the environment change due to rotating machines or forklift trucks, a shift in frequency based on the doppler effect influences the transmitted signal. Simultaneously, the path of the signal versions change, resulting in a new form of the received signal. Hence, the transmission behaviour of such a radio channel is time-variant and the signal power experiences high fluctuations.

3.1 Large Scale Fading

The large scale fading results from widespread movements. It depicts the mean signal power over spatial areas of about 10 wavelengths λ. Consequently, the local mean values of the propagation losses (path loss), which depend on the environment (shadowing, reflexion, diffraction, scattering), are characterised. In this conjunction the log-distance path loss model (Rappaport, 2002) is often used to describe path losses. The model states, that the mean received power P_r decreases logarithmical with the distance d between transmitter and receiver, following $P_r \propto P_0(d_0/d)^\gamma$. d_0 is a reference distance near the transmitter, where the transmit power P_0 is measured with respect to the far-field characteristics of the transmit antenna. The degree of signal attenuation is expressed by the path loss exponent γ. A detailed overview of the values of γ is given in (Rappaport, 2002). In buildings γ may vary very much. At frequencies of 400 MHz…4 GHz γ can take values of $\gamma = \{2, \ldots, 6\}$ (Hashemi, 1993). In analysis of Rappaport (Rappaport, 2002; Rappaport & Mcgillem, 1989, Rappaport,

1989a), performed in five different factory environments, mean values of $\gamma = \{1.7, \ldots, 3\}$ were measured.

3.2 Small Scale Fading

Small scale fading characterises the fast fluctuations of radio channels over short distances (fraction λ). Primarily, these fast fluctuations of the channel are caused by doppler effects and multipath scattering. If, for example, a narrow band carrier signal is transmitted, several randomly organised signal copies arrive at the receiving antenna via different paths. For every location in a propagation environment, the received signal is the sum of all signal versions. If the signal versions, which arrive at the receiver, are uncorrelated in phase, the angles of arrival uniformly distributed, and the signal delay of each path much lower than the alteration speed of the radio channel, then the behaviour of attenuation can be described by two complex gaussian processes with mean values of $\mu = \mu_R + j \cdot \mu_I$. If there is no direct line of sight (NLOS) between transmitter and receiver, the mean value is $\mu = 0$. In this case the probability distribution of the absolute amplitude values corresponds to the rayleigh distribution. If there is a direct line of sight (LOS), the mean value μ takes the amplitude value of the signal version, transmitted over the direct path $\mu = A_{LOS}$. The absolute amplitude values of these channels correspond to the rice distribution. Figure 3 shows a classical course of the absolute amplitude values of a rayleigh fading channel. The deep fades of up to 40 dB are characteristic. Analysis in industrial environments (Rappaport & Mcgillem, 1989) showed a dynamic range of 20 dB in signal power, for stationary transmitters and receivers. When the receiver was moved with a velocity of $v = 0.3\,m/s$, the dynamic range of the received signal increased to 30 dB…40 dB. If a channel experiences such a deep fade, several channel errors occur, whose positions show a strong statistical dependence (Paetzold, 1999). The occurrence of channel errors temporarily appears in complex blocks.

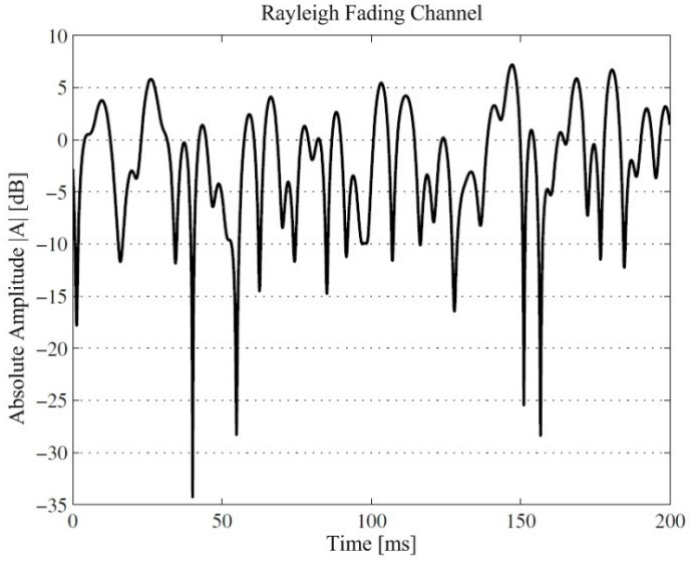


Fig. 3. The course of amplitudes of a rayleigh fading channel.

Since the rayleigh and the rice models are derived on the assumption of a non modulated carrier signal, their application is restricted to narrow band signals.
In order to completely characterise a radio channel with respect to the domains of time and frequency, the time variant impulse response $\underline{h}(\tau, t)$ is an appropriate measure. On the supposition of a wide sense stationary uncorrelated scattering (WSSUS) channel, the following characteristics can be approximated on the basis of Fourier transformations of $\underline{h}(\tau, t)$ and the computation of first and second order statistics (Bello, 1963):

Delay spread:
The delay spread τ_{rms} describes the mean spread in time of transmitted δ-impulse. Scientific studies showed a delay spread of $\tau_{rms} = 20, \ldots, 30ns$ at frequencies of 1.3 GHz in industrial environments (Hashemi, 1993; Rappaport, 1989b). In this conjunction the works of Haehniche et al. (Haehniche et al., 2000; Haehniche, 2001) are of great practical interest. The delay spread for the 2.45 GHz ISM frequency band was analysed in different industrial environments. A mean value of 72 ns and a maximal value of 121 ns were measured. Hoeing et al. (Hoeing et al., 2006) analysed the delay spread in a production cell with several scattering obstacles. The transmission distance was 3 m with LOS between transmitter and receiver. Within the propagation area of interest, fast cyclic movements of machines took place. Under these conditions a delay spread of $\tau_{rms} = 79ns$ was measured, which corresponds to a path difference of about 23.7 m in length.

Coherence bandwidth:
Within a frequency area of Δf, which is smaller than the coherence bandwidth B_C, the course of ampitudes is expected to be constant. Between the delay spread and the coherence bandwidth the approximation $\tau_{rms} \approx B_C^{-1}$ is valid. Haehniche et al. analysed the coherence bandwidth in different industrial environments, as well. Mean values of the coherence bandwidth $B_C = 5.7MHz$ were measured for the 2.45 GHz frequency band. In (Scheible, 2007) a coherence bandwidth of up to 10 MHz is reported for this frequency range.

Coherence time:
The coherence time T_C is a measure for a radio channels alteration speed.

Doppler spread:
The doppler spread D_S describes the mean frequency spread of a tranmitted narrow band carrier signal. Between the coherence time and the doppler spread the approximation $T_C \approx D_S^{-1}$ is valid. The impact of the doppler spread in industrial radio channels may be enorm. Fast moving or rotating machines may induce high values of the doppler spread. Hoeing et al. have measured values for D_S of up to 400 Hz.

On the basis of the presented characteristics, the small scale fading can be further classified with respect to the variance in time and frequency of a radio channel. If the signal bandwidth is much smaller than the coherence bandwidth $B_S \ll B_C$, and the delay spread much smaller than the symbol duration $\tau_{rms} \ll T_S$, the radio channel is characterised as flat fading (non frequency selective). Flat fading channels are often referred to as narrow band channels. If the signal bandwidth is larger than the coherence bandwidth $B_S \gg B_C$, the channel is frequency selective. In this case the delay τ of single paths is larger than the symbol duration T_S, what might induce intersymbol interferences (ISI) at the receiver. The time selectivity of a radio channel may either be described on the basis of the coherence time T_C or the doppler spread D_S. If the symbol duration is much samller than the coherence time $T_S \ll T_C$, the form of the transmitted symbol is not altered by the radio channel. These channels are referred as

slow fading (non time selective). The opposite is a time selective radio channel referred to as fast fading.
For a more detailed description of industrial radio channels the authors refer to (Vedral, 2007).

3.3 Performance-Enhancing Strategies

In order to comply with the challenging requirements of automation in the face of the depicted fluctuations of industrial radio channels, several performance enhancing strategies can be applied. It is obvious, that these methods are most effective, when implemented in the PHY or MAC layers. However, with the given architectures of available transceivers it is often necessary and only possible to implement appropriate protocols on application layer (Pellegrini et al., 2006).
Classical methods to improve the performance of radio channels are error detecting (retransmissions) or error correcting codes (Liu et al., 1997; Haccoun & Pierre, 1996; Biglieri, 2005), which add further redundancy to the transmitted data. Since these methods are typically applied to a single channel, their effectiveness mostly depends on the small scale properties of the channel. Deep fades induce dense blocks of errors, which can be hardly corrected by error correcting codes. The success of a retransmitted signal depends on the duration of these deep fading (coherence time). A way to overcome these problems is the utilisation of diversity techniques. In general diversity describes the transmission of information over different channels. The achievable gain depends on the statistical independence of each transmission channel. With an increasing number of independent transmission channels the probability increases, that at least one channel is in a good state, and the transmitted signal can be decoded at the receiver. If the error generating processes are completely uncorrelated, the theoretical minimal error probability is $P_r = P_e^n$ for n transmission channels. Diversity techniques can be applied in the domains of time, frequency, space and angle. Since time diversity implies an increasing latency, its operation in time critical applications is not suitable. However, by applying spatial or frequency diversity, significant gains at reasonable costs can be achieved.
Spatial diversity may be applied in different forms. A classification is made for single-user and multi-user approaches. In the case of single-user, there is only one transmitter and one receiver, with at least one of which having multiple antennas. In (Diggavi, 2004) it is proven, that the achievable capacity nearly linearly increases with $N \to \infty$, if both transmitter and receiver are equipped with the same number of antennas N. In its simplest form, multiple antennas are used at the receiver (SIMO). The single signal versions are combined at the receiver in order to produce the received signal. Well known combining techniques are switched combining, equal gain combining or maximum ratio combining (Goldsmith, 2005). The achievable diversity gain thereby depends on the statistical independence of the received signals. On the assumption of a rayleigh fading channel the normalised correlation coefficients $\rho(\zeta)$ of two envelopes can be expressed as a function of antenna separation (Clarke, 1969) $\rho(\zeta) = J_0^2 \cdot (2\pi\zeta)$. ζ represents the seperation of two vertical monopole antennas in wavelengths and J_0 is the Bessel function of first kind and zero order (Zeppernick & Wysocki, 1999). In (Vedral et al., 2007) practical measurements, in order to evaluate digital diversity techniques, were performed, based on a multi-transceiver platform, operating in the 2.45 GHz frequency band. By utilising three receiving antennas at a separation of 4.69 cm a diversity gain of 3.5 dB could be realised in an industrial environment. Bit error

rates (BER) could be reduced by half an order of magnitude compared to a single branch. The packet error rate (PER) could even be reduced by more than one order of magnitude. Based on more complex MIMO approaches (Boelcskei, 2006; Paulraj et al., 2004), i.e. applied in the upcoming standard IEEE 802.11n, performance gains can be further increased. The capabilities of multi-user approaches, i.e. relaying (Lanemann et al., 2004; Kramer et al., 2005), for industrial applications has been demonstrated in (Willig, 2008).
A second form of diversity is the transmission of Information over multiple frequencies. The achievable diversity gains depend on the statistical independence of the single transmission channels, as well. To obtain statistical independence between two channels their frequency separation should at least be larger than the actual coherence bandwidth. Following (Clarke, 1969), the normalised correlation coefficient $\rho(\Delta f)$ of two envolpes can be expressed as a function of frequency seperation $\rho(\Delta f) = (1 + (2\pi T \Delta f))^{-1/2}$. Thereby Δf describes the seperation of the two frequencies and T is the maximal delay spread of a current environment. In narrow band systems frequency diversity is often combined with time diversity in the form of "frequency hopping spread spectrum" (FHSS). In wide band systems, which use "orthogonal frequency division multiplex" (OFDM), frequency diversity is often applied on the basis of channel coding combined with interleaving in the frequency domain. In (Todd et al., 1992; Corazza et al., 1996) the performance of frequency diversity at frequencies of 1.75 GHz…1.8 GHz has been evaluated in typical office buildings. At an availability of 99 %, the achieved diversity gains varied between 5 dB...9.6 dB for frequency separations larger than 5 MHz.
Having in mind the limitation of bandwidth and consumption of energy, spatial diversity is the more attractive strategy. However, frequency diversity is also considered a suitable instrument to compensate deep fading. Although it is proven, that optimum combining, using spatial diversity, may increase the signal to noise plus interference ration (SINR) in order to mitigate co-channel interferences (Winters, 1984), the application of frequency diversity is more effective and less complex.

4. Current Wireless Base Technologies and its Utilisation in Factory Automation

As already mentioned, most of the industrial wireless solutions use the unlicensed 2.45 GHz ISM frequency band. This section gives an overview of the regulation and the most important technologies operating in this frequency range.

4.1 Regulation for the 2.4 GHz ISM Frequency Band

Within the scope of the regulation 5.138 and 5.150 of the international telecommunication union, radiocommunication sector (ITU-R), besides others, the frequency range from 2.4 GHz to 2.5 GHz is enabled for industrial, scientific, and medical (ISM) applications. The European norm EN 300 328 (ETSI 2006) regulates the frequency range from 2.4 GHz to 2.4835 GHz for general utilisation in Europe. The maximal EIRP transmit power is limited to 100 mW. For devices, that do not use the modulation of "frequency hopping spread spectrum" (FHSS), the maximal spectral EIRP power density is further limited to 10 mW/MHz. There are no restrictions concerning the duty cycle of the radios. Depending on the application domain and the country, transmit powers above 10 mW have to be registered. In gen-

eral, there are country specific limitations to the utilisation of the 2.45 GHz ISM band (i.e. Spain and France).
In North America, the utilisation of unlicensed frequency bands is ruled by the Federal Communications Commission (FCC 2007) in the document CFR 47, Part 15. The maximal transmit power for the 2.45 GHz band is limited to 1 W for systems using FHSS over more than 75 frequency channels. For systems with less than 75 channels, the maximal transmit power is limited to 125 mW. In addition to that, a spectral power density of 8 dBm/3 kHz must not be exceeded.

4.2 Wireless Local Area Networks - IEEE 802.11

The most popular radio technologies operating within the 2.45 GHz band are compliant to the standards of IEEE 802.11b and IEEE 802.11g. Both standards specify 13 channels with spacing of 5 MHz for Europe and 11 for North America.

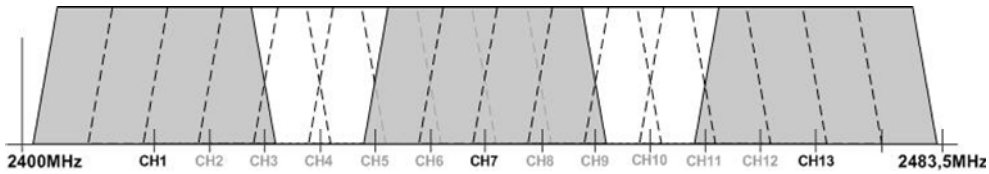


Fig. 4. IEEE 802.11 defines 13 channels for Europe and 14 Channels for North America.

With a transmit bandwidth of about 20 MHz, three non overlapping channels with a spacing of 30 MHz are available. The maximal transmit power is limited to 100 mW.
IEEE 802.11b supports data rates of 1 Mbps…11 Mbps. According to the selected data rates, the modulations of "differential binary phase shift keying" (DBPSK), „differential quadrature phase shift keying" (DQPSK) or, „complementary code keying" (CCK) are used. "Direct sequence spread spectrum" (DSSS) is used as a spreading technique. The amendment of IEEE 802.11g is an extension and supports data rates of up to 54 Mbps by introducing "orthogonal frequency division multiplex" (OFDM) with 52 sub-carriers as a spreading technique. These sub-carriers are either modulated using „binary phase shift keying" (BPSK), „quadrature phase shift keying (QPSK), „16- or 64-quadrature amplitude modulation" (16-QAM, 64-QAM) depending on the selected data rates. Furthermore this standard supports forward error correction (FEC) with coding rates of 1/2, 2/3, or 3/4. As the channel access method, both standards use "carrier sense multiple access/collision avoidance", which is based on a "clear channel assessment" (CCA) module. Prior to any transmission, the CCA module validates the occupation of the medium. If the medium is classified "busy", the transmit operation is interrupted for a pseudo random period of time and the channel is validated again. A prioritised medium access, comprising eight priority levels, was introduced by the extension of IEEE 802.11e. In order to classify the medium, three modes are specified and one of them must at least be supported. In mode 1 the medium is considered busy, as soon as the detected energy is above a predefined threshold. In mode 2 the medium is considered busy, if an IEEE 802.11 modulated signal is detected. In mode 3 the medium is considered busy, if an IEEE 802.11 modulated signal is detected and its energy is above a predefined threshold. In general, the end-user has no access to the configuration of the CCA mode.

In automation applications IEEE 802.11 is recommended by the PROFIBUS & PROFINET International (PI) as a wireless communication system for connecting PLCs and decentralised peripherals. With adapted IEEE 802.11 systems, PROFINET-I/O communications with update times of up to 8 ms can be served. Common use cases are forklift trucks and automated guided vehicles. In mobile scenarios the transition from one cell to another (roaming) is extremely critical. Currently, roaming times of < 50 % can be realised.
The next Amendment of the task group IEEE 802.11n is shortly before being published. This standard specifies either channels with 20 MHz bandwidth and 56 OFDM sub-carriers and channels with 40 MHz bandwidth and 112 sub-carriers within the frequency bands of 2.45 GHz and 5 GHz. By applying performance enhancing techniques like "MIMO", "Channel Bonding", "Frame Aggregation", "Spatial Multiplexing", and "Beam forming", data rates of 300 Mbps and beyond can be achieved. At the moment the draft standard, revision 8, is available (LAN/MAN Standards Committee of the IEEE Computer Society, 2008). The release of the final standard is expected in late 2009. Similar to the standards IEEE 802.11b and IEEE 802.11g a fast market penetration can be expected for the standard IEEE 802.11n, as well.

4.3 Bluetooth – IEEE 802.15.1

The latest specification of Bluetooth version 3.0 (Bluetooth Special Interest Group – SIG, 2009) was published in 2009. The PHY and MAC layer of the Bluetooth version 1.1 are published as the standard IEEE 802.15.1, as well. In its classical form 79 channels, with a spacing of 1 MHz, are specified in the range of 2.402 GHz…2.480 GHz. The radio signals are modulated using "Gaussian frequency shift keying" (GFSK, 1 Mbps), "π/4 differential quaternary phase shift keying" (π/4-DQPSK, 2 Mbps), or "8-ary differential encoded phase shift keying" (8DPSK, 3 Mbps). Bluetooth uses "Time Division Multiple Access" (TDMA) as the channel access method and FHSS for spreading. Three device classes with transmit powers of 1 mW, 2.5 mW and 100 mW are defined.
Bluetooth networks, called piconets, are formed in star topology. A piconet consists of a master and up to seven active slaves. In order to communicate, timeslots with a length of 625 µs are predefined. The specification defines synchronous connections (SCO) for the transmission of i.e. speech and asynchronous connections (ACL) for data transmission. Depending on the type, data packets occupy one to five timeslots and use "automated repeat requests" (ARQ) or FEC as channel coding. In each timeslot, or at leas after the transmission of a data packet, a change in frequency is performed respectively.

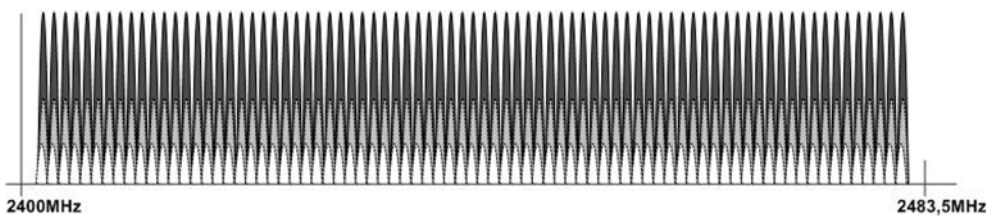


Fig. 5. IEEE 802.15.1 defines 79 Channels within the 2.45 GHz ISM Band.

In avoidance of coexistence problems, the standard supports an "adaptive power control" (APC) and "adaptive frequency hopping" (AFH). When using AFH, frequency channels

occupied by foreign radios are detected and excluded from the hopping scheme. With common Bluetooth transceiver chips a channel is classified busy, when the occupation is higher than 15 %. The adaption of the hopping scheme depends on the implementation and may take up to several seconds. In addition to the adaptive channel classification, frequency channels can be excluded of the hopping scheme manually, in order to avoid frequencies known to be in use by other radios. At least 20 channels have to be used. By doing so, a frequency separation to two coexisting IEEE 802.11 radios can be administered. Solely, the connection setup uses all frequencies. However, some vendors developed standard compliant solutions, which prevent interferences during the connection setup.
Bluetooth is applicable at control as well as sensor/actuator level. With respect to ABBs "Wireless interface for sensors and actuators" (WISA), the PROFIBUS & PROFINET International (PI) actually considers the PHY layer of Bluetooth as the basis for "Wireless Sensor/Actor Networks" (WSANs). A standard shall be published in 2010. A WISA network consists of a base station and up to 120 wireless I/O-concentrators and sensors/actuators in a star topology. The base station acts as the network coordinator and gateway to a super ordinate control system. The I/O-concentrators and sensors/actuators use IEEE 802.15.1 standard compliant transceivers. The base station consists of a special multi-transceiver architecture and thus able to serve multiple devices in parallel. The update time of 120 sensors is typically below 20 ms.
In version 3.0 of Bluetooth, the support of IEEE 802.11 as an "Alternate MAC PHY" (AMP) is introduced. In addition to that the "Bluetooth Low Energy" specification is to be published in late 2009. First transceivers for both technologies shall be available in 2010.

4.4 IEEE 802.15.4

The standard IEEE 802.15.4 specifies 16 channels with a separation of 5 MHz for the 2.45 GHz ISM band. With DSSS as spreading and "offset quadrature phase shift keying" (O-QPSK) as modulation, data rates of 250 kbps are supported. The standard limits the transmit power to 1 mW. However, the regulations allow the operation at transmit powers of up to 10 mW.
As channel access method CSMA/CA corresponding to IEEE 802.11 is utilised. Optionally, the standard supports a synchronised data communication in superframes of durations from 15 ms to 246 s. Each superframe consists of a "contention access period" (CAP) and a "contention free period" (CFP). During the CAP, devices willing to transmit, concurrently access the medium via CSMA/CA. The CFP consists of guaranteed timeslots and gives exclusive access to medium for higher prioritised transmissions. The standard was designed for low power industrial "wireless personal area networks" with low data rates.

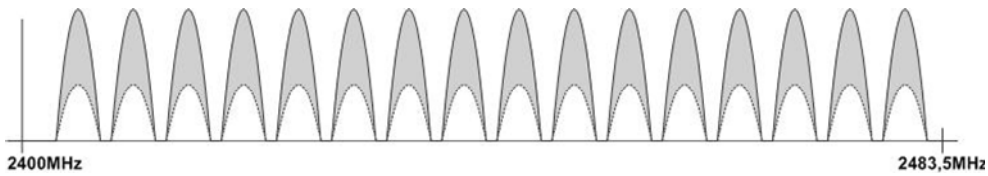


Fig. 6. IEEE 802.15.4 defines 16 Channels within the 2.45 GHz ISM Band.

The technology is wide spread in combination with the higher layers specified by ZigBee. ZigBee supports the operation of large multihop networks and addresses domains like home- and building automation, smart metering, and health care.
Within the scope of the HART 7 specifications, the first wireless standard for process automation, WirelessHART, was published in 2007. WirelessHART is based on the PHY layer of IEEE 802.15.4 and uses the "Time Synchronized Mesh Protocol" (TSMP) for channel access. In order to improve reliability, it is designed to support large multihop networks in full mesh topologies with a high degree of redundant paths. In avoidance of coexistence problems the standard changes frequencies at a rate of 10 ms. Optionally, a channel black list can be used to avoid frequencies currently in use. First products are successfully in use since late 2008.
At the moment "the International Society of Automation" (ISA) is shortly before publishing a second standard for the process automation, ISA 100.11a (ISA, 2009), based on the PHY layer of IEEE 802.15.4.
In the domain of factory automation a few proprietary solutions for the transmission of sensor data based on IEEE 802.15.4 are available.
Right now the task group of IEEE 802.15.4e is working on MAC layer extensions. In order to improve the support of time critical industrial applications, shorter transmit times, improved TDMA techniques and frequency hopping are evaluated. In the long run the extensions of IEEE 802.15.4e shall enable the standard to better support applications in factory automation.

4.5 Coexistence in the 2.4 GHz ISM Frequency Band

With the fast pace growth of wireless solutions, operating in the 2.45 GHz ISM band, in automation as well as the IT, the end-users demand for a good coexistence of the devices is getting obvious. In this respect a technologies coexistence properties depend on several parameters, like the transmit power, signal bandwidth, channel access methods, and duty-cycle, which often are vendor specific.
In IEEE 802.15.2 (LAN/MAN Standards Committee of the IEEE Computer Society, 2003) coexistence is defined as "a systems ability to perform a task in a shared medium, while other systems perform their tasks, complying with the same or a different set of rules". In a shared medium the main source of error is caused by interferences. Interferences appear, when signals overlay in the domains of time, frequency, and space. For the domain of frequency the IEEE Unapproved Draft Std P1900.2/D2.22 (LAN/MAN Standards Committee of the IEEE Computer Society, 2007b) further subdivides interferences into "In-Band", consisting of "Co-Channel-" and "Adjacent Channel- Interference", and "Out of Band", consisting of "Band Edge-" und "Far out of Band Interference". The most common form of appearance are "Co-Channel" interferences, which occur, when more than one system operates on the same frequency.

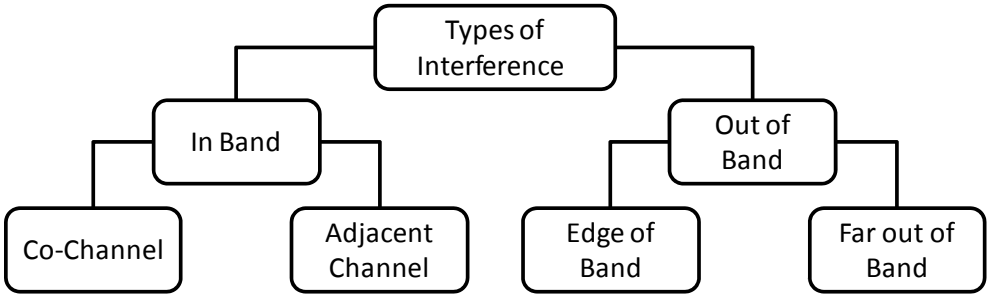


Fig. 7. Types of Interference defined by IEEE P1900.2/D2.22.

The domain of time is determined by the channel occupation in time, the duty cycle, of coexisting systems. The probability of signal interferences increases with the utilisation of the medium in time. The spatial domain is defined by the transmit power, the distance between the systems (Antennas), and the resulting "signal to interference plus noise ratio" (SINR). If the SINR is too low, a signal cannot be detected correctly at the receiver.
Besides these physical properties of interferences, channel access methods have a strong impact on the coexistence of radios. Typically, radio systems operating in the 2.45 GHz ISM band use either TDMA, CSMA/CA, or a mixture of both as access methods. TDMA subdivides the medium into timeslots, which are reserved for exclusive access to the medium. That way, TDMA systems support a deterministic behaviour in time and a good coexistence within the same network. In order to avoid interferences to foreign networks, TDMA is often used in combination with FHSS, additionally allowing to black list frequencies already in use by other systems (i.e. Bluetooth). When using CSMA/CA, the state of the medium is validated before any transmission of data and only performed, if the medium is classified idle. The validation of the medium is either based on an energy threshold, the detection of a valid carrier, or a mixture of both. On the one hand CSMA/CA is able to avoid interferences within the same or foreign networks. On the other hand CSMA/CA is vulnerable to jamming attacks and some kind of unnecessary interferences. Depending on the implementation, the following types of interferences may occur, when using CSMA/CA:

- Type-1: A weak signal, that would not induce interferences at the receiver, is detected at the transmitter, causes the medium to be classified busy, and thus delays the transmission ("Exposed Terminal Problem").
- Type-2: Interferences caused by multiple radios that access the medium at the same time.
- Type-3: The source of interference is out of the detection range of the transmitter, but causes interferences at the receiver ("Hidden Terminal Problem").

There are several strategies to mitigate these interferences within the same network of operation (Tsertou & Laurenson, 2008; Zhang et al., 2008). However, interferences with foreign networks may still appear.
How far interferences actually influence the coexistence properties of a system, always depends on the tasks to be performed. Usually, an underlying (wireless) communication system has a temporal reserve with respect to an application, in order to perform channel coding and retransmissions. If this reserve gets exhausted, the communication system cannot longer serve the application. It is obvious, that with increasing temporal requirements of an

application, the reserve of the communication system decreases and interferences result in application errors faster. Analytical as well as practical studies about the coexistence within the 2.45 GHz ISM band have been subject to several publications. For detailed information on this topic it is referred to (Arumugm et al., 2003; Chiasserini & Rao, 2003; Howitt & Gutierrez, 2003).
The previous descriptions stated the richness of technologies and applications operating in the 2.45 GHz ISM band. Thus, a coexisting operation of different wireless solutions is hardly avoidable. But it is very demanding to consider all parameter of relevance for the different domains of applications, when determining the properties of coexistence of radio technologies. In addition to that, comprehensive studies on the coexistence of new technologies, like IEEE 802.11n, WirelessHART, ISA 100.11a, and Bluetooth Low Energy have not been performed, yet.

Category	Class	Application	Description
Safety	0	Emergency action	(always critical)
Control	1	Closed loop regulatory control	(often critical)
	2	Closed loop supervisory control	(usually non-critical)
	3	Open loop control	(human in the loop)
Monitoring	4	Alerting	Short-term operational consequences (e.g. event-based maintenance)
	5	Logging and downloading/uploading	No immediate operational consequence (e.g., history collection, sequence-of-events, preventive maintenance)

Table 1. Application classes of ISA-SP100.

For that reason, a general process to establish a coexistence management for end user is described in (VDI, 2008). In relation to the application classes defined in (ISA, 2006), it is recommended to assign priorities to the different wireless solutions. The intensity for the frequency management shall be correlated to the assigned priority classes. The process comprises the whole plant location and shall include all persons responsible for planning, installing, and commissioning of wireless devices. Wireless applications either in automation, logistic, or IT have to be considered. The coexistence management is a cyclic process which comprises all stages of stock taking, planning, installation, commissioning, maintenance, operation, and documentation of wireless applications at a location. It is further recommended to involve qualified service providers and own personnel at early phases, in avoidance of malfunctions in the long run.

5. Upcoming Wireless Base Technologies

The development of wireless technologies and extended standards is fast pacing. Especially the progress with respect to ultra wideband (UWB) represents a great potential, to open up new domains of applications in factory automation. First efforts for a standardisation of UWB technologies were initiated by the IEEE 802.15 WPAN High Rate Alternative PHY Task Group 3a (TG3a), founded in 2001. The task groups aim was to develop a high speed

UWB technology, supporting data rates of > 100 Mbps at distances of < 10 m. Unfortunately, the group was not able to reach a consensus between two approaches offered by the leading industrial consortiums of the "WiMedia Alliance" and the "UWB Forum" and hence, disbanded in 2006. However, the approach of the WiMedia Alliance was published as the standard ECMA-368 in 2006 and is available in version 3.0 (Ecma International, 2008) since 2008. The standard uses "Multiband OFDM" (MB-OFDM) as modulation and supports data rates of up to 480 Mbps at distances of < 10 m. MB-OFDM is the basis of "Certified Wireless USB" (CW-USB). The application as an "Alternate MAC PHY" (AMP) is evaluated by the Bluetooth SIG. First transceiver chips and products are available since 2007. In 2007 the IEEE 802.15 WPAN Low Rate Alternative PHY Task Group 4a (TG4a) (LAN/MAN Standards Committee of the IEEE Computer Society, 2007c) published the second UWB standard. IEEE 802.15.4a is a low data rate UWB technology supporting data rates of 0.1 Mbps…27 Mbps. It targets industrial sensor networks with real-time location capabilities. First transceiver chips will be available in 2010.

5.1 Ultra Wideband

In principle UWB is an old technology, whose origins come from military applications of the USA, more than 40 years ago. Whilst back then UWB was used as a tap-proof radio communication, nowadays the applications aim at high speed data transfers and real-time location systems. The first regulation for UWB devices, published by the FCC in 2002 (FCC, 2007), defines UWB as follows. The relative bandwidth has to be larger than 20 % and the absolute bandwidth has to be at least 500 MHz at a 10 dB cut-off frequency. The regulation gives no restrictions concerning signal forming and modulation. Because of the dense occupied frequency spectrum, UWB follows the approach of a parallel utilisation of the spectrum with a large bandwidth and a low spectral density power. In doing so, UWB appears as noise to coexisting narrow band technologies.

Classically, UWB is based on "Impulse Radio" (Nekoogar, 2005), which transmits information via impulses in the baseband without modulation. The UWB spectrum is generated due to extreme short durations (< 1ns) of these impulses. Hence, UWB has the following inherent characteristics :

- Low latency times, due to extreme short symbol durations, what additionally offers the possibilities for precise ranging.
- Robust against the effects caused by multipath scattering. Reflexion and scattering are frequency selective. Using a high bandwidth reduces the probability of deep fading.
- Energy efficiency, due to the low spectral density power.
- Data rates of up to several Gbps.

Especially the first characteristics prove the potential of UWB for industrial communication systems. A typical use case would be a cable replacement for high speed real-time Ethernet field buses. Further use cases are WSANs. First studies on this issue have been published in (Paselli et al., 2008). A general overview of the potential use cases of UWB in industrial applications is given in (Hancke & Allen, 2006).

5.2 Regulation for Ultra Wideband

Since UWB uses frequencies, which are already in use by licensed radio applications, the regulations are relatively restricted in order to avoid interferences to these applications. The first regulation was published by the FCC Part 15 Subpart F in 2002. The document defines seven classes of UWB devices. The classes of importance for factory automation are "Indoor UWB systems" and "Hand held UWB systems". "Indoor UWB systems" may only be used inside buildings and must have a fixed indoor infrastructure (i.e. power supply). "Hand held UWB systems" may operate indoor or outdoor and must not have a fixed infrastructure. The frequency ranges and maximal allowed transmit powers are depicted in table 2.

Frequency [MHz]	**Max. EIRP [dBm/MHz]**	
	Indoor UWB systems	Hand held UWB systems
960 – 1.610	- 75.3	- 75.3
1.610 – 1.990	- 53.3	- 63.3
1.990 – 3.100	- 51.3	- 61.3
3.100 – 10.600	- 41.3	- 41.3
< 10.600	- 51.3	- 61.3

Table 2. Maximum EIRP spectral power densities for "Indoor UWB systems" and "Hand held systems" defined by FCC Part 15 Subpart F.

In further avoidance of interferences to GPS applications the maximal EIRP power density is limited to -83.3 dBm/kHz for the frequency ranges of 1.164 GHz…1.240 GHz and 1.559 MHz…1.610 MHz. Within a frequency spectrum of 50 MHz the maximal power is limited to 0 dBm. It is obvious that the actual range of operation is between 3.1 GHz…10.6 GHz.

The regulation for the frequency range from 3.1 GHz…10.6 GHz for harmonised utilisation of UWB systems in Europe was released in 2007 by the decision of the European commission (European Commission, 2007). The decision defines maximal EIRP power densities in dBm/MHz and within a spectrum of 50 MHz (comp. table 3). In addition to that, the decision differentiates between devices, which implement mitigation techniques in order to increase protection for radio Services. One mitigation technique is defined as „low duty cycle" (LDC). Devices implementing LDC must have a duty cycle lower than 0.5 % per hour and lower than 5 % per second. Furthermore, a single transmit duration must not exceed 5 ms. Another technique is "detect and avoid" (DAA). Devices implementing DAA shall observe the used frequency spectrum with respect to coexisting devices and must adapt their transmit behaviour to avoid interferences.

Frequency [GHz]	**Max. EIRP Power Density (dBm/MHz)**	**Max. EIRP Power Density (dBm/50 MHz)**
< 1.6	- 90.0	- 50.0
1.6 – 3.4	- 85.0	- 45.0
3.4 – 3.8	- 85.0	- -45.0
3.8 – 4.2	- 70.0	- 30.0

4.2 – 4.8	- 41.3 (- 70.0)	- 0.0 (- 30,0)[1]
4.8 – 6.0	- 70.0	- 30.0
6.0 – 8.5	- 41.3	- 0.0
8.5 – 10.6	- 65.0	- 25.0
> 10.6	- 85.0	- 45.0

Table 3. Maximum EIRP spectral power densities for Europe.

Table 2 shows, that the utilisation of frequencies below 6 GHz will be restricted to devices, implementing mitigation techniques. How far real-time applications in factory automation can be served, regarding these restrictions, has to be investigated.

6. Conclusion

Industrial environments are highly demanding for the utilisation of wireless communication systems. However, on the basis of suitable adaptations and performance enhancing strategies several applications in factory automation can already be served by radio solutions. The current state of the art reliably supports update times of about 10 ms on application layer. First standards for the domain of factory automation based on the PHY layer of Bluetooth can be expected in 2010. Due to the huge deployment of wireless technologies, using the 2.45 GHz ISM band, either in automation and IT, the problem of interferences, caused by coexisting devices, increases. In order to guarantee a reliable communication, even for time critical applications, a plant wide coexistence management is absolutely essential. By using other frequency ranges, the emerging UWB technologies give a great potential, to ease these coexistence problems. Furthermore they offer the possibility of addressing applications with temporal requirements of about 1 ms and below, because of their extreme short symbol durations. The research on UWB for industrial applications, especially factory automation, has just started. The upcoming years are going to reveal, whether UWB will enter into the domain of factory automation or not.

7. References

Arumugam, A.K.; Doufexi, A.; Nix, A. R.; Fletcher, P.N. (1003). An Investigation of the Coexistence of 802.11g WLAN and High Data Rate Bluetooth Enabled Consumer Electronic Devices in Indoor Home and Office Environments, *IEEE Trans. on Consumer Electronics*, vol. 49, no. 3, pp. 587–596, Aug. 2003.

AS-Interface (2009) (online). Website: http://www.as-interface.net, visited on May 2009

Bello, P. A. (1963). Characterization of Randomly Time-Variant Linear Channels. In: *IEEE Transactions on Communication Systems 11,* pp. 360–393, Dec. 1963.

Biglieri, E. (2005). *Coding for Wireless Channels*. New York: Springer, 2005.

Bluetooth Special Interest Group – SIG (2009): *Specification of the Bluetooth System*, Version 3.0. Bluetooth Special Interested Group, 2009.

Boelcskei, H. (2006). Mimo-ofdm wireless systems: Basics, perspectives and challenges. *IEEE Wireless Communications 13(4)*, pp. 31–37.

Clarke, R. H.(1969). A statistical theory of mobile radio reception, *The Bell System Technical Journal*, vol. 47, pp. 957–1000, Aug. 1969.

Chiasserini, C.-F.; Rao, R. R. (2003). Coexistence mechanisms for interference mitigation in the 2.4-ghz ism band, *IEEE Trans. on Wireless Communications*, vol. 2, no. 5, pp. 964–975, Sept. 2003.

Corazza, G.E.; Degli-Esposti, V.; Frullone, M.; Riva, G. (1996). A characterization of indoor space and frequency diversity by ray-tracing modeling, *IEEE Journal on Selected Areas in Communications*, Volume 14, Issue 3, Apr 1996 Page(s):411 - 419

Diggavi, S. N.; Al-Dhahir, N.; Stamoulis, A.; Calderbank, A. R. (2004). Great Expectations: The Value of Spatial Diversity in Wireless Networks, *Proceedings of the IEEE*, vol. 92, no. 2, pp. 219–270, Feb. 2004.

Ecma International (2008), Standard ECMA-368: *High Rate Ultra Wideband PHY and MAC Standard*, 3rd Edition.

ETSI (2006). *EN 300 328*, Electromagnetic compatibility and Radio spectrum Matters (ERM); Wideband transmission systems; Data transmission equipment operating in the 2,4 GHz ISM band and using wide band modulation techniques; Harmonized EN covering essential requirements under article 3.2 of the R&TTE Directive", V1.7.1.

European Commission (2007). *COMMISSION DECISION* of 21 February 2007 on allowing the use of the radio spectrum for equipment using ultra-wideband technology in a harmonised manner in the Community, document number C(2007) 522, 2007/131/EC.

Federal Communications Commission FCC (2007). 02 48A1 Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems, February 2002, revision of 2007

Goldsmith, A. (2005). *Wireless Communications*, Cambridge University Press, 40 West 20th Street, NY 10011-4211, 2005.

Haccoun, D.; Pierre, S. (1996). Automatic repeat request," *in The Communications Handbook*, J. D. Gibson, Ed. Boca Raton, Florida: CRC Press / IEEE Press, 1996, pp. 181–198.

Haehniche, J. ; Rauchhaupt, L. (2000). Radio Communication in Automation Systems: the R-Fieldbus Approach. In: *Proceedings of the IEEE Workshop on Factory Communication Systems (WFCS 2000)*, 2000, S. 319–326.

Haehniche, J. (2001). Radio based communication in automation – Overview of technologies (in german), Practical automation (in german), ATP 43 (2001), Jun., Nr. 6, S. 22–27.

Hancke, G.P.; Allen, B. (2006). Ultra wideband as an Industrial Wireless Solution, *IEEE Pervasive Computing*, Vol. 5, Issue 4, pp. 78 – 85, Oct.-Dec. 2006

HART Communication Foundation (2008), *HART Field Communication Protocol Specifications*, Revision 7.2, 2008

Hashemi, H. (1993). The Indoor Radio Propagation Channel. In: *IEEE Transactions on Communications 81* (1993), Mai, Nr. 7, S. 943–968

Hoeing, M.; Helmig, K.; Meier, U. (2006): Analysis on the interference immunity and communication reliability of the Bluetooth technology using the example of an industrial sensor/actor network (in german), In: *VDI Progress Reports (in german)*, Bd. 10, Nr. 772, 2006, S. 155–164

Howitt, I.; Gutierrez, J. A. (2003). IEEE 802.15.4 low rate – wireless personal area network coexistence issues, *in Proc. Wireless Communications and Networking Conference 2003 (WCNC 2003)*, New Orleans, Louisiana, Mar. 2003, pp. 1481–1486.

IEC (2007a), Geneva. Industrial communication networks – Fieldbus Specifications. IEC 61158 Ed. 4.0.

IEC (2007b), Geneva. Industrial communication networks - Profiles - Part 1: Fieldbus profiles. IEC 61784-1 Ed. 2.0.

IEC (2007c), Geneva. Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3. IEC 61784-2 Ed. 1.0.

IO-Link (2009) (online). Website: http://www.io-link.com, visited on May 2009

ISA (2009) SP100.11a Working Group for Wireless Industrial Automation Networks: "Wireless systems for industrial automation: Process control and related applications"

ISA (2006) SP100.11, Call for Proposal, Wireless for Industrial Process Measurement and Control.

Kramer, G.; Gastpar, M.; Gupta, P. (2005). Cooperative strategies and capacity theorems for relay networks. *IEEE Transactions on Information Theory 51(9)*, 3037–3063.

Konnex Association, *Volume 3: System Specification*," Version 1.3, 2006

LAN/MAN Standards Committee of the IEEE Computer Society (2003). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area net-works – Specific requirements – Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands.*

LAN/MAN Standards Committee of the IEEE Computer Society (2005). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area net-works – Specific requirements – Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).*

LAN/MAN Standards Committee of the IEEE Computer Society (2006). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area net-works – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs).* revision of 2006.

LAN/MAN Standards Committee of the IEEE Computer Society (2007a). *Information technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*

LAN/MAN Standards Committee of the IEEE Computer Society (2007b). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area net-works – Specific requirements – Part 19: Unapproved IEEE Draft Recommended Practice for Interference and Coexistence Analysis*, IEEE Unapproved Draft Std P1900.2/D2.22.

LAN/MAN Standards Committee of the IEEE Computer Society (2007c). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (LR-WPANs),* Amendment 1: Add Alternate PHY

LAN/MAN Standards Committee of the IEEE Computer Society (2008). *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11n: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.* Amendment 4: Enhancements for Higher Throughput, IEEE Draft STANDARD, revision of 2008

Liu, H.; Ma, H.; Zarki, M. E.; Gupta, S. (1997). Error control schemes for networks: An overview, *MONET – Mobile Networks and Applications*, vol. 2, no. 2, pp. 167–182, 1997.

Laneman, J. N.; Tse; D.; Wornell, G. W. (2004). Cooperative diversity in wireless networks: Efficient protocols and outage behaviour. *IEEE Transactions on Information Theory 50(12)*, 3062–3080.

Nekoogar, F. (2005). *Ultra-Wideband Communications-Fundamentals and Applications,* Prentice Hall Communications Engineering and Emerging Technologies Series. ISBN: 0-13-146326-8, 2005

Paetzold, M. (1999). *Mobile radio channels (in german),* Wiesbaden : Vieweg Verlag, 1999

Paselli, M.; Petre, F.; Rousseaux, O.; Meynants, G.; Engels, M.; Benini, L.; Gyselinckx, B. (2008). A High-Performance Wireless Sensor Node for Industrial Control Applications, *Third International Conference on Systems*, ICONS 2008, pp. 235 – 240, 13-18 April 2008

Paulraj, A. J.; Gore, D. A.; Nabar, R. U.; Boelcskei, H. (2004). An Overview of MIMO Communications – A Key to Gigabit Wireless. *Proceedings of the IEEE 92(2)*, 198–218.

Pellegrini, F. D. ; Miorandi, D. ; Vitturi, S.; Zanella, A. (2006). On the Use of Wireless Networks at Low Level of Factory Automation Systems, *IEEE Trans. on Industrial Informatics*, vol. 2, no. 2, pp. 129–143, May 2006.

Rappaport, T. S. (2002): Wireless Communications - Principles and Practice. Upper Saddle River, NJ 07458 : Prentice Hall PTR, 2002

Rappaport, T. S.; Mcgillem, C. D. (1989): UHF Fading in Factories. In: IEEE Journal on Selected Areas in Communication 7 (1989), Jan., Nr. 1, S. 40–48

Rappaport, T. S. (1989a): Indoor Radio Communications for Factories of the Future. In: IEEE Communication Magazine 27 (1989), Mai, S. 15–24

Rappaport, T. S. (1989b): Characterization of UHF Multipath Radio Channels in Factory Buildings. In: IEEE Transactions on Antennas and Propagation 37 (1989), Aug., Nr. 8, S.1058–1069

Scheible, G.; Dacfey Dzung; Endresen, J.; Frey, J.-E. (2007). Unplugged but connected - Design and Implementation of a Truly Wireless Real-Time Sensor/Actuator Interface, *Industrial Electronics Magazine*, IEEE, Volume: 1, Is-sue: 2, pp 25-34, 2007

Todd, S.; El-Tanny, M.; Mahmoud, S. (1992). Space and Frequency Diversity Measurements of the 1.7GHz Indoor Radio Channel Using a Four-Branch Receiver, *IEEE Transactions on Vehicular Technology*, Vol. 41, No 3. August 1992

Tsertou, A.; Laurenson, D. (2008). *Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network,* Mobile Computing, IEEE Transactions on, Volume: 7, Issue: 7, pp. 817-831, 2008

VDI (2008) FA 5.21, *Draft VDI/VDE-Directive 2185* : Wireless Communication in the Automation Technology - coexistence management of wireless solutions (in german), Beuth Verlag, Berlin.

Vedral, A.; Wollert, J. F.; Buda, A.; Altrock, R. (2006). The Capability of Bluetooth for Real-Time Transmission in Automation, *in Proceedings of the IASTED Network and Communication Systems (NCS 2006)*, March 2006, pp. 168–175.

Vedral, A.; Wollert, J. F. (2006). Analysis of Error and Time Behavior of the IEEE 802.15.4 PHY-Layer in an Industrial Environment, *in Proceedings of the IEEE Workshop on Factory Communication Systems (WFCS 2006)*, Jun. 2006, pp. 119–124.

Vedral, A. (2007). Digital Analysis, Performance Evaluation and Generative Modelling of WPAN Connections under Industrial Propagation Conditions (in german), PhD thesis, Technical University of Brandenburg in Cottbus, 2007.

Vedral, A.; Kruse, T.; Wollert, J. F. (2007). Development and performance evaluation of an antenna diversity module for industrial communication based on IEEE 802.15.4, *in the proceedings of 12th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA 2007)*, pp. 177 – 186, Sept. 2007.

Willig, A.; Matheus, K.; Wolisz, A. (2005). Wireless Technology in Industrial Networks, *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, 2005.

Willig, A. (2008). How to exploit spatial diversity in wireless industrial networks, *Fieldbuses and Networks in Industrial and Embedded Systems*, Volume #7, Part#1, 2008.

Winters, J.H. (1984). Optimum Combining in Digital Mobile Radio with Cochannel Interference. *IEEE Journal on Selected Areas in Communications*, 2(4):528–539, July 1984.

Zepernick, H.-J., Wysocki, T.A. (1999). Multipath channel parameters for the indoor radio at 2.4 GHz ISM band, *Vehicular Technology Conference*, 1999 IEEE 49th , vol.1, no.pp.190-193 vol.1, July 1999.

ZigBee Standards Organization (2007), ZigBee Specification, 2007.

Zhang, K.; Zhang, D.; Jiang, W. (2008). Mitigation of Exposed Terminal Problem Using Packet Sensing, *CNSR*, pp 263-269, 2008

3

A Real-time Wireless Communication System based on 802.11 MAC

Gennaro Boggia, Pietro Camarda, L. Alfredo Grieco
and Giammarco Zacheo
DEE – Politecnico di Bari
Italy

1. Introduction

IEEE 802.11 Wireless Local Area Networks (WLANs) are very pervasively deployed in the consumer market, due to their ability to provide ubiquitous network access with high flexibility, cheap costs, and ease of installation and maintenance. In such networks, the wireless channel is shared among the stations and, in order to deal with packet collisions, a Carrier Sense Multiple Access with Collision Avoidance algorithm is employed. The scientific community is now investigating in which measure this technology can be exploited also in real-time contexts, where several tasks have to be served with a high degree of determinism to provide the expected service, such as in Networked Control Systems (NCSs). This issue is challenging because the network interface cards available on the market have been mainly conceived to provide a best effort service, without any guarantees on packet delay. Moreover, the unpredictable behavior of the radio channel further worsen the problem.

NCSs exploit a packet switching network to connect spatially distributed sensors, actuators, and controllers (Hespana et al., 2007). In this way, a NCS can accomplish complex control tasks without requiring cumbersome wiring infrastructures. In fact, point-to-point interconnections are replaced by a communication network, which is shared among all the components of the control system using statistical multiplexing. A key issue of NCSs is that the Quality of Control of the system is influenced by the Quality of Service of the underlying communication system (Buttazzo et al., 2007; Baillieul & Antsakls, 2007). To solve the problem three different class of techniques can be jointly adopted:

1. an advanced control design to counteract time-varying packet delays and losses induced by network (Schenato et al., 2007; Nair et al., 2007; Liu et al., 2007; Wu & Chen, 2007);
2. the use of Real Time Operating Systems (RTOSs) at NCS nodes to timely serve events scheduled by control applications (Baillieul & Antsakls, 2007, Kim et al., 2006);
3. the reduction as much as possible of network delays and packet losses (Hespana et al., 2007; Boggia et al., 2008a; Boggia et al., 2008b).

Until now, these topics have been mainly investigated with reference to wired NCSs. But, the trend is toward wireless communications because they can further reduce wiring and they can increase the flexibility of a NCS, giving the chance to build, on the fly, control

systems made be sensors, actuators and controllers placed in the same area (Baillieul & Antsaklis, 2007; Moyne & Tilbury, 2007; Burda & Wietfeld, 2007). Anyway, it is not straightforward to build a wireless networked control system (WNCS) due to the unpredictable behavior of the radio channel (Walke et al., 2006; Cena et al., 2007). Starting from this premise, the present chapter focuses on three main objectives:

i. to analyze the state of the art on wireless real-time systems;
ii. to experimentally evaluate the performance bounds of a 802.11-based wireless real-time communication platform;
iii. to provide guidelines to design an effective TDMA (Time Division Multiple Access) strategy for wireless real-time systems starting from the so derived performance bounds and integrating the leading IEEE 802.11 technology (Walke et al., 2006), the RTnet framework (Kiszka et al., 2005a), and the Xenomay nanokernel (Gerum, 2004).

The rest of the chapter is organized as follows: Sec. 2 summarizes related works on real-time wireless communications technologies. Sec. 3 provides an overview on 802.11 WLANs and RTOSs, as basic elements to realize a wireless real-time system. In Sec. 4, the architecture of a WNCS is described and its performances are evaluated in Sec. 5. In Sec. 6, the design guidelines for a TDMA scheme based on the considered architecture are given. In Sec. 7, the performance of the proposed TDMA scheme are experimentally evaluated. Finally, the last section outlines the conclusions.

2. State of the art on wireless real-time systems

In recent years, research activities in the field of wireless real-time technologies for NCSs have been very active as testified by the relevant amount of literature produced on this subject, by the number of developed simulation/experimentation platforms (Andersson et al., 2005; Cervin et al. 2007; Hasan et al., 2007; Nethi et al., 2007; Biasi et al., 2008; Chen et al., 2008), and by the already available communication technologies (Neumann, 2007; Pellegrini et al., 2006; Willig et al., 2005) for WNCSs.

Herein, it follows a review of the most important contributions that are related to our discussion. In particular, we will focus on the most recent ones, leaving the reader to consult the excellent surveys (Neumann, 2007; Pellegrini et al., 2006; Willig et al., 2005) to gain a full vision of the world of wireless real-time networks.

In (Flammini et al., 2009), a novel wireless real-time communication protocol has been designed and experimentally evaluated. It exploits standard hardware and implements a hybrid medium access strategy. Time Division Multiple Access scheduling is used to ensure time deadlines respect, while Carrier Sense Multiple Access with Collision Avoidance is used for acyclic communications. It has been successfully tested in a prototype network that adopts star topology and can manage up to 16 nodes with a refresh time of 128 ms.

In (Heynicke et al., 2008; Krber et al., 2007), a gateway to interconnect hybrid wireless/wired control networks is proposed. The gateway is based on standard equipments such as the Chipcom CC2400 device by Texas Instruments. Its effectiveness has been demonstrated in an experimental testbed made by 32 nodes handled using four frequency channels and eight time-slots per channel.

In (Boughanmi et al., 2008), the suitability of IEEE 802.15.4 Wireless Personal Area Networks (WPANs) (IEEE, 2006) for wireless networked control systems has been investigated. In particular, using the TrueTime Matlab/Simulink simulator (Cervin et al., 2007), it has been

shown that the joint adoption of the beacon-enabled mode and of the Guaranteed Time Slot mechanism can allow the support up to two control loops with sampling periods not smaller than 15.36 ms. Analogously, in a less recent work (Choi et al., 2006), a wireless real-time network based on the 802.15.4 MAC has been designed, which is able to satisfy deadlines not smaller than 100 ms.

In Sep. 2007, the WirelessHART standard has been issued (Song et al., 2008) with the objective to support process measurement and control applications. WirelessHART is a secure, low-speed, if compared to 802.11g WLANs (IEEE, 1999a), and TDMA-based wireless mesh networking technology. It uses a central network manager to pro-vide routing and communication schedules. At the very bottom, it adopts the IEEE 802.15.4 physical layer and operates in the 2.4 GHz ISM radio band using 15 different channels (Biasi et al., 2008). WirelessHART appears a promising technology in this field and research activities are on going to assess its performance bounds (Biasi et al., 2008).

In (Lee et al., 2008), in order to improve the real-time performance and reduce the transmission delay of IEEE 802.11b WLANs, a four-layer architecture has been proposed and experimentally tested, based on the network driver interface specification (NDIS) (Floroiu et al., 2001) combined with a virtual scheduling algorithm that avoids collisions. In a network scenario with nine nodes, it has been shown that the architecture is able to provide an upper bound on packet delay comprised between 10 ms and 20 ms, depending on the network conditions.

In (Robinson & Kumar, 2007), the problem of selecting what information should be sent between a sensor and a controller in a networked control system where the two components are separated by an unreliable, bandwidth limited communication link, such as a wireless one, has been analyzed. It has been shown that the common practice of sending the most recent observation is not optimal. Moreover, necessary and sufficient conditions for the existence of a combination of past and present measurements that minimizes the state error covariance have been derived. These results could have serious implications in the design of future generation highlevel protocols that modify the contents of packets waiting to be sent by taking into account the status of the previous transmissions.

In (Baliga & Kumar, 2005), the focus has been moved on the issue of middleware for networked control systems which feature the convergence of control with communication and computation. In particular, it has been shown that a software architecture able to integrate the heterogeneous technologies that compose a complex NCS is required. Moreover, the Etherware middleware is proposed and experimentally tested using a vehicular control testbed.

In (Rauchhaupt, 2002), the R-FIELDBUS project, supported by the European Commission in the 5th FP, is described. It is aimed at the implementation of a wireless fieldbus based on the architecture of Profibus DP (Pellegrini et al., 2006). An important result of the R-FIELDBUS project is the accurate investigation carried out on the available radio technologies. As a result, the IEEE 802.11b physical layer, using direct sequence spread spectrum (DSSS) modulation, was selected as the most suitable for industrial applications. Moreover, the adoption of the IEEE 802.11 MAC layer was not recommended because of the randomness possibly introduced in the packet delay. For such a reason, the R-FIELDBUS makes use of the Profibus data link layer.

3. Basic elements for a 802.11-based wireless real-time platform

3.1 The 802.11 wireless LANs

The widespread deployment of IEEE 802.11 WLANs is mainly due to their easy installation, flexibility and robustness against failures (IEEE, 1999a; Varshney, 2003). They allow the transmission at a data rate up to 54 Mbps. The 802.11 Medium Access Control (MAC) employs a mandatory contention-based channel access scheme called Distributed Coordination Function (DCF), based on the Carrier Sense Multiple Access with Collision Avoidance mechanism (Walke et al., 2006).

The fundamental building block of 802.11 architecture is the Basic Service Set (BSS), composed by a group of stations, in the same geographical area, that access the radio channel under the control of DCF. The standard defines different topological configurations, but here we will consider only the Independent BSS, which allows direct communications among stations in the same area (i.e., an ad-hoc network architecture).

Using DCF, for each frame, a station listens to the channel before transmitting. If the channel is sensed idle for a minimum interval time called DCF Interframe Space (DIFS), then the station transmits immediately the frame. Otherwise, if the medium is sensed busy, the transmission attempt is deferred until the expiration of a backoff time. Such a time is a multiple of a *slot time* (depending on the physical layer implementation), where the multiple is a random integer uniformly distributed in the interval [0, CW] and CW is the so called *Contention Window*. The *CW* size is subject to minimum and maximum bounds, CW_{min} and CW_{max}, respectively (Walke et al., 2006).

Each correctly received frame is acknowledged by an ACK frame, that is sent after a Short Interframe Space period (shorter than DIFS), to avoid that other stations use the channel. If the transmission is not successful (i.e., no ACK frame is received by the sending station), the listen-before-talking protocol and the backoff procedure are repeated by doubling the CW value up to the maximum limit of 1023 time slots.

The Collision Avoidance is obtained by the Virtual Carrier Sensing: in the header of each delivered frame there is a duration field, which indicates the time required by a transmission. The duration field is used by each station in the BSS to update the Network Allocation Vector, that accounts for the duration of the current transmission after which the channel can be sensed again for the access (Walke et al., 2006).

When a station senses a busy channel during the backoff period, its timer is frozen and the countdown is resumed after the channel is sensed idle again for a DIFS interval. This mechanism allows stations with larger backoff periods to access the channel in the next contention period. Consequently, the bandwidth allocation is more fair (Walke et al., 2006).

3.2 Background on Real-time Operating Systems

Real time operating systems (RTOSs) play a major role in NCSs. In fact, they timely serve events scheduled by control applications. Herein, we describe some basic principles about RTOSs that are relevant for our framework. A RTOS is a multitasking Operating System (OS) able to guarantee its services with deterministic response times. Several commercial (Barr, 2003) and open source (Massa, 2002) RTOSs are available, some of which are widely used and supported (Massa, 2002). In particular, in the open source community, there was a lot of effort to give hard real-time capabilities to the Linux kernel. Various projects are very active in this field and widely used in academic and industrial environments: among other

solutions, there are the RT PREEMPT kernel patch (McKenney, 2005), which modifies the preemption mechanisms and interrupt handlers inside the kernel to achieve soft real-time requirements, and the RTAI (DIAPM, 2008) and Xenomai (Gerum, 2004) projects, which add a high priority scheduler for real-time tasks running concurrently with the Linux kernel with minimal intervention on the kernel itself. While the former approach allows seamless integration of the real-time tasks using standard APIs (Application Programming Interfaces), it can generally guarantee only soft real-time requirements, given that the real-time task scheduling can be influenced by system events. On the other hand, co-kernel scheduling performance is less influenced by external events, thus satisfying sharper real-time requirements.

3.3 Xenomai

Xenomai is based on the Adeos framework, which is able to provide a flexible environment for sharing hardware resources among multiple operating systems, or among multiple instances of a single OS (Yaghmour, 2001). In order to make multiple kernels run safely in parallel on the same platform, Adeos provides an efficient share to critical hardware resources, giving the opportunity to set priorities to each domain, i.e., each OS running on the machine. The original framework was intended to be a layer for machine virtualization and distributed computing environments. With its simplified version, known as I-Pipe (Interrupt Pipeline), the design has been changed to meet deterministic latencies in interrupt handling, which is fundamental for real-time operations.

Xenomai includes a core components which provides base funcions for realtime operations, which acts as a nanokernel, that is, it relies on the Linux kernel for everything it is not able to do by itself. The main components and functionalities of Xenomai are decribed below.

- **Nucleus**: it is the core of the RTOS, being responsible for thread creation, scheduling, synchronization, memory allocation, and interrupt management (Silberschatz et al., 2004).
- **RTDM**: The Real-Time Driver Model (RTDM) is a framework for real-time driver design, including file and socket I/O and descriptor handling (Kiszka, 2005b).
- **High resolution timers**: Another important feature of Xenomai is the support for high resolution timers, which provide microsecond precision for event scheduling.
- **Skins**: Emulation layers for existing RTOs (VxWorks, pSOS+, VRTX, RTAI, uITRON) (Barr, 2003).

The Native skin of Xenomai is an API with the following main characteristics:

- **Real-time IPC**: a complete set of Inter-Process Communication (IPC) primitives is available; among others, it includes message queues, one-to-one message passing and pipes, the latter providing a latency-free communication channel between real-time tasks and standard non real-time processes (ANSI, 1994).
- **Synchronization**: most of the classical synchronization objects are implemented within the Native skin, such as Mutexes, Counting Semaphores, Event flags and Condition Variables (Silberschatz et al., 2004).
- **Memory**: when dynamic memory allocation is needed by real-time tasks, Xenomai uses a region of memory whose size is selected at startup. The allocation occurs in a time-bounded fashion, and it can be used to map shared memory objects between kernel and user space.

3.4 RTnet, a hard real-time networking framework

RTnet is a hard real-time network protocol stack built on top of Xenomai and RTAI (real-time Linux extensions) (Kiszka, 2005a). It uses standard Ethernet hardware; in particular, it supports several popular network interface card chip sets, including Gigabit Ethernet and an experimental support for IEEE 802.11 WLAN. Currently, only wireless Ralink RT2500 chipset (Ralink, 2006) is supported in RTnet.

RTnet provides a POSIX socket API to real-time user space processes and kernel modules (via the RTDM interface). The main components of RTnet are the stack manager, which deals with the management of incoming packets, and the drivers, which communicate with the hardware. The stack manager is a real-time task with the highest priority: it demultiplexes the incoming packets passing their references to the respective handlers, according to the protocol which the packet belongs to.

The stack manager and the device drivers use a structure named *rtskb* to exchange and store temporarily the packets. Its use is similar to the *sk_buff* structure used in the Linux networking stack (Benvenuti, 2006). The buffer size is statically set to the network MTU (Maximum Transmission Unit), and a fixed number of buffers is allocated and assigned to each component at startup. Each component receiving a *rtskb* filled with a packet gives back an empty one, otherwise the packet is held back. Thus, at the end of each transaction each component holds the same number of *rtskb*.

Other modifications have been done in the UDP/IP implementation, in the routing, and in the IP fragmentation, which are not covered here not being related to our discussion.

RTnet has also an intermediate software layer between UDP/IP and drivers, called RTmac. Such a layer is not mandatory, but it is used for some useful services that RTnet can offer, e.g., automatic configuration (via the RTcfg module) or non real-time traffic tunneling by using virtual interfaces. Moreover, a medium access mechanism inside the RTmac can be defined, to guarantee collision-free and time-bounded transmissions. Those mechanisms are called disciplines, and can be dynamically loaded and unloaded. RTnet comes with a flexible TDMA scheme with a static master/slave approach. It provides clock synchronization, multiple slot assignments, and static transmission priorities. It is designed for wired operation and includes a fine-grained calibration mechanism which mitigates the negative effects of the transmission jitter.

The TDMA mechanism used by RTnet is based on a master-slave approach (Kiszka, 2005a). The master station manages the synchronization, and other stations can act as backup masters in order to compensate a possible master failure. The master station sends regularly a *synchronization* packet, which marks the starting of a transmission TDMA frame. Every frame is divided into transmission timeslots, and it has a numerical identifier which is incremented when a new frame starts. Each station may have one or more slots assigned for transmission, with a separate packet queue for each timeslot. It is also possible to assign multiple slots to each output queue. Each timeslot can be shared between multiple station, which can use it in turn. In Fig. 1 there is a sample configuration for a TDMA cycle. The master station sets the cycle time, while slave stations must configure their transmission slots accordingly. The configuration may be present on each station as a text file or may be distributed by the master station to all slave participants before starting the calibration phase using the RTnet configuration service, called RTcfg. A timeslot assigned to a transmitting station should be specified as an offset relative to the start of the TDMA frame (that is, with respect to the transmission of the synchronisation packet by the master).

When a slave station joins the network, it begins a synchronisation procedure with the master station, which is repeated several times in order to estimate the average transmission delay. The number of repetitions depends on the expected variance of the measurements and has to be chosen appropriately.

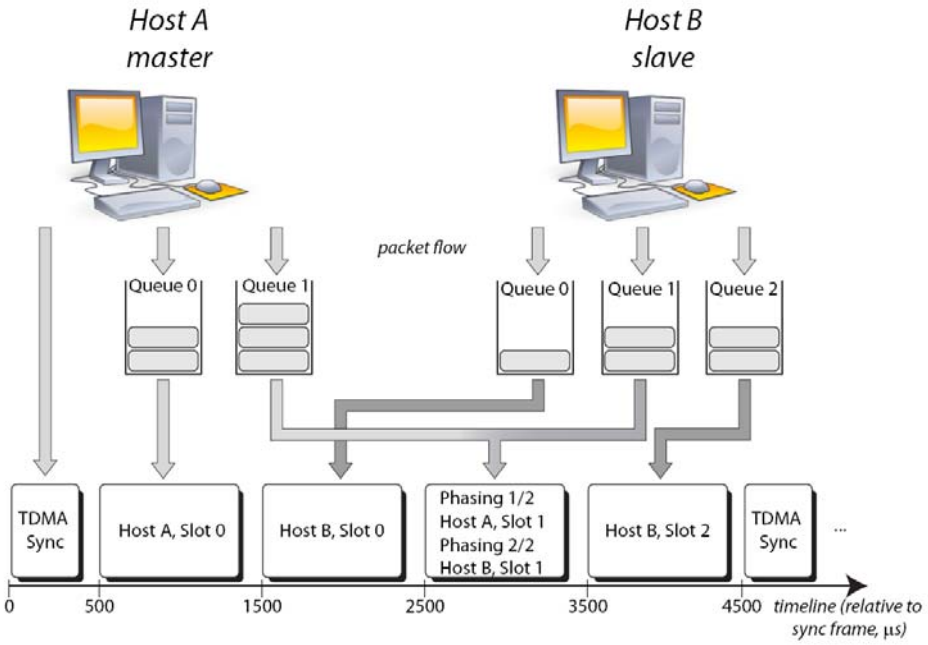


Fig. 1. Example of configuration for the structure of the TDMA frame in RTnet.

When working with wireless devices, most of the 802.11 mechanisms for collision avoidance are not used in RTnet in order to achieve a more deterministic behavior. The bit rate is statically assigned and no rate adaptive mechanisms are provided in the real-time driver. The Minimum and Maximum values for the 802.11 Contention Window can be altered on a per-packet basis or globally; also, the values for interframe spacings and the *slot time* can be altered with respect to the ones provided by the standard (IEEE, 1999a; IEEE, 1999b; IEEE, 1999c; IEEE, 2003; IEEE, 2005). The 802.11 acknowledgment mechanism can be used for signaling a successful frame delivery (ack mode), or can be completely disabled (raw mode); if the acknowledgement frame is not received the adapter considers the sent frame lost and may decide to retransmit it. The maximum of retries can be chosen by the user. It is possible to filter incoming broadcast and multicast packets, in order to reduce processing overhead for unwanted information. The last parameter that can be chosen is the baseband processor sensitivity, S_{BP}, from which the power threshold level for the receiver is computed. This threshold value is used to detect activity on the radio channel and is computed by adding an offset to the S_{BP} value. Thus, a low value for the sensitivity parameter means a low threshold, so that the receiver will be more selective during the channel sensing (this increases the probability of waiting for the channel to become idle). On the other hand, if the

sensitivity value is too high, the receiver would signal that the channel is idle even if there are interferences or too much noise (thus increasing the probability of transmission errors).

4. The Architecture for a Wireless Networked Control System

Considering the previous section, we are able now to describe a system architecture that should be considered a reference framework for realizing a WNCS. Such an architecture is a real-time communications system based on Xenomai and RTnet (see Fig. 2): an application in user space may have one or more execution threads, which in turn can be real-time or non real-time. For real-time operations, a task must use the system calls of Xenomai. A real-time task wishing to use Linux system calls or non real-time library functions loses its priority and it is scheduled as a normal Linux thread until the end of the execution of the non real-time service requested.

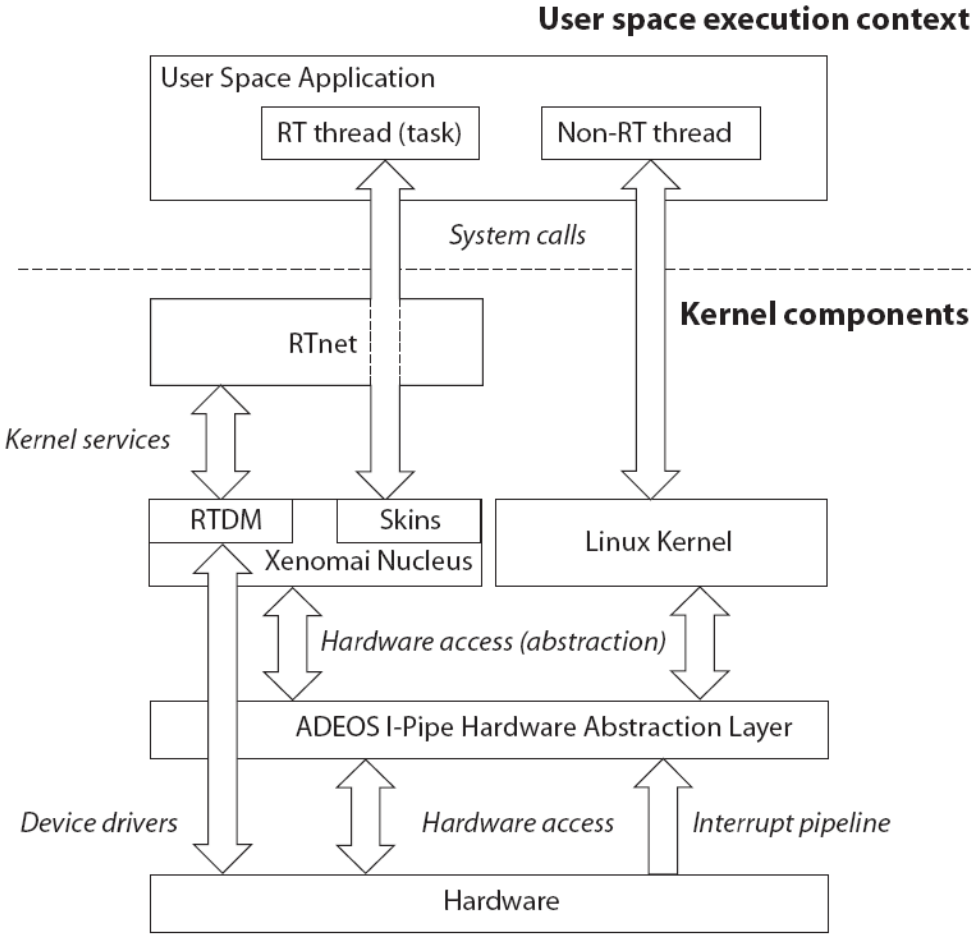


Fig. 2. The RTnet framework and its interaction with other software components.

RTnet runs in Xenomai kernel space, i.e., it is part of the kernel itself, providing network facilities to the user applications by means of a set of system calls. In other words, the stack manager is a real-time kernel task that handles the network packets on behalf of Xenomai nucleus, sending and receiving them from the de-vice drivers that talk directly with hardware. Every data packet coming from the user space is given to the stack manager, which adds the needed headers, handles routing and manages the transmission queue for each network interface. On the other hand, when a packet is received by a network interface, an asynchronous interrupt is triggered by the network hardware device (to signal a packet reception or an error condition), the Adeos I-Pipe redirects the interrupt to Xenomai which gives control to the real-time driver. The packet is copied into memory and is passed to the stack manager, which delivers it to the user application or discards it.

5. Experimental Characterization of the wireless architecture

The described architecture has been tested in an experimental testbed made by four x86-based Personal Computers equipped with the Linux kernel 2.6.20.19. IEEE 802.11g wireless network adapters have been employed, based on the Ralink RT2500 chipset. Each card is equipped with a 2dBi omnidirectional antenna. Xenomai 2.3.4 (Gerum, 200) and Adeos I-Pipe 1.10 (Yaghmour, 2001), with RTnet version 0.9.9 (Kiszka, 2005a) have been used.

Traffic has been generated using an application written using Xenomai's Native API, which is made by several components (i.e., the *sender*, the *responder*, and the non real-time *logger*), as shown in Fig. 3.

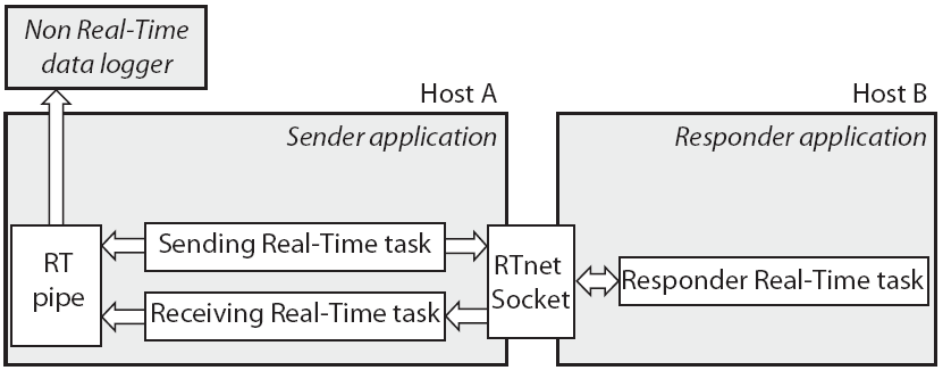


Fig. 3. Software Layout used for measurements.

The *sender* includes a real-time periodic thread which generates packets of fixed size, thus producing a Constant Bit Rate data stream. Each application packet is encapsulated in one UDP datagram. The payload contains a time-stamp measured by the application immediately before sending the packet to the lower layer.

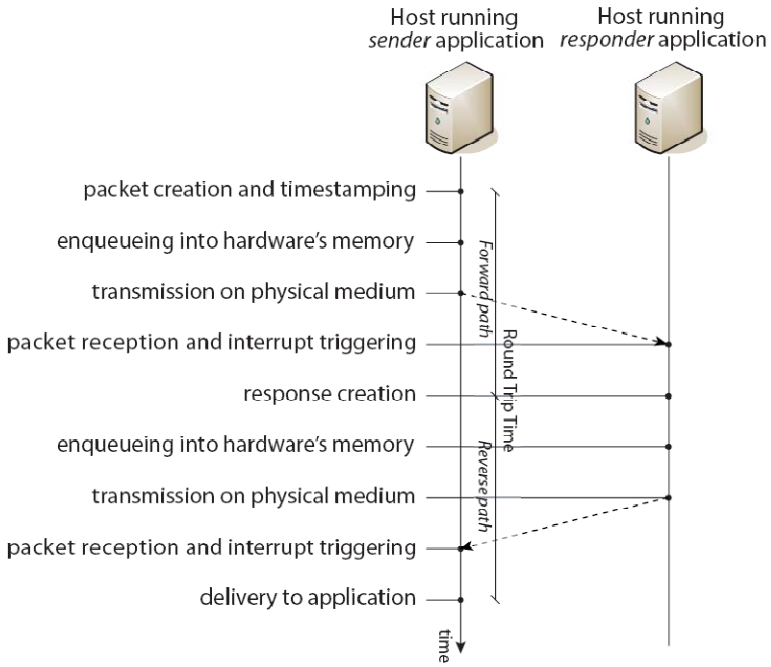


Fig. 4. Components of Round Trip Time.

The *responder* listens for incoming packets from the *sender*; when receiving the packet, it sends back a copy of the packet, which is received by a real-time thread of the sender application. Moreover, the device drivers have been programmed to write the value of the received power (RSSI) into the packets, to constantly monitor the quality of the wireless link between the hosts.

An external, non real-time process collects from a Real-Time Pipe IPC the time values at which the real-time tasks send and receive the packets and the RSSI values, calculates the Round Trip Time (RTT) and stores the RTT and RSSI values on the disk (this approach guarantees that the I/O operations do not affect the realtime behavior of the system). The RTT includes the time elapsed by the packet to traverse the stack manager, the waiting time for the packet in the driver transmission queue before an idle channel, the transmission time, the propagation time, and the time employed by the interrupt handler and the stack manager to process the incoming packet on the receiving side, multiplied by 2 (the packet travels back and forth on the same path), as depicted in Fig. 4. Thus, the measured value is not affected by skew and offset errors, given that transmission and reception times are taken from the same clock on the sender side.

Experiments have been conducted by varying the bit rate, the background traffic load, the source transmission rate, the packet size, CW limits, interframe spaces, and the baseband processor sensitivity.

Tab. 1 summarizes all experiment parameters. The machines have been placed in a typical open space laboratory environment, and two different sets of experiments have been done.

Parameter	1st experiment	2nd experiment
DIFS [μs]	28, 10	10
S_{BP}	50-100	70
Packet Size of the Real-time Traffic [bytes]	128	128
P_{size}: Packet Size of the Background Traffic [bytes]	None	500, 1000
T_{RT} : Transmission period for Real-time traffic [ms]	10	1, 4
Transmission period for Background traffic [ms]	None	5
Bitrate [Mbit/s]	6-54	6-54
CW_{min}	31, 1	1
CW_{max}	1024, 1	1
Retry limit	0	0-2
Number of hosts	2	4

Table 1. Testbed parameters.

5.1 Experimental results without interfering traffic

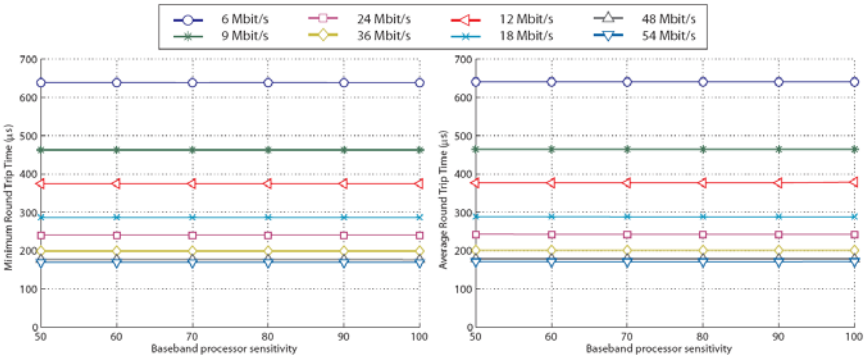
In the first set of experiments, a the testbed is made by two nodes, namely A and B, placed at a close distance (around 1.5 meters), with the antennas in line of sight. The application layer of node A generates a 128 bytes long packet every 10 ms. Node B responds to packets sent by node A as shown in Fig. 3. According to Table 1, experiments are conducted for many values of S_{BP} and bit rate.

In each experiment, 10000 packets are transmitted. Each experiment has been repeated 10 times, for each couple of bit rate and S_{BP} values. In the following, in each figure, both the average value of the ten runs and the 95% confidence interval are reported. Measurements have been done turning off the ACK mechanism provided by the MAC layer.

The first set of experiments has been divided in two parts: in the first one, the wireless adapters have been configured with CW_{min} = 31, CW_{max} = 1023, and DIFS = 28 μs, which are typical values of a best-effort service. In the second part, in order to reduce the latencies due to the 802.11 MAC protocol, CW_{min} and CW_{max} have been set to 1 and the DIFS has been reduced to 10 μs, thus providing a high priority service.

Fig. 5 shows the minimum RTTs obtained for the best effort service. As expected, the minimum RTT increases when the bit rate decreases.

The most relevant contribution to the minimum RTT value is given by the time needed to access the physical medium and to transmit the data. The 802.11g standard (Walke et al., 2006) states that a wireless device must wait at least a DIFS before sending a packet, which is 28 μs long. Moreover, the transmission starts with a fixed size header (i.e., the PLCP, *Physical Layer Convergence Protocol* preamble) used for synchronization between the RF devices, which is 20 μs long (i.e., t_{PLCP} = 20μs). Then, the transmission of the frame can start. Given that the application layer payload is 128 bytes long, the whole frame is 198 bytes long (i.e., a total number of n_{bits} = 1584), adding header overhead due to UDP (8 bytes), IP (20 bytes), Ethernet encapsulation (14 bytes), and 802.11 MAC header and FCS trailer (24+4 bytes). Furthermore, a 6 μs long trailer (called Virtual Extension, VExt) is added at the end of each transmission (i.e., t_{VExt} = 6μs).



(a) Minimum RTT (b) Average RTT

Fig. 5. Minimum and Average RTT for the best-effort service.

Thus, in absence of collisions, neglecting the very low propagation delay and the latency introduced by the real-time protocol stack, the theoretical minimum RTT is given by

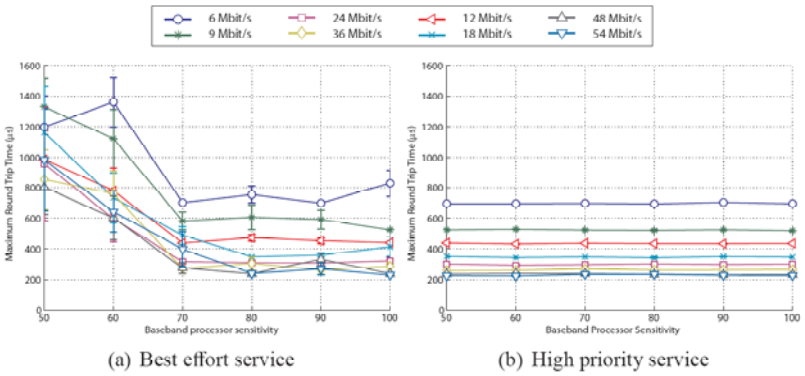
$$RTT = 2\,[DIFS + t_{PLCP} + t_{VExt} + n_{bits}/bitrate] \quad (1)$$

which leads to the results shown in Table 2.
Such values, compared to the measured minimum values for RTT, confirm that the software latencies are almost negligible with respect to the transmission time.
Moreover, Fig. 5 shows that, for the best effort service, the average RTT is very close to the minimum one. This demonstrates that the proposed real-time communication system is able to deliver a very high quota of packets with the smallest delay.
Minimum and average RTTs obtained for the high priority service, being very similar to the one reported in Fig. 5, have not been shown.

Bitrate [Mbit/s]	Theoretical [µs]	Measured [µs]
6	636.00	638.44
9	460.00	462.50
12	372.00	374.55
18	284.00	286.48
24	240.00	240.01
36	196.00	198.53
48	174.00	176.09
54	166.66	168.93

Table 2. Minimum theoretical and measured RTTs.

The importance of a high priority service can be demonstrated by analyzing Maximum RTTs (see Fig. 6). In fact, when the best effort service is used, the maximum RTT (see Fig. 6) depends on both bit rate and baseband sensitivity. The reason is that, as S_{BP} decreases, the transmitter is more likely to detect a busy channel and then, as a consequence, latencies increase. This effect is not evident for the high priority service because shorter DIFS and CW bounds are able to reduce the impact of carrier sensing.

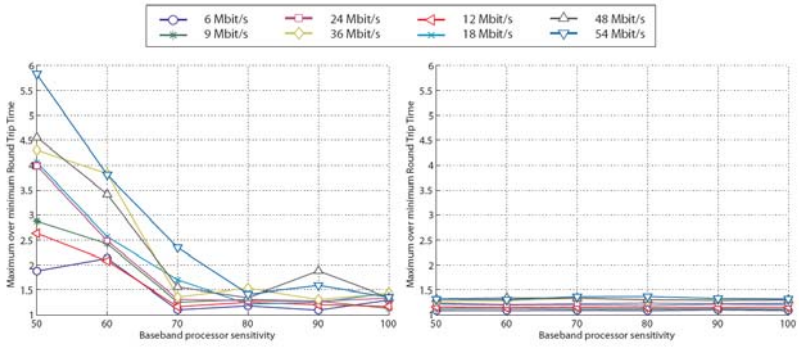


(a) Best effort service (b) High priority service

Fig. 6. Maximum RTT.

To give a further insight, Fig. 7 shows the ratio between the maximum and the minimum RTT. In the best-effort case, only when S_{BP} is greater than 70 and the bit rate is smaller than 48 Mbit/s, the ratio is smaller than 1.5. On the other hand, as expected by the previous findings, when the high priority service is considered, S_{BP} has not any influence on the maximum over minimum RTT ratio, which is always smaller than 1.4.

Fig. 8 shows the Packet Loss Ratio (PLR) computed as missing packets over total packets sent. The number of missing packets is calculated by the sender host by monitoring the number of missing response packets sent by the receiver. Note that the retransmission mechanism of the MAC layer has been turned off, so that this value actually reflects the number of packets that cannot be successfully transmitted at the first try. For both best effort and high priority services, the PLR is smaller than 0.05% when the bit rate is smaller than 48 Mbit/s; moreover, for such bit rates, in many experiments the number of lost packets equals zero. For bit rates of 48 and 54 Mbit/s a higher PLR has been measured because, if we take into account that the transmission power has been kept constant during the experiments, the higher the bitrate the weaker is the modulation scheme used to transmit with respect to noise and interferences (Walke et al., 2006).



(a) Best effort service (b) High priority service

Fig. 7. Maximum over minimum RTT.

From the results presented above, we can derive as rules of thumb to use bit rates smaller or equal to 36 Mbit/s and high priority settings in order to have a robust real-time wireless communications. Anyway, these results represent optimistic performance bounds because obtained without interfering traffic. To assess their real validity, in the next sub-section we will repeat the experiments in presence of concurrent data flows when a high priority service is exploited.

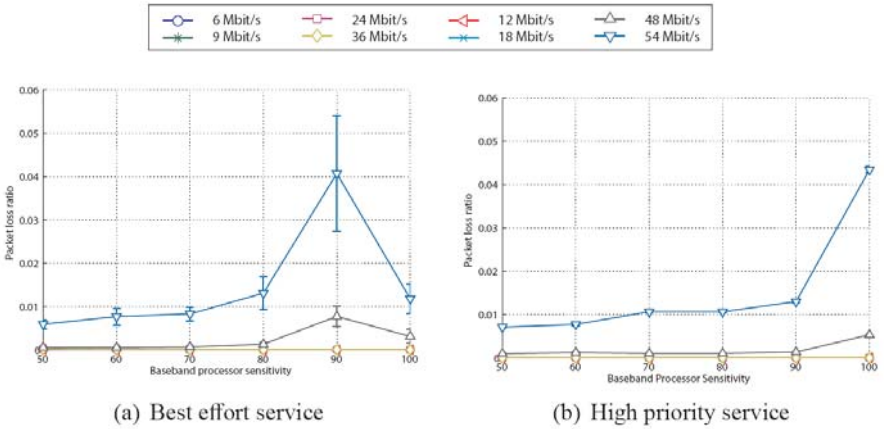
5.2 Experimental results with four hosts and interfering traffic

In the second set of experiments, four machines (namely A,B,C,D) have been placed at the corners of a square area with side length of about 7 meters, at different heights and with some obstacles between them. Hosts A and B exchange real-time data, whereas C and D generate interfering traffic. In particular, using the software described in Fig. 3, A and C generate a CBR traffic sent to B and D, respectively.

For what concerns hosts A and B, interfaces are set in order to provide a high priority service. With respect to the previous experiments, the impact of 802.11 frame retransmission mechanism has been also considered. For that purpose, experiments have been conducted with 802.11 acknowledgment turned off and on, respectively. When turned on, the maximum number of retransmission for each frame has been varied from 1 to 2.

Regarding hosts C and D, default settings for 802.11 interfaces have been used, i.e., CW_{min}=31, CW_{max}=1023, DIFS equal to 28μs, bitrate equal to 54 Mbit/s, acknowledgment mechanism turned on with 7 maximum retries.

According to Table 1, for real-time traffic, 128 bytes long packets are transmitted with inter-transmission times (T_{RT}) equal to 1 ms or 4 ms, whereas for the interfering traffic, the inter-transmission time is 5 ms. By taking into account retransmissions, we have measured an actual inter-transmission time of 1 ms at MAC layer, which is comparable with the inter-transmission time of the real-time traffic. Thus, a stress test has been considered.

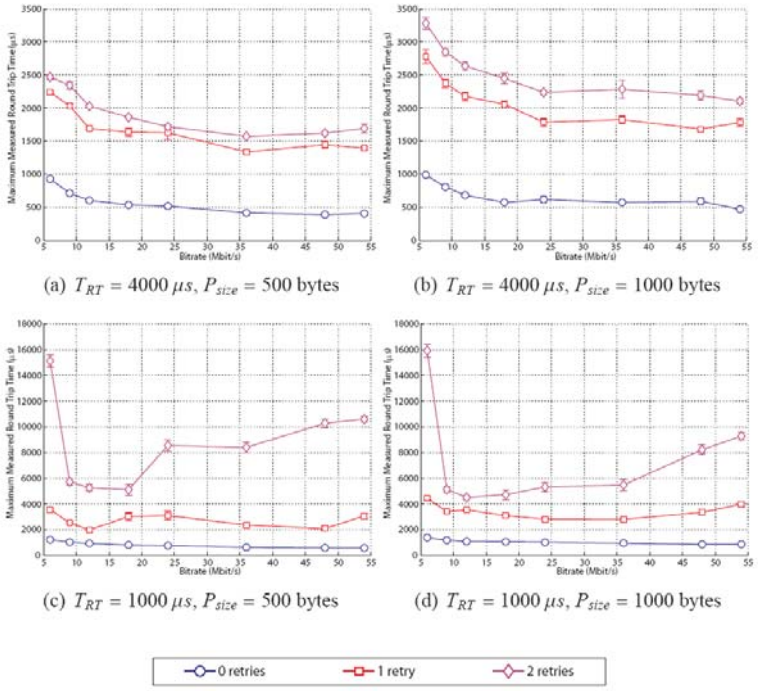


(a) Best effort service (b) High priority service

Fig. 8. Packet Loss Rate.

Two different sizes for the packet size, P_{size}, of the background traffic have been used (500 and 1000 bytes), in order to evaluate the impact of such factor on the performance of the system.

Lastly, we have only considered a S_{BP} value equal to 70, which provided the best performance in the previous set of experiments.
The minimum measured values for RTT are very similar to those already shown in Fig. 5, thus they have not been reported again. The maximum RTT is shown in Fig. 9.
Obviously, for T_{RT} equal to 1 ms we obtain a larger maximum RTT with respect to the case of T_{RT} equal to 4 ms. The reason is that if the transmission rate of real-time packets increases, the probability of finding the channel busy grows too. It is worth to note that, when the inter-transmission time is equal to 4 ms, the RTT is never bigger than 4 ms, which means that transmissions of a packet never overlap with subsequent transmissions. The same conclusion does not hold when the inter-transmission time is 1 ms, which implies that the delay experienced by a real-time packet can influence the delay of future packets. As a consequence, the measured value for the maximum becomes extremely high, about 16 ms in the worst case.
A bitrate of 6 Mbit/s gives the worst performance due to the largest transmission time. Anyway, only a minimal fraction of the frames sent is affected by such high delays. As depicted in Fig. 10, the higher bound on the RTT for 98% of the successfully received frames is much smaller than the maximum measured RTT. Higher bitrates (48 and 54 Mbit/s) already showed to be less robust in the previous experiment, leading to higher values for the overall delay, while a bit rate of 6 Mbit/s needs too much time for the frames to be transmitted without collisions. Intermediate values of the bit rate (9 to 36 Mbit/s) give a RTT that is almost smaller than 2 ms. Thus, with a good degree of approximation we can state that the one-way delay will be no more than a millisecond for such packets.



(a) $T_{RT} = 4000\ \mu s$, $P_{size} = 500$ bytes (b) $T_{RT} = 4000\ \mu s$, $P_{size} = 1000$ bytes

(c) $T_{RT} = 1000\ \mu s$, $P_{size} = 500$ bytes (d) $T_{RT} = 1000\ \mu s$, $P_{size} = 1000$ bytes

Fig. 9. Maximum measured Round Trip Time.

The average RTT is depicted in Fig. 11. It is a decreasing function of the bitrate up to 36 Mbit/s (again, higher the bitrate is and higher the collision probability is). If the retransmissions are enabled, the average value slightly increases, due to the transmission of the acknowledgment frame. In fact, after the successful reception of a frame the host must wait for a SIFS before sending the acknowledgment frame. Thus, the RTT increases as another packet must be sent.

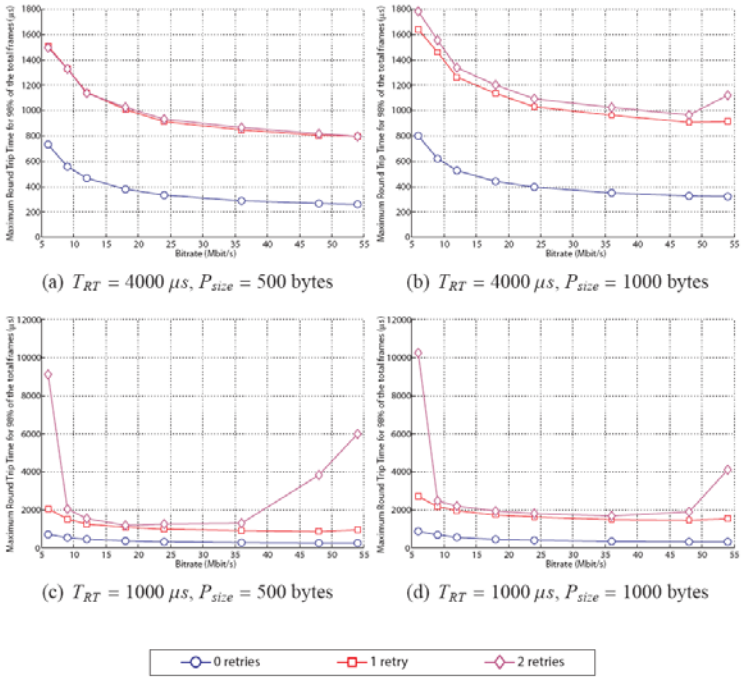
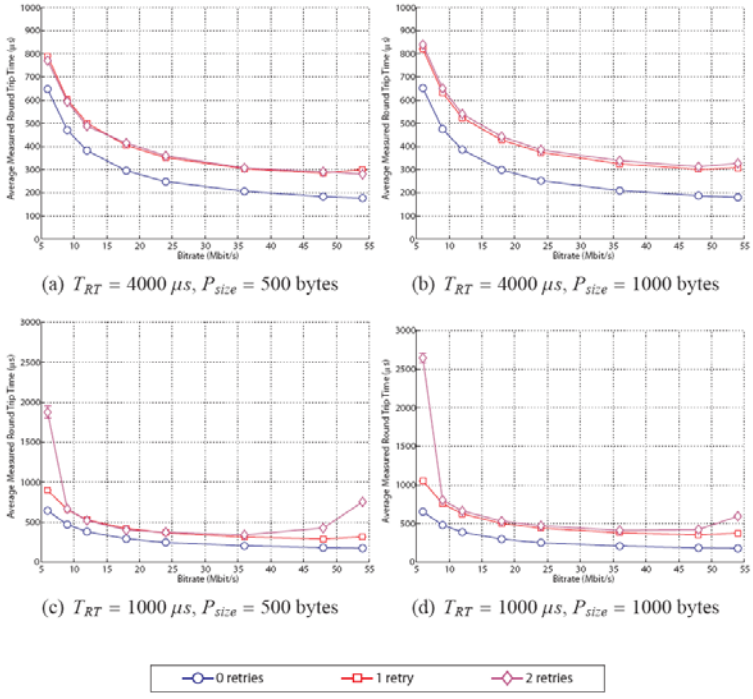


Fig. 10. Maximum measured Round Trip Time (98% of the received frames).

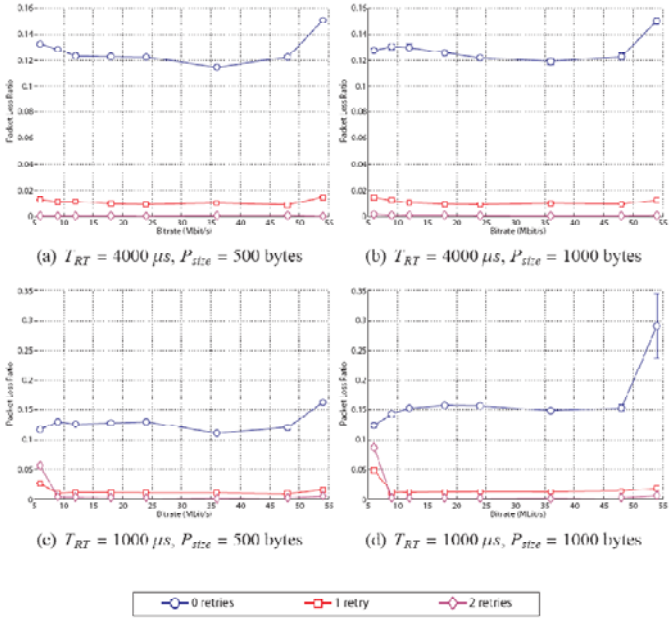
As expected, the performance of the system in terms of packet loss is greatly influenced by the presence of interfering traffic on the channel (see Fig. 12). Without a retransmission mechanism, packet losses are unacceptable, ranging from 12% to 15%. Apart from the 54 Mbit/s bitrate value, the packet loss rate is not directly related to the bitrate value. The retransmission mechanism reduces the PLR to values smaller than 1%, except for the 6 Mbit/s case, where the losses are high even with retrasmissions enabled. Thus, bitrates between 12 Mbit/s and 36 Mbit/s seem to be the best tradeoff between packet loss rate and maximum delay, showing a robust and reliable behavior in all the considered scenarios.
The empirical Cumulative Distribution Functions of the RTT for a bitrate of 36 Mbit/s (which has given the best tradeoff between PLR and delay) is depicted in Fig. 13. If we do not use the retransmission mechanism the knee of the function is very close to the minimum value for the RTT, but the graph has been traced only considering the successfully transmitted packets, so it does take into account the heavy PLR. With retransmissions enabled, the system guarantees that at least 75% of the packet exchanged can be delivered in a timely manner, that is, with a value which is quite close to the minimum measured.



(a) $T_{RT} = 4000\ \mu s$, $P_{size} = 500$ bytes (b) $T_{RT} = 4000\ \mu s$, $P_{size} = 1000$ bytes

(c) $T_{RT} = 1000\ \mu s$, $P_{size} = 500$ bytes (d) $T_{RT} = 1000\ \mu s$, $P_{size} = 1000$ bytes

Fig. 11. Average measured Round Trip Time.



(a) $T_{RT} = 4000\ \mu s$, $P_{size} = 500$ bytes (b) $T_{RT} = 4000\ \mu s$, $P_{size} = 1000$ bytes

(c) $T_{RT} = 1000\ \mu s$, $P_{size} = 500$ bytes (d) $T_{RT} = 1000\ \mu s$, $P_{size} = 1000$ bytes

Fig. 12. Packet Loss Rate.

To conclude, experiments clearly show that the considered communication architecture, with a proper setting of networking interface card parameters, provides very tight bounds on both packet delays and losses. In particular, using high priority settings, with bitrates ranging from 12 -36 Mbit/s and retry limit equal to 2, the 98% of packets experience a RTT no larger than 2 ms with negligible packet losses, even in the presence of interferring traffic. These results demonstrate that a further step toward Wireless Networked Control Systems has been done.

6. Proposed TDMA scheme for a Wireless Networked Control System

The RTnet TDMA access discipline has been conceived for wired networks, which are much more reliable and less subject to electromagnetic interferences than wireless ones. As a consequence, the expected delay jitter of wireless networks can make troublesome its exact calibration.

In fact, the carrier sense mechanism of 802.11 may delay the transmission of a packet, making it overlap with subsequent timeslots and causing a chain effect which could break the whole system synchronisation. Lastly, the transmission of outgoing packets on each station is triggered by the reception of the synchronisation packets; a station that does not receive such synchronisation packets loses the chance to transmit into the current TDMA frame.

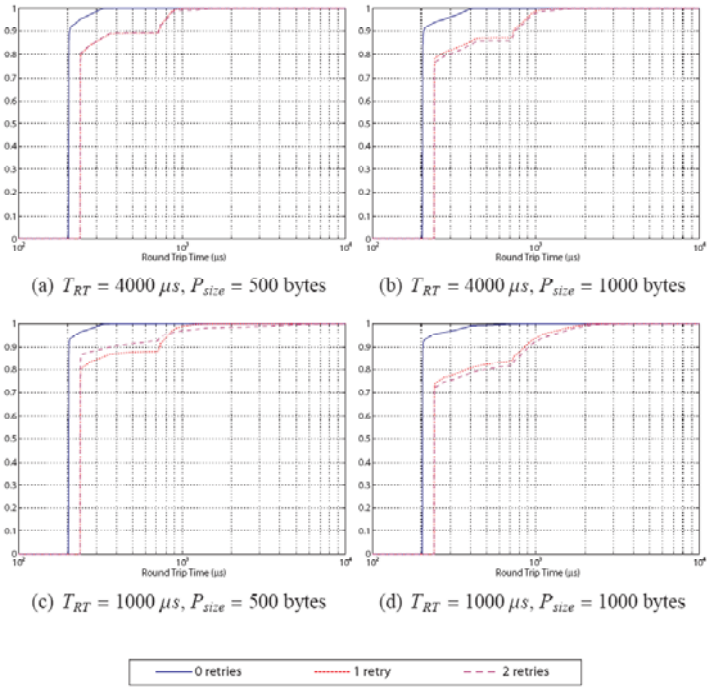


Fig. 13. Cumulative distribution functions of measured RTT.

This can lead to instability, as the output queues may fill up causing unacceptable delays or packet losses. In order to make the current TDMA implementation work reliably over 802.11 networks, we hereby propose a few modifications that could be applied to cope with the characteristics of the wireless channel:

- **Protection mechanisms**: to reduce the probability of packet collisions, a protection scheme should be used. The IEEE 802.11g amendment provides a CTS-to-self mechanism that a station can use to assign to itself the right to access the channel, as it received a RTS (Request to Send) from another station (Walke et al., 2006).
- **Ordered retransmission scheme:** the retransmission mechanism used in 802.11 tries to retransmit the lost packet as soon as the channel is sensed idle again. This approach can not be safely used with a TDMA scheme, as the packet retransmission may overlap with a timeslot assigned to another station. It is necessary to implement a retransmission scheme able to defer the retransmission only when all the stations have already transmitted their data.
- **Automatic recovery of timing**: when a synchronisation packet is not correctly received, a station should be able to synchronize automatically with the start of the TDMA frame, in order not to lose its chance to transmit during the current TDMA frame. This can be safely done for short periods of time using a proper weighted average of the interarrival times of the last synchronisation frames, if a station does not miss more than a few synchronisation rounds consecutively. Otherwise the start of TDMA frame may not be calculated correctly anymore due to clock drift. The use of backup masters can reduce the occurrences of such event.
- **Active monitoring**: a station wishing to transmit may choose to actively monitor the channel activity before sending the packet to the hardware. If the channel is sensed busy for too much time, the driver may decide to drop the packet instead of sending it, in order to avoid overlapping with other timeslots.

These solutions should be easily implemented on generic 802.11 network adapters, as long as the source code for the device driver is available and the firmware lets the user control the basic MAC parameters. The proposed approach should make the real-time stream more robust with respect to interferences. The actual performance increase has still to be evaluated. At the moment, we have implemented an ordered retransmission scheme which defers the retransmission of a packet to a special timeslot at the end of the cycle. We have considered the TDMA frame structure made of two parts, as described in Fig. 14.

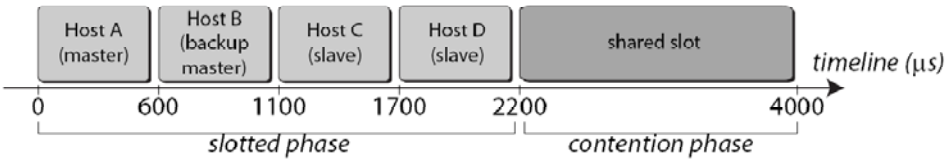


Fig. 14. Proposed TDMA frame structure.

The first part, called *slotted phase*, contains normal TDMA timeslots, which can be assigned to each station. The second part, called *contention phase*, contains a single timeslot, which is shared by all the stations for retransmissions. For the time being, a single retransmission attempt for each frame has been considered.The 802.11 acknowledgment mechanism can be

used or the packets can be acknowledged at application layer. Thus, any packet that has not been acknowledged at the end of the slotted phase can be sent during the contention phase.

7. Experimental Analysis of the proposed TDMA scheme

The real-time communications architecture used to test the TDMA scheme is the same one considered in Sec. 5. Each machine has been equipped with a Ralink RT2500 WLAN adapter. The host have been placed in a laboratory open space, at a distance of approximatively 10 meters among each other. A simple TDMA network has been created among the hosts, one of which has acted as master, another as backup master, and the others as slaves, as reported in Fig. 14. The DIFS value has been set to 10 μs, and the CW_{min} and CW_{max} values have been set to their standard minimum value (Walke et al., 2006). Finally, the S_{BP} value has been set to 80, which is a good tradeoff between latencies and packet loss rate, as shown in Sec. 5. These settings guarantee the highest priority on the wireless channel. We have considered all bit rate values allowed by 802.11g amendment for OFDM operation, from a minimum of 6 Mbit/s to a maximum of 54 Mbit/s Walke et al., 2006). A wireless channel almost free from other transmissions has been considered during the measurements. The IEEE 802.11 retransmission mechanism has been turned off.

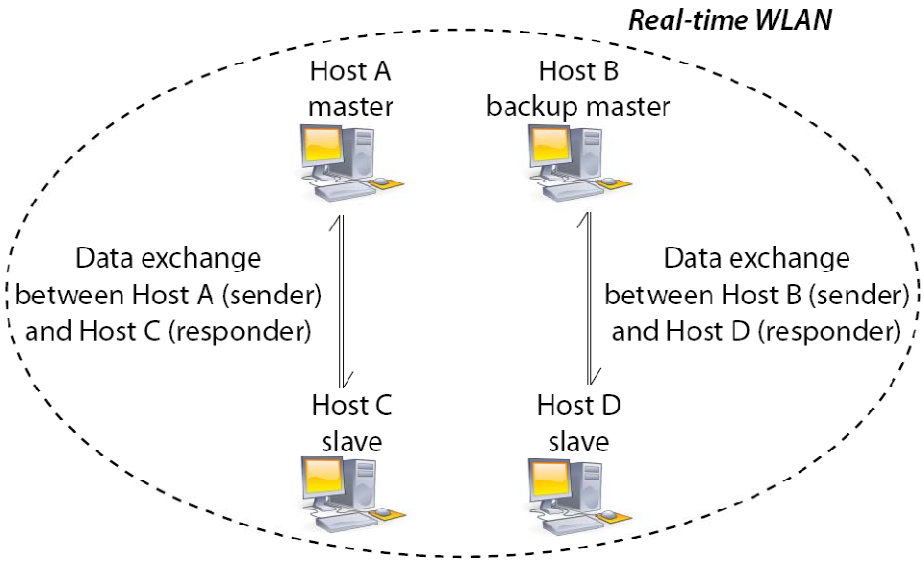


Fig. 15. Testbed used for measurements.

Traffic has been generated using a custom application made by several components (i.e., the *sender*, the *responder*, and a non real-time data logger). The *sender* includes a periodic thread which generates 128 bytes long packets, synchronised with the TDMA frame (i.e., 4 ms), so that it generates a new packet for every cycle; packets at application layer are encapsulated in UDP datagrams. The sync reception acts as a trigger for packet creation. The packet

contains a unique identifier and a time-stamp taken by the application when the packet is created. The *responder* listens for incoming packets from the sender; when receiving the packet, it sends back a copy of the packet, which is received by a thread of the sender application. An external process collects from a Real-Time Pipe IPC the transmission and reception time values, calculates the Round Trip Time (RTT) and stores the values on disk. The *sender* has been run on hosts A and B, sending traffic to hosts C and D respectively, which were running the *responder* software, as depicted in Fig. 15.
When the *sender* host does not receive an answer during the slotted phase, it retransmits the packet in the contention phase, and so does the *responder*. Moreover, if the measured RTT of a packet is greater than the TDMA frame length, the responder host may have lost a TDMA synchronization packet. Thus, a response packet is still held into the output queue of the responder host. When this happens, the *sender* notifies the *responder* which uses the contention phase to send the packet left in its queue. In each experiment the hosts exchanged 5000 packets, and each experiment has been repeated 5 times, for a total of 25000 packets exchanged for each bit rate value considered.
Fig. 16 shows the minimum and the maximum RTT measured for both real-time streams. Given that hosts A and B take a new timestamp at the beginning of each frame, the minimum RTTs are very close to the timeslot offsets of the responder hosts C and D, respectively. Moreover, they decrease with the bit rate due to a smaller transmission time. The maximum RTT is mostly due to the missed reception of the sync frame by the slave nodes. The obtained values indicate that the retransmission scheme is able to recover the normal system behavior in at most two TDMA frames.

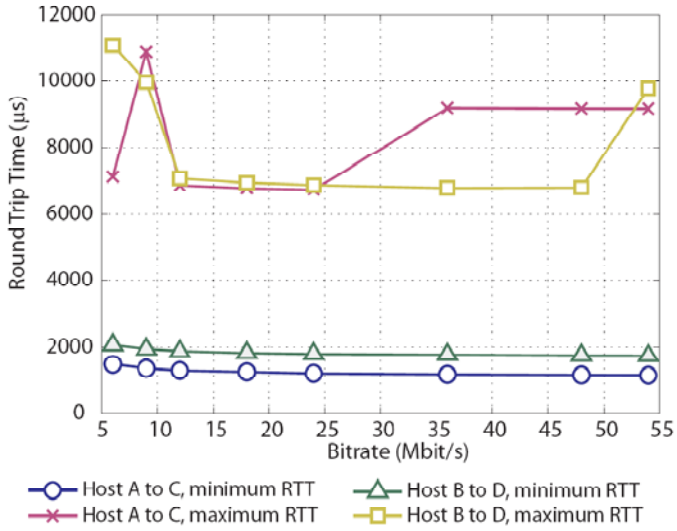


Fig. 16. Minimum and Maximum RTT.

Fig. 17 shows the percentage of lost and recovered packets for each considered bit rate. Most of the packets have been recovered by the retransmission scheme, though it allows just one retransmission. Losses are very small, always less than 0.3% of the total sent packets.

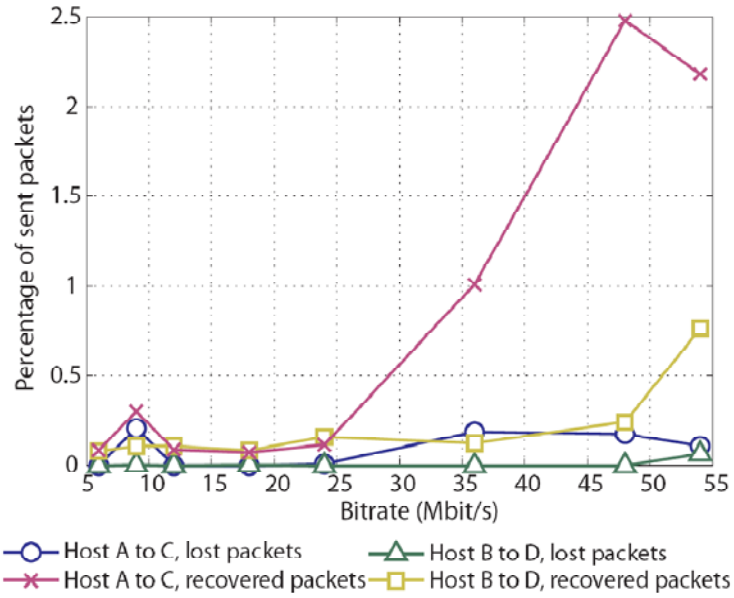


Fig. 17. Percentage of lost and recovered packets.

8. Conclusions

This chapter has analyzed the feasibility of a 802.11 based wireless real-time communication system. For that purpose a wireless communication architecture that properly integrates the leading IEEE 802.11 technology, the RTnet framework, and the Xenomay nano-kernel has been implemented. This architecture has been experimentally tested for various transmission data rates, baseband processor sensitivities, and sampling intervals, with and without interfering traffic. Experiments have demonstrated that by properly setting protocol parameters a robust real-time service can be provided.

9. References

Andersson M.; Henriksson D.; Cervin A. & Arzen K. E. (2005). Simulation of wireless networked control systems, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, Dec. 2005.

ANSI (1994). *Portable Operating Sytem Interface (Posix)*. ANSI, IEEE, 1994.

Baillieul J. & Antsaklis P. J. (2007). Control and Communication Challenges in Networked Real-time Systems. *Proceedings of the IEEE*, Vol. 95, No. 1, (Jan. 2007) 9-28.

Baliga G. & Kumar P. R. (2005). A middleware for control over networks, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, Dec. 2005.

Barr M. (2003). Choosing an RTOS. *Tech. Rep. Embedded Systems Programming*, Jan. 2003.

Benvenuti C. (2006). *Understanding Linux Network Internals*. O'Reilly, 2006.

Biasi M. D.; Snickars C.; Landernas K. & Isaksson A. J. (2008). Simulation of process control with wirelesshart networks subject to packet losses, *Proceedings of 4th IEEE Conference on Automation Science and Engineering*, Washington DC, USA, Aug. 2008.

Boggia G.; Camarda P.; Grieco L. A. & Zacheo G. (2008a). Toward wireless networked control systems: an experimental study on real-time communications in 802.11 wlans, *Proceedings of 7th IEEE International Workshop on Factory Communication Systems, WFCS*, Dresden, Germany, May 2008.

Boggia G.; Camarda P.; Grieco L. A. & Zacheo G. (2008b). An experimental evaluation on using TDMA over 802.11 MAC for wireless networked control, *Proceedings of Emerging Technologies and Factory Automation, ETFA*, Hamburg, Germany, Sep. 2008.

Boughanmi N.; Song Y. & Rondeau E. (2008). Wireless networked control system using IEEE 802.15.4 with GTS, *Proceedings of 2nd Junior Researcher Workshop on Real-Time Computing, JRWRTC*, Rennes, Brittany, Oct. 2008.

Burda R. & Wietfeld C. (2007). Multimedia over 802.15.4 and ZigBee Networks for Ambient Environmental Control, Proceedings of the IEEE VTC Spring, Dublin, Ireland, Apr. 2007.

Buttazzo G.; Velasco M. & Martì P. (2007). Quality-of-Control Management in Overloaded Real-time Systems. *IEEE Trans. on Computers*, Vol. 56, No. 2, (Feb. 2007) 253–266.

Cena G.; Bertolotti I. C.; Valenzano A. & Zunino C. (2007). Evaluation of response times in industrial WLANs. *IEEE Trans. on Industrial Informatics*, Vol. 3, No. 3, (Aug. 2007) 191–201.

Cervin A.; Ohlin M. & Henriksson D. (2007). Simulation of networked control systems using truetime, *Proceedings of the 3rd International Workshop on Networked Control Systems: Tolerant to Faults*, Nancy, France, Jun. 2007.

Chen J.; McKernan A.; Irwin G. W. & Scanlon W. G. (2008). Experimental characterisation and analysis of wireless network control systems, *Proceedings of the IET Irish Signals and Systems Conference, ISSC*, Galway, Ireland, Jun. 2008.

Choi D. H.; Lee J. I.; Kim D. S. & Park W. C. (2006). Design and implementation of wireless fieldbus for networked control systems, *Proceedings of SICE-ICASE International Joint Conference*, Bexco, Busan, Korea, Oct. 2006.

DIAPM (2008), Real-time application interface (RTAI) for Linux. *Tech Rep. Politecnico di Milano*, 2008, available online: http://www.rtai.org.

Flammini A.; Marioli D.; Sisinni E. & Taroni A. (2009). Design and implementation of a wireless fieldbus for plastic machineries. *IEEE Trans. on Industrial Electronics*, Vol. 56, No. 3, (Mar. 2009) 747–755.

Floroiu J.; Ionescu T. C.; Ruppelt R.; Henckel B. & Mateescu M. (2001). Using NDIS intermediate drivers for extending the protocol stack. a case-study. *Computer Communications*, Vol. 24, No. 7-8, (Apr. 2001) 703–715.

Gerum P. (2004). Xenomai – implementing a RTOS emulation framework on GNU/Linux. Available online : http://www.xenomai.org/documentation.

Hasan M. S.; Yu H.; Griffiths A.& Yang T. C. (2007). Simulation of distributed wireless networked control systems over MANET using OPNET, *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, London, UK, Apr. 2007.

Hespanha J. P.; Naghshtabrizi P. & Xu Y. (2007). A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, Vol. 95, No. 1, (Jan. 2007) 138–162.

Heynicke R.; Kruger D.; Wattar H. & Scholl G. (2008). Modular wireless fieldbus gateway for fast and reliable sensor/actuator communication, *Proceedings of Emerging Technologies and Factory Automation, ETFA*, Hamburg, Germany, Sep. 2008.

IEEE (1999a). *IEEE 802.11, Information Technology -Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN MAC and PHY Specifications, 1st ed., ANSI/IEEE Std. 802.11, ISO/IEC 8802-11*. IEEE standard for Information Technology, 1999.

IEEE (1999b). *IEEE 802.11, Supplement to IEEE Standard for Information Technology. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN MAC and PHY Specifications: Higher-Speed Physical Layer Extension in the 5 GHz Band, IEEE Std 802.11a, ISO/IEC 8802-11:1999/Amd 1:2000(E)*. IEEE standard for Information Technology, 1999.

IEEE (1999c). *Supplement to IEEE Standard for Information Technology. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN MAC and PHY Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std 802.11b)*. IEEE standard for Information Technology, 1999.

IEEE (2003). *Supplement to IEEE Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std 802.11g*. IEEE standard for Information Technology, 2003.

IEEE (2005). *Amendment to Standard for Information Technology. LAN/MAN Specific Requirements -Part 11: Wireless MAC and PHY Specifications: MAC Quality of Service (QoS) Enhancements, IEEE 802.11e/D13.0*. IEEE standard for Information Technology, 2005.

IEEE (2006). *Std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE standard for Information Technology, Sept. 2006.

Kim W.; Ji K. & Ambike A. (2006). Real-time Operating Environment for Networked Control Systems. *IEEE Trans. on Automation Science and Engineering*, Vol. 3, No. 3, (Jul. 2006) 287–296.

Kiszka J. (2005b). The real-time driver model and first applications. *Tech. Rep. Xenomai*, available online: http://www.xenomai.org/documentation/

Kiszka J.; Wagner B.; Zhang Y. & Broenink J. (2005a). RTnet – a flexible hard real-time networking framework, *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*, Catania, Italy, Sep. 2005.

Krber H.J.; Wattar H. & Scholl G. (2007). Modular wireless real-time sensor/actuator network for factory automation applications. *IEEE Trans. on Industrial Informatics*, Vol. 3, No. 2, (May 2007) 111–119.

Lee S.; Park J. H.; Ha K. N. & Lee K. C. (2008). Wireless networked control system using NDIS-based four-layer architecture for IEEE 802.11b, *Proceedings of IEEE Int. Workshop on Factory Communication Systems, WFCS*), Dresden, Germany, May 2008.

Liu G. P.; Xia Y.; Chen J.; Rees D. & Hu W. (2007). Networked Predictive Control of Systems with Random Network Delays in both Forward and Feedback Channels. *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 3, (Jun. 2007) 1282–1297.

Massa A. (2002). *Embedded Software Development with ECos*. Prentice Hall PTR, 2002.

McKenney P. (2005). A realtime preemption overview. *LWN.net*, available online: http://lwn.net/Articles/146861.

Moyne J. R. & Tilbury D. M. (2007). The Emergence of Industrial Control Networks for Manufactoring Control, Diagnostics, and Safety Data. *Proceedings of the IEEE*, Vol. 95, No. 1, (Jan. 2007) 29–47.

Nair G. N.; Fagnani F.; Zampieri S. & Evans R. J. (2007). Feedback Control Under Data Rate Constraints: An Overview. *Proceedings of the IEEE*, Vol. 95, No. 1, (Jan. 2007) 108–137.

Nethi S. ; Pohjola M.; Eriksson L. & Jntti R. (2007). Platform for emulating networked control systems in laboratory environments, *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM)*, Helsinki, Finland, Jun. 2007.

Neumann P. (2007). Communication in industrial automation what is going on?. *Control Engineering Practice*, Vol. 15, No. 11 (Nov. 2007) 1332-1347.

Pellegrini F. D.; Miorandi D.; Vitturi S. & A. Zanella (2006). On the use of wireless networks at low level of factory automation systems. *IEEE Trans. on Industrial Informatics*, Vol. 2, No. 2, (May 2006) 129–143.

Ralink (2006) Technologies. Ralink rt2500 chipset overview. *Tech Rep. Ralink Technologies*, available online: http://www.ralinktech.com

Rauchhaupt l. (2002). System and device architecture of a radio based fieldbusthe rfieldbus system, *Proceedings of the 4th IEEE International Workshop on Factory Communication Systems*, Vasteras, Sweden, Aug. 2002.

Robinson C. L. & Kumar P. R. (2007). Sending the most recent observation is not optimal in networked control: Linear temporal coding and towards the design of a control specific transport protocol, *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Louisiana, USA, Dec. 2007.

Schenato L.; Sinopoli B.; Franceschetti M.; Poolla K. & Sastry S. S. (2007). Foundations of Control and Estimation over Lossy Networks. *Proceedings of the IEEE*, Vol. 95, No. 1, (Jan. 2007) 163–187.

Silberschatz A.; Galvin P. B. & Gagne G. *Operating System Concepts (7th Edition)*. Wiley, 2004.

Song J.; Han S.; Mok A. K.; Chen D.; Lucas M.; Nixon M. & Pratt W. (2008). Wirelesshart: Applying wireless technology in real-time industrial process control, *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, St. Louis, MO, USA, Apr. 2008.

Straumann T. (2001). Open source real time operating systems overview, *Proceedings of the 8th International Conference on Accelerator and Large Experimental Physics Control Systems*, San Jose, CA, USA, 2001.

Tabbara M.; Nesic D. & Teel A. R. (2007). Stability of Wireless and Wireline Networked Control Systems. *IEEE Trans. on Automatic Control*, Vol. 52, No. 9, (Sep. 2007) 1615–1630.

Varshney U. (2003). The Status and Future of 802.11-Based WLANs. *IEEE Computer*, Vol. 36, No. 6, (Jun. 2003) 102–105.

Walke B. H.; Mangold S. & Berlemann L. (2006). *IEEE 802 Wireless Systems*, John Wiley and Sons, 2006.

Willig A.; Matheus K. & Wolisz A. (2005). Wireless technologies in industrial networks. *Proceedings of IEEE*, Vol. 93, No. 6, (Jun. 2005) 1130–1150.

Wu J. & Chen T. (2007). Design of Networked Control Systems with Packet Dropouts," *IEEE Trans. on Automatic Control*, Vol. 52, No. 7, (Jul. 2007) 1314–1319.

Yaghmour K. (2001). Adaptive domain environment for operating systems. *Tech. Report Adeos*, available online: http://www.opersys.com/ftp/pub/Adeos/adeos.pdf.

4

When the Industry Goes Wireless: Drivers, Requirements, Technology and Future Trends

Simon Carlsen[1] and Stig Petersen[2]
[1]*StatoilHydro ASA, Harstad,* [2]*SINTEF ICT, Trondheim*
[1,2]*Norway*

1. Introduction

Through the last ten to fifteen years wireless communication technology has become a natural and fully integrated part of our everyday lives. The two most exposed applications, digital mobile telephony and wireless computer networks, are so common that it is hard to imagine a world without these technologies. In addition, a number of different everyday devices in the home, in the car or in the office communicate with each other over short range wireless links, utilizing technologies like Bluetooth or similar.
Even though what is said above may seem obvious to most people in the industrialized world, the situation is somewhat different when it comes to applications of wireless technology in the industry itself. Compared to the numerous applications of wireless communication that we all are so familiar with and have learned to consider as indispensable from a consumers' point of view, the benefits of wireless solutions in industrial applications have until the last few years not been so obvious. Of course, different industries and companies are at different stages regarding the implementation and adoption of wireless technology, but in general we see a conservative approach, and the reasons for such a progression are many.
This chapter will deal with some important aspects as the industry slowly evolves from a wired world into the wireless domain. It is organized as follows; section 2 examines the motivations and drivers for introducing wireless technology within the industry, section 3 presents the industrial requirements which the wireless technology must fulfil in order to be a viable option to today's wired solutions, section 4 gives an overview of the most relevant international standards for industrial wireless communications, and section 5 concludes the chapter by identifying the current trends and important future research areas for industrial wireless technology.

2. Applications, drivers and motivation

To enable the introduction of any new technology in the enterprise, a major driver and motivational factor is the potential financial gains, i.e. reduced costs and/or increased

revenue. Secondly, if new technology has the potential to benefit other important aspects such as health, safety or the environment (HSE), they would also be considered interesting for the industry. Potential areas in which wireless technology can be beneficial to the industry can be divided into three distinct applications; mobile ICT (information and communication technology), wireless instrumentation, and asset and personnel tracking.

2.1 Mobile ICT

The development and rapid deployment of systems adhering to the IEEE Std 802.11 for wireless local area networks (WLANs) have enabled Internet access to mobile devices such as laptops, personal digital assistants (PDAs) and high-end mobile phones, from nearly anywhere at any time. WLAN access points are deployed in office buildings, public spaces, airports, cafeterias and in private homes, providing either free or purchasable Internet access to everybody in the vicinity. While wireless access in the home, office or public spaces mainly has focused on internet access and access to enterprise systems on the office network, the focus is somewhat different for industrial applications. Some relevant application areas involving the use of mobile ICT in the industry includes:

- Simplification of work processes
- Simplify or automate routine test procedures
- 'Bringing the control room to the field'
- Online field access to status, maintenance logs etc for instruments and components as a part of fault diagnostics procedures
- Inspection and maintenance tasks by means of WLAN enabled mobile cameras, where the field operator communicates in real-time over video and audio with remote expert centres

Commonly, mobile ICT applications in the industry are associated with local on-site WLAN networks, which to date has more or less followed the same implementation strategy as for WLANs' in office environments. The benefits for deploying local network infrastructure are many. For example, it ensures sufficient bandwidth for demanding applications, and the security and data integrity aspects are locally controlled.

In some industries, however, the costs for deploying local infrastructure for enabling wireless coverage in the process areas can be significant. This is particular relevant in industries which comprise explosive atmospheres, such as Oil & Gas and mining. In such areas, very strict restrictions apply regarding requirements for all electrical apparatus intended for use inside specific zones. In Europe, the ATEX directive (European Parliament and the Council, 1994) contains the governing rules, regulations and requirements for the use of electrical equipment in hazardous environments. Other countries have similar directives, for example in North America and Canada the North American Hazardous Locations Installation Codes (National Electric Code for the US and the Canadian Electrical Code for Canada) define rules and regulations on equipment and area classifications requirements for hazardous locations. Network equipment planned for use in such areas must be certified to conform to these regulations. In practise, this involves either regular equipment built into specially designed enclosures, or it demands for complete redesigns of the equipment itself. The certification is a comprehensive process that can only be carried out by selected certification agencies. This leads to significant increases in equipment cost. As an example, a WLAN access point manufactured for corporate use has a cost of approximately 800 – 1,000 USD in its ordinary version. The ATEX-certified version (the

same access point built into an enclosure, and the unit certified as a whole), has a cost in the order of 8,000 – 10,000 USD, e.g. the price has increased by a factor of ten. In addition, strict installation procedures for equipment in process plants are further cost-driving parameters. The above facts, combined with a general lack of pre-certified WLAN equipment in the market designed for use in explosive atmospheres, has been a showstopper for the rapid deployment of large-scale mobile ICT in industries like the Oil & Gas. For these reasons, these industries have started looking at public networks such as GPRS and UMTS as alternative access channels.

We end this section by giving an example on the use of mobile ICT to simplify a work process in the process industry. The example is taken from (Petersen et al., 2008).

2.1.1 Example – Simplifying maintenance routine jobs

Traditionally, work processes in process industries involve a number of manual operations. A typical workflow for operation and maintenance tasks is presented in Fig. 1. The flowchart illustrates how several activities have to be performed in a given order. If we consider a maintenance operation where notifications and work orders are required, typically the whole process has the following (simplified) progression from a field operators' point-of-view:

1. Initiate operation
2. Create a notification. This is commonly done with the corporate Enterprise Resource Planning (ERP) system
3. Await confirmation from ERP system
4. Create work order through ERP
5. Await signed work permit
6. Prepare operation
7. Plan operation
8. Execute maintenance operation
9. Close operation
10. Verify technical condition restored
11. Update documentation, both in technical documentation systems and ERP

Wireless network access in the field can help simplify this process. Consider a Personal Digital Assistant (PDA) with wireless access to the company's backbone systems. The PDA is equipped with an RFID or barcode reader for reading information from tagged plant equipment.

When the field operator detects a faulty component that is subjective to maintenance, the tag of the component is read or scanned with the PDA. A notification describing the upcoming maintenance operation is created on the PDA. This information is then transmitted via the wireless network into the ERP system.

As soon as the necessary confirmation is received, a work order is created. During the maintenance operation, the field operator can update the technical documentation online from his mobile device. As a finishing activity, a verification of the technical condition of the component has to be performed, commonly requiring that the operator is physically present in the field. When the verification is passed, the operator is able to remotely flag the status of the work order as finished using the PDA.

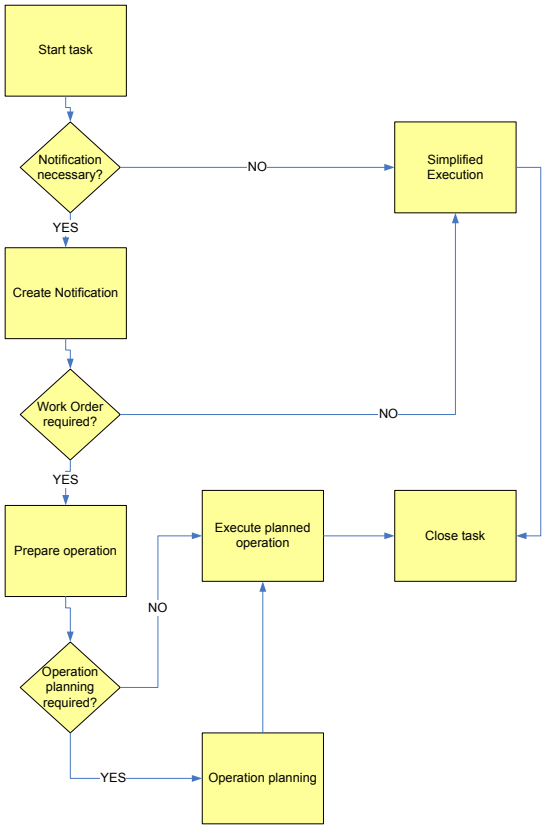


Fig. 1. Typical workflow for mainteance operation

2.2 Wireless instrumentation

Recent advances in wireless technology have enabled the development of low-cost, low power wireless sensors capable of robust and reliable communication (Akyildiz et al., 2002). The IEEE Std 802.15.4 (IEEE 802.15.4, 2006) defines the physical layer (PHY) and the medium access control sublayer (MAC) for low-rate wireless personal area networks. Inherent features such as ultra-low complexity, cost and power makes it a very suitable standard for wireless sensor network (WSN) solutions (Yu, Q et al., 2006). With a growing number of both standardized and proprietary solutions based the IEEE Std 802.15.4 PHY and MAC appearing on the market, it has quickly become the *de facto* standard for WSNs.

Using sensors to monitor both the performance and the operational environment of industrial plants and facilities allows for greater insight into operational requirements and potential safety problems. The sensors are used to monitor a wide range of parameters, e.g. pipeline pressure, flow, temperature, vibration, humidity, gas leaks, fire outbreaks and equipment condition. The collected sensor data is then used to make informed just-in-time decisions on plant performance and operational conditions. It is expected that the

continuing advances in WSN technologies will enable wireless sensing, monitoring and control applications within the following industrial areas (Petersen et al., 2007):

- Condition and performance maintenance monitoring
- Area and property surveillance and monitoring
- Environmental monitoring
- Emergency management
- Process control

In addition, eliminating the need for cables will contribute to reduced installation costs, and extend coverage into areas previously either too remote or too hostile to be viable for wired instrumentation. Furthermore, using wireless sensors provides the possibility of doing temporary installations and mobile installations, for example in conjunction with turnarounds and shutdowns. So, flexibility and installation time are major beneficial factors making the use of wireless sensors very cost-effective compared to traditional instrumentation.

As an example on a real-world application, a wireless sensor network was installed at an oil production platform in the North Sea. The complete scenario is extensively described in (Carlsen et al., 2008)

2.2.1 Example – Using Wireless Sensor Networks to Enable Increased Oil Recovery

The actual oil field is in its tail-end lifecycle, and, combined with the geological structure consisting of many small oil accumulations, occasional loss of flow from the wells was not readily detected, which lead to unplanned stops in the production. During the construction stage back in the 1980's, no flow metering devices were installed inside the flow lines. Calculations performed by staff personnel at the actual license show that unpredicted stops in production due to unexpected loss of well pressure counts for annual financial losses in the order of 40 million USD. Based on these calculations, it is clear that a reliable, easy to install detection system for alerting upcoming pressure losses is very attractive for gaining increased revenue.

The installation and maintenance of a traditional detection system (flow meters inside the pipes) is complex and requires a complete production shutdown, and was not considered an alternative. Another showstopper for introducing wired sensor equipment in a live production environment is the need for cables. As these units need both wired power and a wired communication link, the complexity and cost factors are high.

A simple approach to determine loss of flow in a well is to measure the temperature of the well flow line, some distance downstream of the wellhead. This is based on the principle that loss of flow causes a reduction in surface temperature of the pipe as heat is lost to the surroundings. The typical well fluid temperature is approximately 60ºC, thus the temperature measurement can be performed on the pipe's surface. This eliminates the need for an invasive installation, which greatly simplifies installation. Until recently, loss of flow from individual wells was detected by plant operators manually probing the surface temperature of the flow lines during inspection rounds one or two times each 12 hour shift. By introducing battery-operated wireless temperature sensors clamped on to the outer surface of the pipes, the installation of the sensor unit is simpler and wires can be eliminated. The installation is not time-consuming and can be performed during normal operation of the facility.

The bottom line is that the wireless sensor network approach has been very successful. The estimated increased revenue has been achieved, and the wireless network has been close to 100 % reliable with no loss of sensor data through the 1 ½ year period it has been in operation. Integration with the existing PCDA system was implemented utilizing the serial MODBUS interface of the wireless gateway, and real time monitoring with automatic triggering of alarms when the pressure declines is managed from the PCDA. Because of the immediate success of the pilot installation, several other Oil & Gas producing facilities in the North Sea recently have deployed similar wireless sensor networks.

2.3 Asset and personnel tracking

Keeping track of assets and personnel is getting increasingly important as industrial operations are becoming more and more complex. We see a growing number of new products and concepts for local area tracking of assets and people. These are alternative solutions in applications where public services such as GPS (Global Positioning System) or similar is not a viable alternative. It is common to distinguish between Real-Time Localization Systems (RTLS), where the tag position is being updated in real-time and passive tracking utilizing RFID, in which the tag is detected when passing a checkpoint.
Some industrial application areas involving tracking solutions include:

- Keeping track of containers and goods at supply bases
- Keeping track of expensive tools and parts, as lost and misplaced equipment can interrupt production or slow down planned work. It also contributes to reduce duplicate captures of similar assets
- Keeping track of people in emergency situations, for example by RFID-based counting and identification of people at choke points (meeting points or mustering stations)
- Keeping track of goods and assets in the whole logistics chain, from manufacturer to end-user

The following section provides an example of a planned asset tracking system for simplifying the logistics on an onshore supply base, serving offshore Oil & Gas installations in the North Sea. The example is taken from (Petersen et al., 2008).

2.3.1 Example – Improving container logistics

At a supply base, there are a large number of container movements every day, year round. Keeping track of the physical location of each container is considered a challenge, and requires extensive logistics. In addition, it is essential to know the contents of each container. Keeping track of individual container positions along with container metadata is an application where wireless networks can improve efficiency.
Each container is equipped with an electronic tag, serving as a unique ID. Before the container leaves its origin, the tag information is updated. Using a centralized database, the tag ID can be linked with container specific data.
During transportation, the container passes several checkpoints equipped with RFID readers, enabling tracking of the container on its way towards the destination. Events during transport (customs inspection, reloads etc) is logged online and added into the central database. When the container arrives at the supply base, its presence is detected by either an RFID chokepoint or the plant-wide wireless network. A field operator equipped

with a mobile device, e.g. PDA or laptop computer, can access the metadata of the container by making a request to the central database.
As the container is moved around at the base, its physical position can be monitored by utilizing the positioning capabilities of the plant-wide RTLS-enabled WLAN network. Typically, positioning data is linked with a map of the site, visualizing the position of the container in real-time. When container goods is added or removed, the field operator can update the central database using a wireless mobile device.
In order for the concept of container tracking to be successful, tight integration between the wireless network, the positioning application, the database server, user applications, the Enterprise Resource Planning (ERP) system and mobile devices is required.

3. Requirements

A set of requirements have been identified for the use of wireless technology in industrial applications. Some are of a general nature and adhere to all types of wireless equipment, devices and networks. They are:

- Security – Every mobile communication technology should support mechanisms for securing the information flow and ensure data integrity. As a minimum, link layer encryption comprising 128-bit keys should be a general requirement for industrial applications.
- Mechanical Reliability – For industrial applications, the equipment should be industry-grade with respect to mechanical quality and robustness (IP-rating etc.)
- Certification for Operation in Explosive Areas – In some industries many areas are defined as explosive zones. All equipment for use in these areas must be certified according to the national regulations. In the EU, the law is the ATEX directive.
- International standards – Technologies for wireless communication should be comprised by international standards. This ensures interoperability between equipment from different vendors.
- ISM (industrial, scientific and medical)-bands – To ensure global, license free operation, wireless systems should wherever possible use the international ISM frequency bands for the radio communication.
- Coexistence – Friendly coexistence with other systems operating in the same portions of the frequency spectrum. That is, not cause interference to other systems, and be resilient to interference from other systems.
- User Interface - the user interfaces for wireless systems must be able to provide a simple and intuitive interface for advanced configuration, control and management.
- Cost-effective – Wireless systems must be cost-effective, both in terms of installation and daily operation, compared with wired alternatives.

In addition to these general requirements, a set of specific requirements for each of the identified main application areas for industrial wireless technologies have been worked out.

3.1 Mobile ICT

As mobile ICT involves the widest spans of applications within wireless communication, the requirements naturally become very application dependent. Anyway it is still possible to

identify some requirements related to mobile ICT in industrial settings are of a general nature. These include:

- Security and Authentication – Among the most important issues in IEEE 802.11 networks. As the wireless network commonly represents an extension of the corporate network providing access to backhaul systems, the highest levels of security and authentication mechanisms should be implemented. Security should be employed at both link and network layers (Layer 2 and 3 in the OSI model, respectively), and should preferably be centrally managed. Features such as rotating encryption keys and exchange of certificates through dedicated servers (RADIUS or similar) should be a requirement. For all mobile devices that provide logon and user authentication features, this should be enabled using identities and passwords that can be tracked back to the individual user
- Bandwidth – Industrial WLAN applications commonly require less bandwidth than corporate or consumer market applications (but higher reliability). Medium bandwidth is a general requirement, but this is of course application dependent
- Reliability – IEEE 802.11 networks inherently do not provide the necessary level of reliability to make them suitable for any application of critical nature. A medium to high level of reliability, up to 99 %, is a reasonable requirement for the industry. Reliability can be increased by the use of redundant networks or mesh topologies
- Scalability – Easily scalable as the demands for wireless coverage and/or the number of users increases
- Seamless integration – The backhaul network should be fully transparent to the mobile client. Virtually no difference between a wired and a wireless client from a users' point of view
- Site management – To avoid local configuration and administration of huge numbers of infrastructure components in the wireless network, centralized management, configuration and monitoring of the network should be a requirement
- Simplified and automated work processes – Depending on the application, the introduction of a mobile ICT solution in the industry could have as a requirement that a specific work process should be simplified, for example a routine maintenance task being reduced by a given number of man hours. Another application could set up as a requirement that the new mobile ICT solution should fully automate the former manual task

3.2 Wireless Instrumentation

For wireless instrumentation, and in particular wireless sensor networks, the following requirements have been identified (Petersen et al., 2007)

- Reliability – For general monitoring applications, reliability should be > 99.99 %, e.g. maximum acceptable data loss is 1 sample out of 10,000 samples. Note that even a network with a significant packet loss can achieve 100 % reliability due to retransmissions and redundant paths
- Battery lifetime – For battery operated wireless sensors with a one minute update rate, the battery lifetime should be in excess of 5 years

- Update Rate – Requirements for necessary sensor data update rate should be stated. For IEEE 802.15.4 networks, update rates down to 1 minute is practically achievable. Note the trade-offs between update rate and power consumption
- Simple maintenance routines – Wireless instruments should be designed and installed in such a manner that routine maintenance, e.g. exchange of batteries, can be easily performed
- Transparency to wired systems – From an end-users' (operator) point of view, there should be virtually no difference between a wired instrument and its wireless counterpart
- Integration to control room – Wireless instruments should integrate with existing control and monitoring systems over standard industrial interfaces (field buses etc)
- Security and authentication – The networks should be resilient to both active and passive security threats and attacks (Cayirci & Rong, 2009).

3.3 Asset and personnel tracking

As is the case for mobile ICT, the requirements for asset and personnel tracking depends on the specific usage scenario. The following general requirements should be applicable to most industrial asset and personnel tracking applications:

- Precision – Precise requirements on position resolution and accuracy should be worked out, as this contributes to the premises for which localization technology that should be chosen
- Real-Time – Depending on the application, real-time requirements regarding update of position should be stated
- Infrastructure demands – Asset and personnel tracking solutions can either utilize public infrastructure, existing enterprise infrastructure, or deploy new infrastructure depending on the application and the technology
- Redundancy – Depending on the level of criticality of the application, requirements for fail-safe operation and redundant solutions should be carried out. Keywords are redundant networks, alternative networks and uninterruptible power supplies
- Client side or network side positioning – Depending on the application, the calculations for determining the mobile devices' position should either be carried out on the device itself (which requires computational power), or on a central server located at the backhaul side of the network
- Maintenance free tags – Locatable tags must have a lifetime in the order of several years in order to be maintenance free and cost effective

4. Wireless Technology and International Standards

This chapter provides a survey of the most relevant technologies and international standards for industrial wireless communication. For some applications, solutions from the consumer and office markets are adapted by the industry. However, in many cases this technology, or the equipment itself, does not fulfil the industrial requirements. Modifications, or even redesigns, are therefore often needed in order to enable industrial deployment. The growing demand for industry specific applications of wireless technology

within mobile ICT, wireless instrumentation and asset and personnel tracking has lead to the development of specific technology and international standards for industrial use:

- **Mobile ICT**
  IEEE Std 802.11
- **Wireless Instrumentation**
  IEEE Std 802.15.4, ZigBee, WirelessHART and ISA100.11a
- **Asset and personnel tracking**
  Various RFID standards and the ISO 24730 for Real-Time Localization Systems

4.1 IEEE Std 802.11

The 802.11 working group of the IEEE standards body is responsible for defining and maintaining a set of standards for Wireless Local Area Networks (WLAN). The original (legacy) IEEE Std 802.11-1997 defined three Physical (PHY) Layer specifications and one common Medium Access Control (MAC) specification. Since then further work has been carried out to extend the initial PHY specifications to provide higher data rates, leading to IEEE Std 802.11a and IEEE Std 802.11b, both released in 1999, and IEEE Std 802.11g, released in 2003. In 2007, all the addendums to the legacy IEEE Std 802.11-1997 was merged and published as IEEE Std 802.11-2007 (IEEE 802.11, 2007). The new revision collects many of the changes and amendments performed and published by IEEE 802.11 Task Groups. In addition, the IEEE Std 802.11n (IEEE 802.11n, 2009) was published in 2009.
Table 1 provides an overview of the different 802.11 protocols.

Protocol	Release Date	Frequency (GHz)	Data Rate (Mbps*)
802.11	1997	2.4	2
802.11a	1999	5	54
802.11b	1999	2.4	11
802.11g	2003	2.4	54
802.11n	2009	2.4 and/or 5	600

* *Megabit per second*

Table 1. Overview of the IEEE Std 802.11 protocols

4.1.1 IEEE 802.11 Operation Modes

The IEEE Std 802.11 defines two pieces of equipment, a wireless station/client and an Access Point (AP), which acts as a bridge between the wireless stations and wired networks. The AP acts as the base station for the wireless network, aggregating access for multiple wireless stations onto the wired network.
There are two operation modes in IEEE 802.11, infrastructure mode and ad-hoc mode**.** In infrastructure mode the wireless network consists of at least one AP connected to a wired network infrastructure, and a set of wireless stations. This configuration is called a Basic Service Set (BSS). An Extended Service Set (ESS) is a set of two or more BSSs forming a single sub-network. Ad-hoc mode is a set of wireless stations that communicate directly with one another without using an AP or any connection to a wired network.

4.1.2 The legacy IEEE Std 802.11-1997

The original IEEE Std 802.11-1997 defines operation in the 2.4 GHz band, supporting data rates of 1 and 2 Mbps. The IEEE Std 802.11 divides the 2.4 GHz band into 14 channels, each with a bandwidth of 22 MHz. However, due to national rules and regulations, channel 14 is only available in a select few countries (Japan, Spain), and channels 12 and 13 are prohibited in North American and some Central and South American countries. The centre frequency of the channels are spaced 5 MHz apart, which means that neighbouring channels overlap in frequency.

IEEE Std 802.11-1997 uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CA is referred to as the Distributed Co-ordination Function (DCF). This requires each station to listen for other users. If the channel is idle, the station may transmit. However if it is busy, each station must wait until transmission stops at which time the receiver sends an ACK. Then each station must wait for a time equal to the Distributed Inter-Frame Space (DIFS), plus a random number of slot times for the next transmission in order to avoid collisions over the medium.

4.1.3 IEEE Std 802.11a

The IEEE Std 802.11a (IEEE Std 802.11, 2007) operates in the 5 GHz band, using the same core protocol as the other IEEE 802.11 specifications The 5 GHz band offers the advantage of avoiding the popular and crowded 2.4 GHz band, but the higher frequency reduces the communication range, and makes it more sensitive to interference from walls or other architectural components. This necessitates the use of more access points to achieve comparable coverage to its 2.4 GHz counterparts. IEEE Std 802.11a uses a 52-subcarrier Orthogonal Frequency-Division Multiplexing (OFDM) modulation scheme with a maximum raw data rate of 54 Mbps.

Although IEEE Std 802.11a offers increased bandwidth capacity over IEEE Std 802.11b, it is not widely adopted. It has become difficult to acquire IEEE Std 802.11a AP or PC cards as IEEE Std 802.11b/g have developed as the *de facto* standard for both the consumer and industrial markets.

4.1.4 IEEE Std 802.11b

The IEEE Std 802.11b (IEEE Std 802.11, 2007) is an amendment to the original IEEE 802.11-1997 standard. It supports data rates of 5.5 and 11 Mbps in the 2.4 GHz band by using Complementary Code Keying (CCK) with Quadrature Phase Shift Keying (QPSK) modulation and Direct-Sequence Spread-Spectrum (DSSS) technology.

The IEEE Std 802.11b defines dynamic rate shifting, allowing data rates to be automatically adjusted for noisy conditions. This means that IEEE Std 802.11b devices will transmit at lower data rates (5.5 Mbps, 2 Mbps or 1 Mps) under noisy conditions. When the devices move back within the range of a higher-speed transmission, the connection will automatically speed up again.

4.1.5 IEEE Std 802.11g

The IEEE Std 802.11g (IEEE Std 802.11, 2007) is another amendment to the IEEE Std 802.11-1997. It further extends the maximum raw data rate in the 2.4 GHz band to 54 Mbps. IEEE Std 802.11g hardware is backwards compatible with IEEE Std 802.11b, but the presence of an

IEEE Std 802.11b client in a IEEE Std 802.11g network will significantly reduce the overall data rate of the IEEE Std 802.11g network.
The IEEE Std 802.11g adds a new section to the IEEE Std 802.11 PHY; the Extended Rate PHY Specification (ERP). The ERP adds the data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps to the rates already defined by IEEE Std 802.11-1997 and IEEE Std 802.11b. Of these rates, support for 1, 2, 5.5, 11, 6, 12 and 24 Mbps is mandatory.
Two additional optional ERP-PBCC modulation modes with data rates of 22 and 33 Mbps are also defined. In addition, another optional modulation mode, DSSS-OFDM (Direct Sequence Spread Spectrum-Orthogonal Frequency Division Multiplexing) with data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps is defined.
An ERP device is capable of operating in any combination of available modulations.

4.1.6 IEEE Std 802.11n

The IEEE Std 802.11n (IEEE Std 802.11n, 2009), ratified in 2009, supports operation in both the 2.4 and 5 GHz bands simoultaneously. It is backwards compatible with all three previous IEEE Std 802.11a/b/g protocols. IEEE Std 802.11n opens for a theoretical maximum raw data rate of 600 Mbps. One of the major additions to the IEEE Std 802.11n PHY is the added capability of using 40 MHz channel bandwidth (Perahia & Stacey, 2008). This channel bonding merges two adjacent 20 MHz channels into one single channel. The adjacent channel interference is equal for 20 MHz and 40 MHz operation. As the 2.4 GHz band only has three available non-overlapping channels (channel 1, 6 and 11), this means that IEEE Std 802.11n equipment may occupy 2/3 of the available spectrum.
To handle coexistence and interoperability issues arising when using a 40 MHz channel, the two 20 MHz channels used to form the 40 MHz channel are defined as primary and secondary channels. Control and management (beacons) are transmitted on the primary 20 MHz channel, and legacy 20 MHz devices (IEEE 802.11a/b/g) will use the primary channel for all communication. In addition, the legacy portion of the 20 MHz mixed format preamble is replicated over both 20 MHz channels. Even though 40 MHz channel bandwidth is possible (and allowed) in the 2.4 GHz band, it is not recommended duo to potential coexistence issues with other network and devices operating in the 2.4 GHz band.
IEEE Std 802.11n will also make some improvements to the IEEE Std 802.11 OFDM mechanisms. A short guard interval has been introduced, reducing the guard interval of the OFDM data symbols from 0.8 μs to 0.4 μs. The overall symbol length is reduced from 4 μs to 3.6 μs. The guard interval of the preamble is not modified to ensure compatibility with legacy devices. The short guard interval corresponds to an increased data rate of 11 % compared to legacy devices. An option to make the preamble more efficient is also included in the IEEE Std 802.11n, using a Greenfield preamble. The Greenfield preamble reduces the overhead of the PHY preamble by 12 μs compared to the legacy mixed format preamble. The Greenfield preamble is however not compatible with legacy devices.
Several modifications to the IEEE Std 802.11 MAC layer are done in order to increase the throughput of an IEEE Std 802.11n network. Results from the IEEE 802.11e task group work on Quality of Service enhancements to the IEEE 802.11 family have been added. This includes data burst - where several data packets from a single source are transmitted continuously without pausing between each packet – and immediate block acknowledgement. Other new MAC mechanisms for the IEEE Std 802.11n is reduced inter-frame space (RIFS), where the space/time between two following frames are reduced. For

one-way data intensive applications (i.e. file upload or download), it is also an option to completely remove the inter-frame spacing through data aggregation.
Several additions to the IEEE 802.11n standard make the data exchange more robust than for the legacy standards. The receive diversity enabled by MIMO allows for maximal-ratio combining (MRC), and the possibility of transmitting different bits over separate antennas. With MIMO there is also the possibility of having more antennas than spatial streams. Improved error detection and correction codes are also introduced in IEEE 802.11n, both space-time block coding, and the optional low density parity check (LDPC) codes.

4.2 IEEE Std 802.15.4

The IEEE Std 802.15.4 defines the physical (PHY) and medium access control (MAC) layers for low-rate wireless personal area networks (IEEE 802.15.4, 2006). The standard specifies operation both in the 868/915 MHz band and in the 2.4 GHz band. Two new, optional high-data-rate PHYs in the 868/915 MHz band were introduced in the 2006 revision of the standard.
The IEEE Std 802.15.4 defines a total of 27 channels, numbered 0 to 26. Channel 0 is in the 868 MHz band with a centre frequency of 868.3 MHz. Channels 1 through 10 are located in the 915 MHz band. The channel spacing is 2 MHz, with channel 1 having a centre frequency of 906 MHz. Channels 11 through 26 are located in the 2.4 GHz band. The channel spacing is 5 MHz, with the centre frequency of channel 11 being 2.405 GHz.

4.3 ZigBee

The ZigBee specification (ZigBee Alliance, 2006) defines network and application layers on top of the IEEE Std 802.15.4 PHY and MAC, enabling a low-rate, low power WSN. ZigBee is primarily targeting home automation and consumer electronics applications (Verdone et al., 2008). As a ZigBee network operates on the same static channel (one of the 16 available channels defined by IEEE Std 802.15.4) throughout its entire lifetime, it is susceptible to noise and interference. This has lead to ZigBee not being regarded as robust enough for industrial environments and applications. To combat this, the ZigBee Alliance has created the ZigBee PRO specification (ZigBee PRO, 2007) which is specifically aimed at the industrial market. ZigBee PRO offers both enhanced security features and the ability for a network to change its operating channel when faced with large amounts of noise and/or interference.

4.4 WirelessHART

The HART Field Communication Specification, Revision 7.0 (HART 2007) which was ratified in September 2007, has presented the industry with the first open standard, often referred to as WirelessHART, specifically targeting wireless instrumentation for factory automation. WirelessHART is based on the IEEE Std 802.15.4 PHY, although WirelessHART only defines operation in the 2.4 GHz band.
WirelessHART employs a frequency hopping, multi-hop, mesh network topology, using time-division multiple access (TDMA) for channel access. The network communication is divided into time slots, and each communication link in the network is given its own, reserved time slot in order to ensure contention free utilization of the radio channel. This requires all nodes in the network to be time-synchronized, normally using the gateway as

the master clock. With its self-healing and self-configuration capabilities, the deployment of a WirelessHART network does not require detailed understanding of low level communication and radio propagation aspects (Kim et al., 2008).

4.5 ISA100.11a

The ISA100 standards committee of the International Society of Automation (ISA) is working on a family of standards defining wireless systems for industrial automation and control applications (ISA100 Standards Committee, 2008The first released standard was the ISA100.11a (ISA100.11a, 2009), ratified in September 2009, providing secure and reliable wireless communication for noncritical monitoring and control applications. Critical applications are planned to be addressed in later releases of the standard.

The ISA100.11a is based on the IEEE Std 802.15.4 PHY and MAC. It operates in the 2.4 GHz band, and defines a frequency hopping, multi-hop mesh network. Like WirelessHART, TDMA is used as the channel access method, along with network self-configuring and self-healing algorithms. The ISA100.11a enables a network to carry existing wired fieldbus protocols, allowing existing wired installations to be conveniently converted to a wireless infrastructure, with a transparent data transfer between systems.

The ISA100 has also established a subcommittee to investigate options for the convergence of WirelessHART and ISA100.11a, the ISA100.12. The aim of this committee is to merge the two standards into a single standard which will be merged into a future release of the ISA100.11a.

4.6 RFID – Radio Frequency Identification

The term Radio Frequency Identification (RFID) is used for describing identification technologies where a *reader* identifies one or several *transponders* by using electromagnetic waves. The technology has its conceptual origins from IFF (Identify Friend or Foe) systems used to identify aircraft during World War II.

There are no formal definitions of the concept "RFID – Radio Frequency Identification". However, the following description should cover most aspects of RFID technologies:

> **Radio Frequency Identification - RFID**
> Identification performed by the use of electromagnetic waves. The identification process involves at least one *reader* and one *transponder*, and the process is initiated by the reader generating an electromagnetic signal. Compatible transponders within reach of this signal will make a response, enabling them to be detected and identified by the reader.

According to this description, a transponder should only respond after being interrogated by a reader, indicating that transponders should be completely silent (or "invisible") when not interrogated. This restriction is convenient when it comes to distinguishing traditional RFID technology from other identification solutions based on for instance WLAN and Bluetooth. Note however that some proprietary technologies may be described as RFID solutions without conforming to this principle.

4.6.1 Variants of RFID technology

There are several types of RFID technologies on the market. Depending on how they work and how they are constructed, they can be placed in different categories. The two main distinctions are whether the transponders have internal battery or not, and whether the technology is based on *inductive* or *electric* communication (Finkenzeller, 2003).

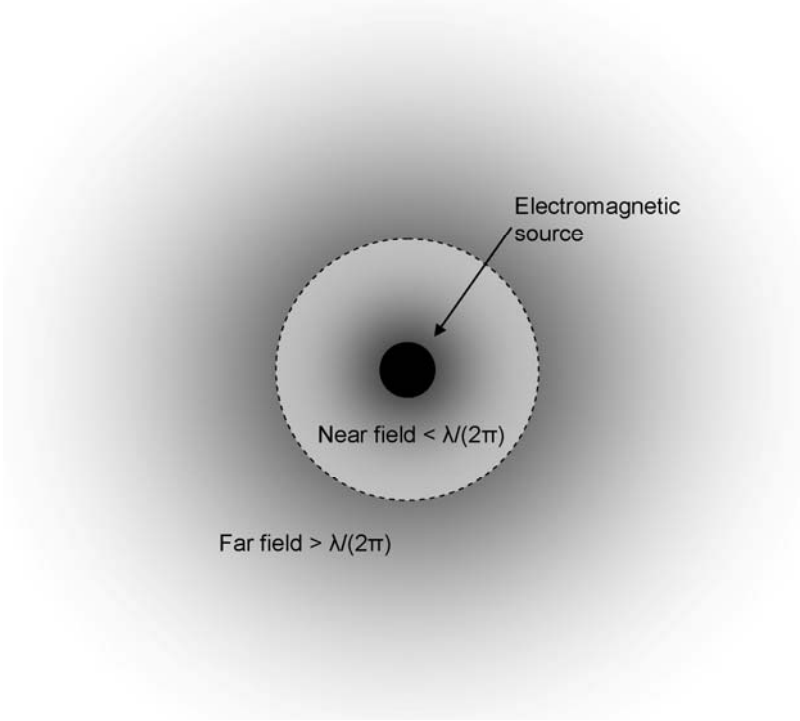


Fig. 2. Illustration of near field and far field (wavelength is denoted by λ)

Active vs. passive transponders

All RFID transponders need energy in order to operate, and many transponder types harvest energy from the reader's electromagnetic field in order to function. Such transponders are called *passive*, as they can only operate when energized by an external energy source. *Active* transponders on the other hand, use an internal energy source (battery) to yield a response. Both technologies have their advantages; while passive RFID transponders, in theory, have no life-time limitations, active transponders can provide much longer read ranges due to their internal power source. A semi-active (sometimes called semi-passive) transponder is a mixture of both active and passive transponders. These do contain a battery, but this battery is only used for internal purposes and not for communication (in which case the transponders would have been classified as active).

Inductive vs. electric communication

RFID technology is either based on *inductive (*magnetic*)* or *electric (*radio-based*) communication*, depending on their operating frequencies (Finkenzeller, 2003). This is due to

the nature of electromagnetic waves, which are created by the magnetic field enclosing the antenna. When a transponder is within a reader's near field (defined as the volume enclosed by a ball with a radius equalling 0.16 times the wavelength of the electromagnetic field), the electromagnetic field has not yet completely formed, and the most efficient way to communicate is by using a coil. Conversely, when a transponder is within a transmitter's far field (that is, outside the near field), the magnetic field is no longer present, and one has to use the electromagnetic field to communicate (as illustrated in Fig. 2). For far field communication, antennas are the most efficient way of receiving and transmitting electromagnetic waves. Note however that the term *antenna* in practice is used for both coils (used for inductive coupling) and "traditional" antennas that are used for electromagnetic communication.

4.6.2 RFID Standards

There are several RFID solutions on the market, of which some are proprietary while others are based on open standards. The most commonly used standards are listed in Table 2.

Frequency	Technology	Standards	Range	Applications
125-135 kHz	Inductive coupling	ISO 11784	< 50 cm	Animal identification
		ISO 11785		
		ISO 14223		
13.56 MHz	Inductive coupling	ISO 14443	< 10 cm	Ticketing, identification and payment
		ISO 15693	< 1 m	
860-960 MHz	Radio backscatter	ISO 18000-6	< 2m	Electronic Product Code

Table 2. Overview of RFID standards

4.7 Real-Time Location Tracking Systems

This section gives a brief introduction to the technology forming the basis for various Real-Time Location Tracking Systems (RTLS). There exist a number of different technologies for determining the position of a device in real-time. As wireless technology increasingly enters the industry, we see a fast growing number of applications utilizing different methods for localization of assets or personnel. In 2006, the International Organization for Standardization, ISO, released the ISO 24730 describing RTLS systems in information technology. The ISO 24730 classifies RTLS into the following:

- Locating an asset via satellite. Accuracy to 10 m
- Locating an asset in a controlled area, e.g. warehouse or similar. Accuracy down to 3 m. Requires local infrastructure
- Locating an asset in a more confined area. Accuracy down to centimetres. Requires local infrastructure
- Locating an asset over a terrestrial area utilizing cell-phone base stations or similar. Accuracy 200 m

In addition, for RFID, two additional methods for locating an object are defined:

- Locating an asset by detecting if it has passed a checkpoint A but not passed checkpoint B

- Locating an asset by the aid of a homing beacon in such manner that a person with a mobile device can find the asset

Observe that the two latter methods are not true RTLS.
In the following section, we give an overview of some of the most widely used technologies for positioning. Each technology has its benefits and disadvantages, depending on the application.

4.7.1 Cell of Origin

In cellular communication systems, e.g. GSM, one simple method to locate a client is by monitoring which base station (cell) the client is associated with. In the case the client is within coverage of several cells, determinate which cell that detects the highest RF signal strength from the client. The cell of origin technique is inaccurate and in general provides a very coarse indication on the clients' position. The position of the client falls within the coverage area of the particular cell. Depending on the cell density and network topology, this area can be large, thus leading to poor determination of position. However, cell of origin is widely used in e.g. emergency call services and is valuable in many applications. If we draw parallels to WLAN networks, the cell of origin principle could be termed as "nearest access point detection".

4.7.2 Received Signal Strength (RSS)

Measuring a clients' RSS at the base station gives an indication of the distance between the two, by the use of lateration techniques. Parameters that must be taken into account are transmitting power, cable losses and antenna gain, and eventually antenna directivity.
Using a suitable mathematical model, the *path loss* between mobile device and access point can be calculated. A common model for indoor propagation at 2.4 GHz is (Cisco Systems, 2008):

$$PL = PL_{ref} + 10 \cdot \log(D^n) + S \quad [\text{dB}] \tag{1}$$

where

PL	is the total path loss between access point and client	[dB]
PL_{ref}	is the reference path loss at a distance equal to 1 m	[dB]
D	is the distance between access point and client	[m]
N	is the path loss exponent, specific to the environment	
S	is the contribution from *shadow fading*	[dB]

Path loss represents the difference between transmitted and received power, and can be seen upon as the signal attenuation throughout the environment. Common factors contributing to signal attenuation are:

- Free space propagation attenuation, equals –6dB per doubling of distance
- Reflections from ground and surroundings
- Diffraction (wave bending effects)
- Scattering (fractional spreading)

The path loss exponent is an empirical parameter specific to the local environment. Typical values for n lie in the range ~2 for open space environments, to >2 in environments with obstructions (Cisco Systems, 2008).
Shadow fading is also related to the environment. The degree of indoor fading varies depending on the number of obstacles present. In a relatively obstructive indoor environment, with many partitions, walls etc, S may be in the range ±7 dB (Cisco Systems, 2008).
The received signal strength P_{RX} is:

$$P_{RX} = P_{TX} - Loss_{TX} + G_{TX} - PL + G_{RX} - Loss_{RX} \quad [\text{dB}] \tag{2}$$

where (all units in dB)

P_{TX}	is the transmitted power
$Loss_{TX}$	is the total loss factors at the transmitter
G_{TX}	is the antenna gain at the transmitter
P_L	is the path loss from eqn. (1)
G_{RX}	is the receiver antenna gain
$Loss_{RX}$	is the total loss factors as seen from the receiver

To calculate the distance D between client (WLAN tag) and receiver, eqn. (2) can be substituted:

$$D = \sqrt{\log^{-1}\left(\frac{P_{RX} - P_{TX} + Loss_{TX} - G_{TX} + PL_{ref} - S + Loss_{RX} - G_{RX}}{-10}\right)} \quad [\text{m}] \tag{3}$$

In a Wi-Fi infrastructure, the base station is the WLAN access point. With one access point, eqn. (3) represents a circle with radius D. The client is assumed to be somewhere on the circumference on this circle. With three or more access points, *tri-lateration* or *multi-lateration* techniques (similar to the methods used for ToA, see section 0) can be utilized to determine the position of the client.
In principle, both clients and access points can measure signal strength. Because of variations in hardware quality and implementation methods among the different WLAN client vendors, the reported signal strengths on the client-side may not be consistent and thus not very reliable. In addition, RSSI detection functionality demands for additional hardware and computational power compared to what's found on a simple WLAN tag.
Leaving the RSSI measurements to the backhaul side of the network is a more robust solution. Most WLAN sites use backhaul equipment (access points etc.) from the same vendor, which yields identical RSSI measurement metrics. A great number of commercial WLAN tracking systems use network side measurements.
RSSI based location utilizes existing WLAN infrastructure, having the advantage that no specialized network infrastructure need to be deployed. From this point of view an RSSI based localization system is attractive with respect to installation cost and complexity.

At the same time, there are several drawbacks with RSSI localization, regarding localization accuracy. Non-ideal RF propagation properties due to anisotropic conditions in the environment is one of the main factors contributing to degraded accuracy. Thus the theoretical path loss model can deviate significantly from the real-world situation. Typical parameters obstructing the path loss model are multipath propagation, radio interference and attenuation (screening from obstacles).

4.7.3 RF Pattern Matching

Pattern matching further refines the RSSI approach for localization. RF pattern matching relies on comparing an objects' current signal strength pattern against a pre-established location database of signal strength patterns collected in the calibration phase. The calibration process (often referred to as the *training* phase) is commonly a time-consuming task involving measuring signal strengths in a large number of pre-determined positions throughout a grid enclosing the coverage area. Once calibrated, the location accuracy within the area can be good in the order of down to a couple of meters. However, as time passes the physical environment is likely to change (equipment, machinery or inventory moves around, climatic conditions change etc) and the accuracy slowly degrades. To accuracy, periodic re-calibration is necessary.

4.7.4 Time-Based Location Tracking Techniques

The common aim for time-based location tracking techniques is, as the name indicates, to give a measure of the distance between the client and one or more base stations utilizing time measures. Two common methods are built on the Time of Arrival (ToA) and the Time Difference of Arrival (TDoA) principles. In this text, the theory behind these positioning principles is taken from (Forssell, B., 1991) and (Cisco Systems, 2008).

4.7.5 Time of Arrival (ToA)

For line-of-sight situations, in particular outdoor environments, the Time of Arrival principle is one of the most accurate methods for determining the position of a receiver. Satellite positioning systems such as the Global Positioning System, GPS (US), Glonass (Russia) and the upcoming Galileo system (EU) use ToA for determining the distance between the base station and the client. For these systems the base station is a space satellite and the client is a mobile receiving unit on the Earths surface.

In ToA, synchronized clocks in the base station and the client are used to measure the time delay between the two. The base station transmits a signal containing information about the exact moment of time t_1 when the transmission occurred. On the client side, this time instant is subtracted from the actual time t_2 and yields a time difference $\Delta t = t_1 - t_1$

Knowing the propagation speed of radio waves c (which is close to the speed of light), the distance r between base station and client can be calculated from

$$r = c \cdot \Delta t = c \cdot \left(t_2 - t_1 \right) \tag{4}$$

Assuming that the exact position of the base station is known, by receiving signals from three (or more) transmitters, the client position can be calculated by performing *triangulation* or *multi-lateration* operations.

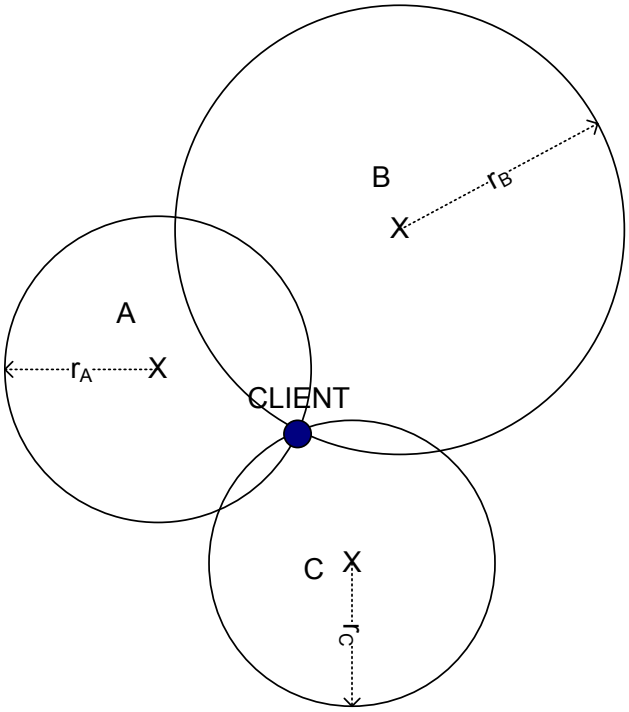


Fig. 3. Triangulation (Tri-lateration)

Using eq. (4), the distance from each base station can be calculated. For every base station - client distance r_n, it is assumed that the position of the client is somewhere along the circle with radius r_n. Signals from three base stations lead to a point of intersection of the three circles, which represents the actual 2D-position of the client. This is illustrated in Fig. 3. For 3D positioning, the client must be within range of at least four base stations. In this case, the intersection point of four constructed spheres represents the 3D-position of the client.

A ToA based positioning system need very accurate and synchronized clocks. To achieve a sufficient level of accuracy, satellite navigation systems use atomic clocks. The client must be time-synchronized with the space satellite. The precision of the time-synchronization is of critical importance to the system performance, as a time measurement error on the scale of a few hundred nanoseconds might cause localization errors in the order of tens of meters.

Furthermore, all calculations for determining the actual position are performed on the receiver side of the network, which demands for sufficient computational power on the client. Additionally, the propagation of radio waves through the atmosphere is exposed to varying delays. This means that the speed of propagation, c, is not constant. For short-range communication this may not be a problem, but in satellite based navigation systems the distance from base station to receiver is in the order 19,000 km to 23,000 km (system dependent), thus varying propagation speeds through the Earths atmosphere will indeed affect the accuracy. Another interfering factor is multipath transmission. The performance of a ToA system is degraded in situations where the received signal is a reflection of the

original signal. Military versions of GPS (and Glonass) transmit on a second frequency to correct for the abovementioned inaccuracies, but receivers are not available for the civilian market. Instead, to increase the localization accuracy for high-precision civilian GPS applications, terrestrial reference stations are used. This technique is referred to as *Differential GPS*, or DGPS.
Achieving true ToA in for example a WLAN network is a challenging task that is difficult to implement with current technology, especially on the client side.

4.7.6 Time Difference of Arrival (TDoA)

When ToA requires synchronized clocks of both base stations and clients, the Time Difference of Arrival (TDoA) approach does not require time synchronization on the client side. Only clock synchronization between base stations is necessary. This is advantageous in the sense that keeping the receiver accurately synchronized with the base stations requires very high precision electronics circuits at the receiver side, leading to expensive client equipment. Also, the commonly high power consumption of ToA-based receiving devices makes them less suitable for battery operation.
In TDoA *relative* time differences between several base stations are calculated. The client transmits a signal which is received by the base stations. No timestamp is sent along, thus the starting time of the transmission is unknown. The signal is picked up by the base stations within range. The relative time differences between the different base stations are then calculated. This technique greatly reduces the complexity of the receiver, as the calculations to determine the clients' position is truly performed at the backhaul side of the network. At least three location receivers are required to perform a 2D location of a client. The mathematical method used to implement TDoA is commonly known as *hyperbolic lateration*. An example of a TDoA based global navigation system is the Long Range Aid to Navigation (LORAN) system. The current generation is the LORAN-C, which serves mainly as a global maritime navigation system with base stations forming different 'chains' around the earth. Although still in service, its use is rapidly declining with GPS as the primary replacement.
Utilizing TDoA for WLAN tracking requires specially designed access points (sometimes referred to as *Location Receivers*). The WLAN client will be the transmitting device, for example WLAN tags transmitting beacons which are picked up by the location receivers. Considering the case with two location receivers *A* and *B*, the time difference of arrival between *A* and *B* is calculated from the following:

$$TDoA_{(B-A)} = \left| T_B - T_A \right| = k \tag{5}$$

The calculated value of $TDoA_{(B\text{-}A)}$ can be used to construct a hyperbola with foci at both location receivers *A* and *B*. The tag position is considered to be somewhere along the constructed hyperbola, at a distance *k(c)* meters from the two foci points. All possible locations of the mobile device can thus be represented by:

$$\left| D_{XB} - D_{XA} \right| = k(c) \tag{6}$$

The actual position is represented by a point along the hyperbola. With only two location receivers, no further determination of the exact position can be calculated. Adding a third Location receiver *C*, a second hyperbola can be constructed, e.g. between *A* and *C*.

$$TDoA_{(C-A)} = \left| T_C - T_A \right| = k_1 \tag{7}$$

Again, the position of the mobile device now can be assumed to be somewhere along this second hyperbola, at a distance $k_1(c)$ meters from the foci points of *A* and *C*. Thus:

$$\left| D_{XC} - D_{XA} \right| = k_1(c) \tag{8}$$

The actual position of the mobile device (WLAN tag) can be represented by the intersection point of the two hyperbolas, as illustrated in Fig. 4.

Like ToA, there might be situations where there exists more than one possible solution for the mobile devices' position. In such cases, a fourth base station is needed in order to perform TDoA *hyperbolic multi-lateration.*

TDoA perform best in outdoor environments with little multipath propagation. Also, semi-outdoor environments such as stadiums, logistics sites etc can often achieve good localization accuracy using TDoA. For indoor environments, TDoA is best suited in buildings that are relatively large and open-spaced. In narrow, crowded indoor areas TDoA suffers from reflection and scattering issues. Increasing the bandwidth of the TDoA signal helps improve the performance. The 2.4 GHz implementation commonly used in the WLAN version of TDoA, is described in ISO 24730. The coding used is BPSK/DSSS (Binary Phase Shift Keying/Direct Sequence Spread Spectrum) with a designated bandwidth of 60 MHz. This allows for improved performance in environments with multipath propagation.

Following the TDoA approach for localization in WLAN networks requires the introduction of specialized network infrastructure alongside with the existing access points. Some vendors provide units that have integrated both WLAN access point and location receiver into the same physical enclosure. Upcoming products feature both TDoA tracking functionality and ordinary IEEE 802.11 WLAN access integrated onto the same electronic circuits.

Table 3 summarizes the different tracking technologies introduced in this section.

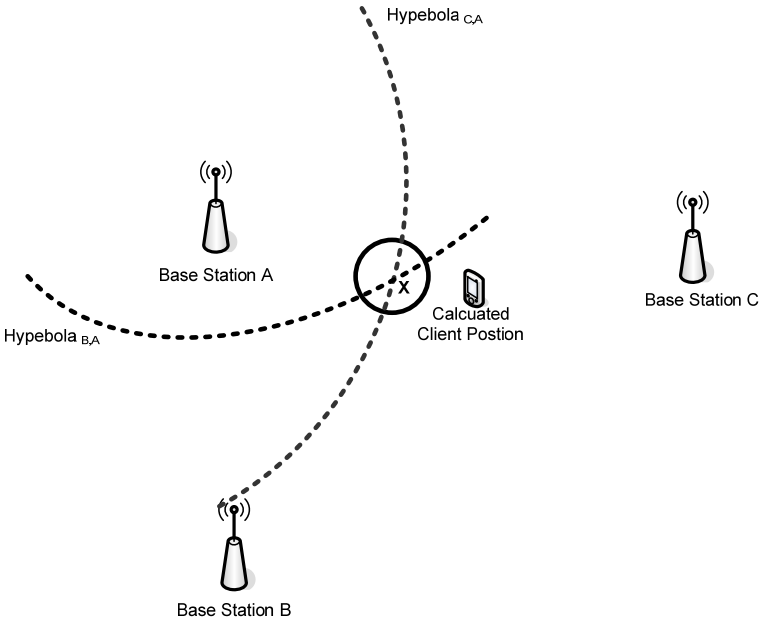


Fig. 4.Time Difference of Arrival (TDoA) with three base stations A, B and C.

Principle	Typical System	Resolution	Line of Sight	Infrastructure cost
Cell of Origin	Cellular communication (GSM, WLAN, etc)	Poor	No	Very low
Received Signal Strength	WLAN networks	Medium	No	Low
RF Pattern Match	WLAN networks with add-ons	Medium to Good	No	Medium
Time of Arrival	GPS or similar	Very good	Yes	Low, but expensive clients
Time Difference of Arrival	Maritime nav. systems, enhanced WLAN networks	Good to Very good	Yes	High

Table 3. Overview of some radio based technologies for real-time localization

5. Future Trends

The basic technologies for wireless communication are already established and proven robust enough for a number of industrial applications, and the number of wireless applications is growing exponentially. But the fact that different wireless suppliers until recently have come up with different devices for different applications, often employing

proprietary radio technologies and communication protocols, have lead to a significant increase in options and complexity for end users. Key questions have emerged in the end users mind about the feasibility of using these devices. This has lead to a growing effort on the standardization of technologies, and this work is still in progress as some aspects of the technology are still are not mature enough for industrial deployment.

For wireless technology development in general, different radio technologies with respect to encoding, channel modulation and protocols, each optimized for the respective application area, is likely to be the case also in the coming years. Related to the OSI Reference Model, it is assumed that the two lowest layers (the physical layer and the data link layer) still need to be service-specific also in the near future. The above layers, however (again referring to the OSI reference model), should be subject to standardization of different wireless technologies. The approach for a common middleware ensures total integration and full flexibility among different wireless technologies individually tailored for different services.

Within the field of wireless instrumentation, several standards have already been established: IEEE 802.15.4, ZigBee, WirelessHART and the ISA100.11a. Currently, these standards are targeting non-critical monitoring and control applications, as there still exists challenges related to the core technology when it comes to wireless process control. As the properties of the wireless medium is of a more stochastic and dynamic nature with respect to interference, transfer delay and service availability compared to wired field buses, issues around reliability and real-time requirements still need to be solved for wireless technology to be a viable option for process control. The IEEE Std 802.15.4 for wireless sensor networks is probably not sufficient for process control applications in its present form. For example, in IEEE 802.15.4 high data reliability is achieved through dynamic routing paths, leading to unpredictable transfer delays. One approach to provide the required quality of service level for wireless process control is to develop a new set of middleware on top of existing base technologies (Chen, D et al., 2004). Others are looking into the development wireless versions of established (wired) industrial communication field buses. More effort needs to be put into the research on wireless process control, with close collaboration between the academia and the industry.

For mobile ICT applications, the IEEE 802.11 family of standards have become the *de facto* standards for wireless networking. For this technology, the future trends will be in the direction of higher bandwidth, extended range, heightened security and increased reliability and robustness in challenging environments. Some of these improvements will be present in the upcoming IEEE 802.11n, which is scheduled for ratification in 2010.

Within mobile ICT and wireless technology for non-critical applications, there still exist some challenges related to the development of intelligent usage areas. More effort should therefore be put into the research on such topics, among them:

- Simplifying work processes
- Automation of routine work operations
- Man-Machine interfaces
- Seamless integration and accessibility to backhaul systems
- Applications to improve Health, Safety and Environmental aspects
- Condition monitoring and maintenance
- Production optimization
- Operational forecasting and prediction
- From decision support to true remote operations

To achieve this, uniform semantics and ontologies across the industry need to be worked out. The time for internal concepts and solutions specific to each organization has definitely passed as a result of globalization in an increasingly complex world. Future requirements for the use of wireless technology in industrial applications will be in the direction of open, standardized solutions which allow full flexibility and compatibly across companies, industries and geography, while at the same time avoiding dependencies to specific vendors. An initiative to standardize the technology and semantics for RFID in the Oil & Gas industry has been initiated by The Norwegian Oil Industry Association (OLF). The project is lead by OLF and the Norwegian research organization SINTEF, with participants from all major license holders on the Norwegian continental shelf. The goal of the project is to investigate Oil & Gas specific requirements for RFID-solutions, and to create a guideline covering aspects such as technology, data capture, communication and system integration within different business applications relevant to the Oil & Gas Industry. Similar initiatives for other technologies and applications will be a necessity to achieve the vision that, in the coming years, wireless technology will be a major and fully integrated part of what is increasingly referred to as "Digital Ecosystems" (DES). A digital ecosystem is a distributed, adaptive, open socio-technical system with properties of self-organization, self-healing, scalability and sustainability, inspired by natural ecosystems (Boley, H. & Chang, E.,2007).

6. References

Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). A Survey on sensor networks, *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, ISSN 0163-6804.

Boley, H. & Chang, E. (2007). Digital Ecosystems: Principles and Semantics, Proceedings of the Inaugural IEEE-IES Digital Ecosystems and Technologies Conference, pp. 398-403, ISBN 1-4244-0470-3, Cairns, Australia, Feb. 2007.

Carlsen, S.; Petersen, S.; Skavhaug, A. & Doyle, P. (2008). Using Wireless Sensor Networks to Enable Increased Oil Recovery, *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1039-1048, ISBN 978-1-4244-1505-2, Hamburg, Germany, Sept. 2008.

Cayirci, E. & Rong, C. (2009). *Security in Wreless Ad Hoc and Sensor Networks*, John Wiley and Sons, ISBN 978-0-470-02748-6, Chippenham, Wiltshire, Great Britain.

Chen, D; Nixon, M.; Aneweer, T.; Shepard, R. & Mok, A. (2004). Middleware for Wireless Process Control Systems, *Architectures for Cooperative Embedded Real-Time Systems Workshop*, 2004.

Cisco Systems, (2008). *Wi-Fi Location Based Services 4.1 Design Guide*, Text Part Number: OL-11612-01, Cisco Systems.

European Parliament and the Council, "Directive 94/9/EC", 1994.

Finkenzeller, K. (2003). *RFID Handbook 2nd Edition*, John Wiley and Sons, ISBN 0-470-84402-7, Chippenham, Wiltshire, Great Britain.

Forssell, B. (1991). *Radionavigation Systems*, Prentice Hall Europe, ISBN 978-0-1375-1058-0, Oslo, Norway.

HART Field Communication Protocol Specifications, Revision 7.0, 2007, HART Communication Foundation, Austin, Texas.

IEEE 802.11-2007, IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007, IEEE Computer Society, Washington, DC.

IEEE 802.11n, IEEE Draft Standard for Information Technology – Telecommunications and information exchange between systems – Local and Metropolitan Area Networks – Specific Washington, DC.

IEEE 802.15.4-2006, IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 2006, IEEE Computer Society, Washington, DC.

ISA 100.11a – 2009, Wireless Systems for Industrial Automation: Process Control and related Applications, 2009, ISA 100 Standards Committee, USA

ISA100 Standards Committee (2008). *The ISA100 Standards – Overview and Status*. Presented at the ISA EXPO 2008, Houston, Texas, USA.

Kim, A. N.; Hekland, F.; Petersen, S. & Doyle, P. (2008). When HART Goes Wireless: Understanding and Implementing the WirelessHART Standard, *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation 2008*, pp. 899-907, ISBN 978-1-4244-1505-2, Hamburg, Germany, Sept. 2008.

Perahia, E. & Stacey, R. (2008). *Next Generation Wireless LANs*. Cambridge University Press, ISBN 978-0-521-88584-3, Cambridge, United Kingdom.

Petersen, S.; Doyle, P., Aasland, C. S.; Vatland, S.; Andersen, T. M. & Sjong, D. (2007). Requirements, Drivers and Analysis of Wireless Sensor Networks for the Oil & Gas Industry. *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 219-226, ISBN 978-1-4244-0825-2, Patras, Greece, Sept. 2007.

Petersen, S.; Carlsen, S. & Skavhaug, A. (2008). Layered Software Challenge of Wireless Technology in the Oil & Gas Industry, *Proceedings of the 19th IEEE Australian Conference on Software Engineering*, pp. 37-46, ISBN 978-0-7695-3100-7, Perth, Australia, March 2008.

Verdone, R.; Dardari, D.; Mazzini, G. & Conti, A. (2008). *Wireless Sensor and Actuator Networks*, Elsevier Academic Press, ISBN 978-0-12-372539-4, Great Britain.

Yu, Q.; Xing, J. & Zhou, Y. (2006). Performance Research of the IEEE 802.15.4 Protocol in Wireless Sensor Networks, *Proceedings of the 2nd IEEE/ASE International Conference on Mechatronic and Embedded Systems and Applications*, pp. 1-4, ISBN 0-7803-9721-5, Beijing, China, Aug. 2006.

ZigBee-2006 Specification, 2006, ZigBee Alliance, San Ramon, California.

ZigBee PRO Specification, 2007, ZigBee Alliance, San Ramon, California.

5

Rapid application development for wireless sensor networks

Mohammad Mostafizur Rahman Mozumdar
and Luciano Lavagno
Politecnico di Torino, Torino
Italy
Laura Vanzago
STMicroelectronics, Milano
Italy

1. Introduction

In the last decade, the landscape of wireless sensor network (WSN) applications has been extending rapidly in many fields such as factory and building automation, environmental monitoring, security systems and in a wide variety of commercial and military areas. Advancements in microelectro-mechanical systems and wireless communication have motivated the development of small and low power sensors and radio equipped modules which are now replacing traditional wired sensor systems. These tiny modules usually called "motes" can communicate with each other by radio and act like as neurons to collect information from the environment. Platforms for WSNs, including processors, sensors, radios, power supplies, operating systems and protocol stacks, are almost as diverse as the application areas, with only a few standards (e.g. TinyOS (Levis et al., 2004) and the ZigBee (2006) protocol), which are still far from being universally recognized and truly interoperable.

Application development for WSNs is quite challenging, because in principle it would require both detailed knowledge of the application area and of the available hardware and software platforms. Moreover, design aids, in the form of both functional simulation, power and performance analysis and on-target debugging are still very rudimentary. Many hardware and software platforms include only LEDs as a debugging aid.

The available functional analysis packages, such as TOSSIM (Levis et al., 2003) for debugging of TinyOS application, OmNet (1992) and NS-2 (2001), fall into two main categories. One is very platform- and OS-specific (such as TOSSIM), and provides essentially a binary API to model the OS and the motes, with limited facilities for re-using existing channel models, tracing, collecting statistics and so on. The other are generic network simulators (such as OmNet, NS, etc.), sometimes enhanced with models tailored to the radios and channels used by WSNs. Both have significant drawbacks when it comes to complex application development. The first group makes it virtually impossible to port an application to a different platform (e.g. from TinyOS to MANTIS (Bhatti et al., 2005) or to a

ZigBee compliant platform or vice versa). The second group still leaves a lot of detailed platform-dependent code to be developed and debugged. Integrated use of a network simulator followed by a platform simulator is the most commonly used path, but still requires one to port code between a number of environments. Moreover, in case a bug is found at the end, one has to resort to led-based debugging, which is extremely time consuming.

In order to solve these problems, we wanted to be able to model the application using high level abstractions, and simulate it using configurable and realistic topologies for the network itself. Then we wanted to be able to automatically generate code for several target operating systems. In this chapter, we present a framework (Mozumdar et al., 2008a) for modeling, simulation and automatic code generation of sensor network applications based on MathWorks (1984) tools. In our framework, applications can be modeled using Stateflow state charts (SF. 2009) (and Simulink block diagrams, even though StateCharts were the best tool for the application we considered as case study). Then the application developer can configure the connectivity of the sensor network nodes and can perform behavioral simulation and functional verification of the application. After modeling and simulation, this framework can generate the complete application code for several target operating systems from the simulated model.

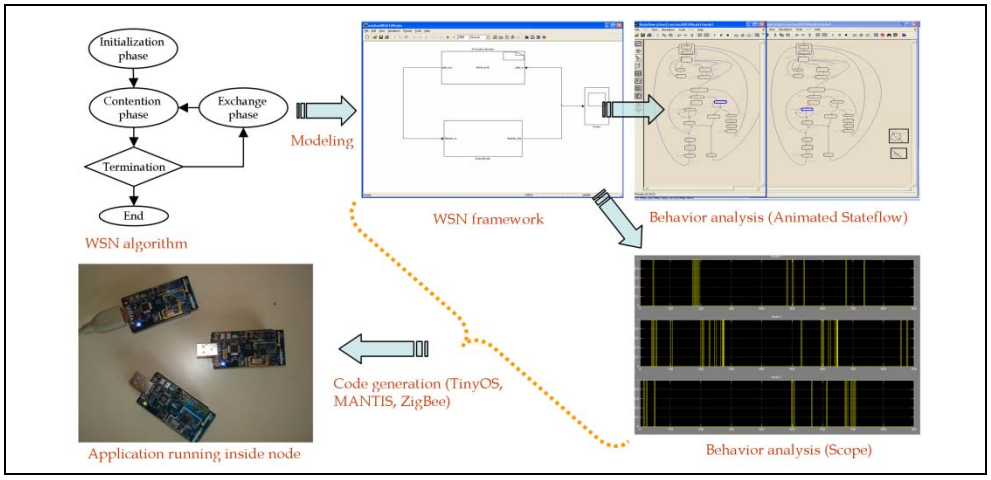


Fig. 1. A complete view of the framework

The application developer can thus use the broad variety of debugging and analysis tools provided by MathWorks, such as animated state chart displays, scopes, plots, as well as exploit a large number of available pre-designed Simulink blocks. To the best of our knowledge, this is the first time that a framework of this sort has been developed and tested. A complete view of the whole framework is depicted in figure 1.

While working on code generation for the various kinds of target platforms, we also identified a coding style for functions written in ANSI C that maximizes the ease of porting the code, especially if coupled with the basic platform abstraction API that we developed.

In this chapter, we use as example target platforms TinyOS, MANTIS and the Ember implementation of the standard ZigBee, since they provide very different programming

models and abstractions (e.g. non-preemptive scheduler with split-phase coding versus multi-threaded kernel). Hence they are maximally different representatives of the programming platforms used by WSN developers.
In (Cheong et al., 2005), a graphical development and simulation environment for TinyOS-based applications called *Viptos* is described. *Viptos* provides graphical development and interrupt-level simulation of actual TinyOS programs, with packet-level simulation of the network. It also allows the developer to use other models of computation available in Ptolemy II (Eker et al, 2003) for modeling various parts of the system. To model an algorithm using *Viptos*, the users are bound to code it for TinyOS, which implies that the user should have sufficient knowledge of TinyOS. In our framework, the users can model the application by using Stateflow and need not have any knowledge of TinyOS, MANTIS or ZigBee. In short, our framework provides more freedom by decoupling the application from the platform and also supports several platforms for code generation.
In (Vieira et al.2005), the authors describe a visual development framework for multi-platform wireless sensor networks, which is capable of generating application code for TinyOS and Yet Another Tiny Operating System (Yatos) (Almeida et al. 2003). This tool supports only code generation of the developed model for the WISDOM (Vieira et al. 2005) framework and it does not support functional verification of the designed model. Here also the model development is biased to TinyOS and Yatos, since these two target platforms share the same component based programming style.
The idea of generating WSN application code from a single higher level abstraction has also been demonstrated in (Abdelzaher et al., 2004, Gummadi et al., 2005, Newton & Welsh 2004, Bakshi et al., 2005) using functional and macro-programming. All these approaches introduce new programming languages, while in our case we advocate either to use a specific programming style in C, or to use an existing well-known graphical language (Stateflow). Although the approaches listed above introduce higher level abstractions, they did not propose a methodology to generate application code for multiple software platforms (all of these approaches generate application code only for TinyOS). In this chapter, we identified a single programming style (Mozumdar et al, 2008b) that is compatible with most kinds of WSN software platforms (e.g. MANTIS, TinyOS and Zigbee).

2. Methodology

The complete framework for modeling, simulation and automatic code generation is depicted in figure 2. The WSN algorithm (application, middleware or device drivers) will be at first modeled by using Simulink and Stateflow blocks. We have designed blocks that specifically help WSN modeling such as the *sensor node* and *communication medium* described later. These blocks are completely parameterized and can be used for model development like usual Simulink blocks. *Sensor node* blocks are connected to the *communication medium* block which provides a mechanism for the application developer to define the connectivity between the nodes in sensor network.

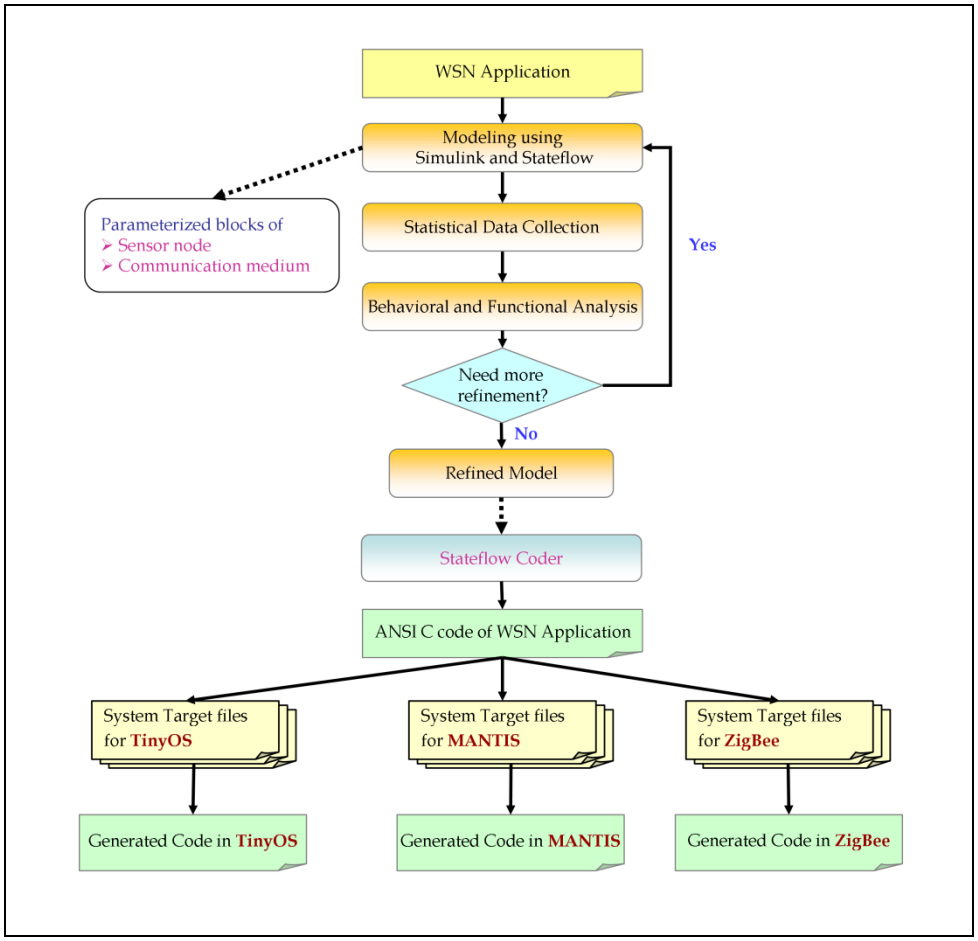


Fig. 2. Framework for modeling, simulation and code generation of WSN application

The *communication medium* block is implemented in C, so it can be modified to reuse any existing channel and connectivity models. The *sensor node* contains mainly a timing generator, a random number generator, and a parameterized Stateflow block which actually implements the application running inside each single node (shown in figure 3). The application block is a library object and each *sensor node* contains an instance of it. Therefore, every node of the framework is running an independent copy of same algorithm. It is of course also possible to model sensor networks having different algorithms running in different nodes. In that case, one needs to create a small Stateflow library and instantiate objects from it as needed. To model a new sensor network application based on this framework, the application developer only needs to modify the template algorithm implementation and set the connectivity of the nodes in the *communication medium* block. Then simulation can be started and statistical data can be collected using animated state charts, scopes and displays to perform functional analysis of the algorithm. The algorithm

implementation can be refined if the analysis of the results suggest to do so. Eventually the developer will get a refined model which represents the desired behavior.

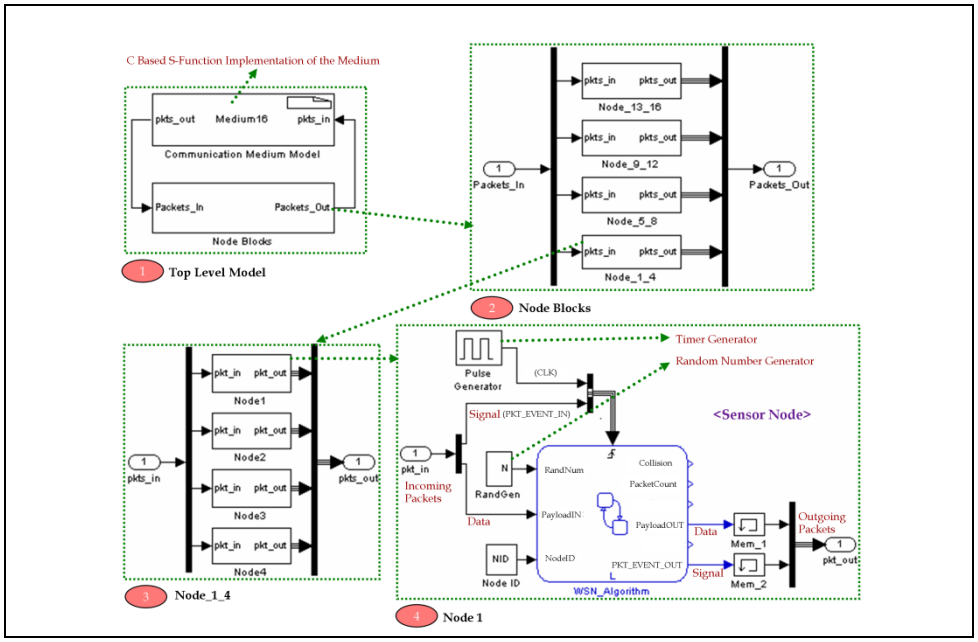


Fig. 3. A simple simulation framework

The next step will be to generate code automatically for TinyOS, MANTIS or ZigBee from the Stateflow representation of the algorithm, using a customization of Stateflow Coder (SF. 2009) which can generate ANSI C code for Stateflow blocks. In order to adapt the generated ANSI C code to the target operating system, Target Language Compiler (TLC) (RTW. 2009) scripts are used. TLC provides mechanisms by which one can generate platform specific code by taking sections (such as includes, defines, functions, etc) from ANSI C code and also by adding custom code for the target platform. In order to ease platform independent development, we provide a set of generic library functions which can be used from Stateflow to access platform specific operating system functionalities (such as *led_toggle*, *led_on*, *led_off*, *sendPacket*, *receivePacket*). From the Stateflow implementation perspective, the application developer does not have to think about the actual implementation of these generic functions in TinyOS, MANTIS or in ZigBee, since they have been implemented in the TLC library and can be targeted to any of the supported operating system and hardware nodes. By using TLC scripts (which are also called *System Target Files*), the developer now can generate automatically a TinyOS application (composed of .nc, .h and makefiles) or a MANTIS application (composed of .c, .h and makefiles) or a ZigBee end device application (also composed of .c, .h and configuration files), and then can compile and execute them for the target platform without any modification.

3. A Simple Simulation Framework

We will now demonstrate a simple sensor network model (shown in figure 3) that has been designed based on our framework components (*sensor node*, *communication medium*). In this model, we consider sixteen sensor nodes, all connected to the communication medium block to form a sensor network. At the top level, the model has two major components:-

3.1 Communication Medium Model

This block contains the medium logic and also models the connectivity between nodes. The logic of the communication medium block is implemented by a C based S-Function (MathWorks. 1984), which contains a (parameterized) 16x16 matrix to define the connectivity of the nodes in the sensor network (shown in the figure 4). In a synergistic research effort, we are also working on a library of radio channel and protocol stack modules at different levels of abstraction (bit, packet). For example in figure 4, node 1 (row 1) is connected to nodes 3, 10, 12 and 15. Packets are the inputs and outputs of the communication medium block, where incoming packets from the nodes will be at first processed by the medium logic and then fed to the appropriate nodes based on the connectivity setup of the sensor network. In this chapter, we consider a very abstract model describing a simple medium logic which at any point of time computes the input (packet) of a node as the summation of outputs (packets) of nodes connected to it.

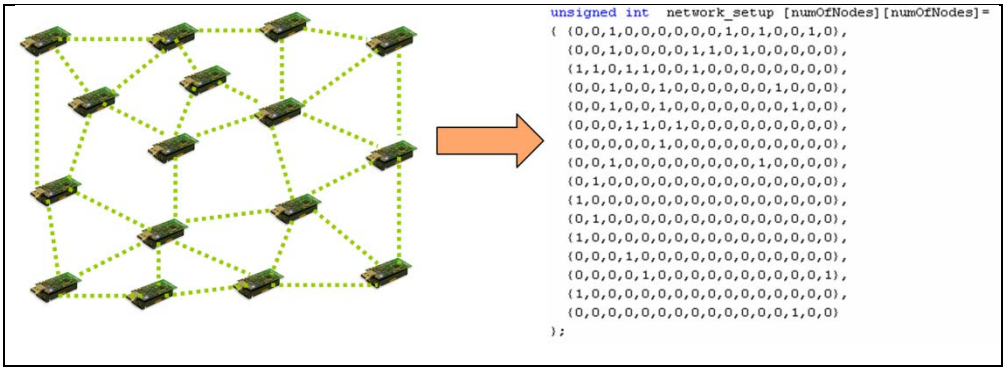


Fig. 4. Connectivity matrix for the 16 nodes sensor network

3.2 Node Block

This block contains sixteen nodes as shown in figure 3. The individual node model is fully parameterized and contains mainly a timer generator, a random number generator and a Stateflow application block. The timer is used for generating *CLK* events for the algorithm running inside the Stateflow block. Incoming and outgoing packets of nodes consist of *data* and *signal* information.

The *data* field contains the payload of the packet and *signal* (which triggers the Stateflow block) generates a packet arrival event which is processed by the Stateflow algorithm inside. The application developer now can perform functional analysis of the algorithm and modify it based on execution data provided by Simulink and Stateflow. In this example, we have

shown a framework of sixteen nodes but the user can easily design a network with a larger number of nodes by slightly modifying the sensor node and communication medium blocks.

4. Multi-Platform Code Generation

After functional analysis of the algorithm, the next step is to generate application code automatically for the target operating systems. For WSN application development one currently has two options, either with a research WSN operating systems solution (such as TinyOS, MANTIS, Contiki (Dunkels et al. 2004), FreeRTOS (Barry, 2003), etc.) or with the ZigBee industry standard. Most of the operating systems are built on a very lightweight event based mechanism (such as TinyOS, Contiki, etc.), however some use a more traditional thread based model (such as MANTIS). On the other hand, ZigBee only defines some layers of the WSN protocol stack and it does not cover the operating system and its libraries. Application development is done on the top of software platforms where user code calls non-standard APIs to interact with the platform. Such heterogeneity of software platforms seriously hinders application porting, and motivated us to consider look for different software platforms used in the sensor network domain and to look for possibilities to port code between them. In this context, we have chosen three software platforms that cover a broad range of programming styles. The candidates are TinyOS (event based), MANTIS (thread based) and ZigBee (an event-based industrial standard).
Our approach is to model sensor network application independent of the platform by using Stateflow, and then automatically generate platform specific application code from that abstraction. We will illustrate the flow by taking a simple WSN application and modeling it in Stateflow (as explained in the next section). For this goal, we designed generic functions that are used to bridge between the model-generated code and the underlying platform (TinyOS, MANTIS and ZigBee).
When developing an application on a platform like those described above, one must consider the services that it provides, in particular:

- the tasking and synchronization models,
- the libraries implementing frequently used functions.

If we consider the first aspect, TinyOS and ZigBee are reasonably similar, because they do not provide a preemptive task abstraction. Hence lengthy library calls cannot be implemented synchronously (by calling and waiting), but must be implemented asynchronously, by splitting them into a request and a response. This splitting allows one to write extremely efficient code, since context swapping can be implemented simply by the interrupt handling hardware. However, writing code in this style is more tedious, since it forces one to save and restore "permanent" state information by hand. MANTIS on the other hand offers a more traditional multi-tasking (to be more precise, multi-threading) environment, which is more familiar and friendly to programmers.
In order to write portable code with these different tasking models, we had to resort to a "reentrant state machine" programming paradigm, where the user code for an application is written as a single procedure, which can be called and return as part of a single "reaction" to external events. Fortunately this programming paradigm is supported by code generation tools for *synchronous reactive* models (Halbwachs, 1993), (e.g. including the StateChart model supported by Stateflow) which:

- Provide the programmer with a procedural abstraction giving the illusion of several concurrent threads of code, all running synchronously with respect to each other and hence all fully deterministic,
- Generate a reentrant state machine code that can be run in a non-preemptive environment.

This programming style is perhaps less efficient than the "native" split-phase programming mode of TinyOS and ZigBee, because it requires one to save the FSM state. But it makes control much more explicit and easy to follow than code in which the "state" is implicitly kept by the signaling between cooperating software and hardware components.

In the next section we introduce a simple WSN application that will be used as an example for porting code between WSN platforms. Sections 6, 7 and 8 describe techniques to port the application in MANTIS, TinyOS and ZigBee respectively. In these sections, we also provide a short description of the platform itself. The code writing approaches that we will explain in the following sections can also be automated by scripts in a model-based design context (for example by TLC scripts). The application code of MANTIS, TinyOS and ZigBee is very different from each others, so the script performs the following tasks to generate platform specific code:

- Copy into the target files the platform specific application independent base code including a type conversion header file that will convert all C types to platform specific types and platform specific implementations of the library functions.
- Generate platform specific application files by taking different sections (such as includes, defines, functions, etc) from the C code generated from Stateflow.
- Generate *make* or *configuration* files for each platform.

5. A Simple WSN Example

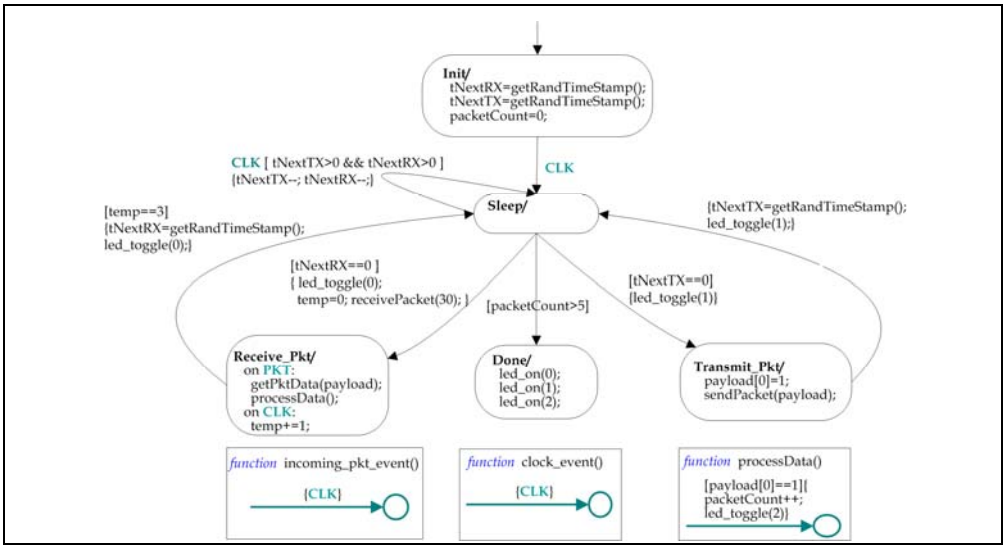


Fig. 5. Stateflow of the simple WSN application

In this section, we will describe a simple WSN application to illustrate application development in MANTIS, TinyOS and ZigBee. This simple application contains most typical ingredients of sensor network applications such as transmitting, receiving, processing of packets and sleeping. In this application example, we do not include a sensing task, but the corresponding development problems are covered by other functionalities, such as incoming events and processing data (which are included in our simple application).
The application transmits and receives packets randomly until it receives six packets, then it stops communications and turns on all LEDs of the node. We also performed more extensive application modelling in this style, but for simplicity we use this very simple application. The Stateflow model of the application is shown in Figure 5.
PKT and *CLK* are external inputs of the algorithm. The *PKT* event is generated after receiving a packet and the *CLK* event is generated when the periodic timer expires. Here, the periodic timer is set to generate a *CLK* event every 10ms. The application starts by initializing the next receiving (*tNextRX*) and transmitting (*tNextTX*) timestamps. To set these timestamps, it calls a library function *getRandTimeStamp* which returns a random number. Then it sets the number of received packets to zero. At the next *CLK* event, the application moves to the *Sleep* state from the *Init* state. In the *Sleep* state, the receiving and transmitting timestamps will be decremented by one at every occurrence of the *CLK* event. At the expiration of the transmit timestamp, the algorithm will make a transition to the *Transmit_Pkt* state and toggle led 1. In this state, it sets the first byte of the payload to 1 and sends the packet by calling library function *sendPacket*. After transmitting the packet, the application makes a transition to the Sleep state, sets the next transmission time-stamp and toggles led 1. In the same way, when the receiving time-stamp expires, the algorithm makes a transition from the Sleep state to *Receive_Pkt* state and it calls the *receivePacket* function to configure the radio in receiving mode for a specified duration (in this case 30ms). In the *Receive_Pkt* state, the algorithm waits for the *PKT* and *CLK* events. After receiving a *PKT* event, it calls library function *getPktData*, which copies the packet data field into a local variable (payload). Now the algorithm calls a local function *processData* where it checks the first byte of the packet data and if it is equal to 1, then it increases the received packet counter and toggles led 2 to give us a visual indication of successful reception of a packet. After expiration of the receiving time slot, the algorithm makes a transition to the *Sleep* state from the *Receive_Pkt* state. While making the transition, it sets the next receiving timestamp and toggles led 0.
In this manner, the algorithm makes transitions between the *Sleep*, *Transmit_Pkt*, and *Receive_Pkt* states until in the *Sleep* state it notices that the number of received packets is greater than five. Then it makes the final transition to the *Done* state where it turns on all three LEDs and stops all communications to the external world.
This simple application, just like many protocol components and WSN applications, can be conveniently modeled as a state machine, either written directly in C/C++ or generated (by Real Time Workshop) from the Stateflow model as shown in Example 1. Interactions of the state machine with the rest of the platform (HW and protocol stack) are dependent on the underlying software architecture. In this case, the incoming events are *CLK* and *PKT* and the outgoing actions are sending a packet (*sendPacket*), setting the radio in listening mode for certain amount of time (*receivePacket*) and switching the leds on the board (*led_toggle, led_on*). Handling these incoming events and outgoing actions depend on the underlying software platform while the rest of the implementation of the state machine remains mostly the same.

In the next sections, we will show how to port this C implementation of the state machine in MANTIS, TinyOS and ZigBee respectively.

6. MANTIS

MANTIS is a light-weight multi-threaded operating system that is capable of multi-tasking on energy constrained distributed sensor networks. The scheduler of MANTIS supports thread preemption which allows the operating system to switch between active threads without waiting. So the responsiveness of the operating system to critical events can be faster than in TinyOS which is non-preemptive. The scheduler of MANTIS is priority-based with round robin. The kernel ensures that all low priority threads execute after the higher priority threads. When there is no thread scheduled for execution, the system moves to sleep mode by executing the idle-thread. Kernel and APIs of MANTIS are written in standard C.

```
void state_machine(void)
{If (for_the_first_time) {
  current_state = IN_Init;                  // Storing the current state
  tNextRX = getRandNumber();        // Generic function to get random number
  tNextTX = getRandNumber();
  packetCount = 0;
}else{
  switch(current_state) {
  case IN_Init:
    if(incoming_event== event_CLK)  // Handling CLK event
      current_state =  IN_Sleep; break;
  case IN_Sleep:
    if((incoming_event== event_CLK) && ((tNextTX > 0) && (tNextRX > 0))){
      tNextTX--;tNextRX--;  current_state = IN_Sleep;
    }else if (tNextRX == 0) {
      led_toggle(0); temp=0;                 // Generic function to toggle led
      receivePacket(30);                     // Generic function to receive packets
      current_state =  IN_Receive_pkt;
    }else if(packetCount > 5) {
      current_state = IN_done; led_on(0);led_on(1);led_on(2);
    }else if (tNextTX == 0)){
      led_toggle(1); current_state = IN_Transmit_pkt;  payload[0] = 1;
      sendPacket(payload);              // Generic function to send packet
      }
    break;
  case IN_Receive_pkt:
    if(temp == 3){
      tNextRX =  getRandNumber();  led_toggle(0); current_state =  IN_Sleep;
    }else {
      if(incoming_event== event_PKT) {  // Handling PKT event
        getPktData(payload);          // Generic function to get packet content
        process_data();}
```

```
    if(incoming_event== event_CLK)   // Handling CLK event
      temp++;
   }
   break;
 case IN_Transmit_pkt:
   tNextTX =  getRandNumber(); led_toggle(1); current_state=  IN_Sleep;    break;
 case IN_done:
   break;
 default:
   current_state =  IN_NO_ACTIVE_CHILD; break;
 }
}
}
```

Example 1. C code generated by RTW for the state machine of figure 1

6.1 Application porting in MANTIS

MANTIS provides a convenient environment to develop WSN applications. All applications begin with a *start* which is similar to *main* in C programming. One can spawn new threads by calling *mos_thread_new*. MANTIS supports a comprehensive set of APIs for sensor network application development (MOS. 2003), most frequently used APIs are listed below for simple application development based on categories.

- Scheduler : *mos_thread_new*, *mos_thread_sleep*
- Networking : *com_send*, *com_recv*, *com_recv_timed*, *com_ioct*, *com_mode*
- Visual Feedback (Leds) : *mos_led_on*, *mos_led_off*, *mos_led_toggle*
- On board sensors (ADC) : *dev_write*, *dev_read*

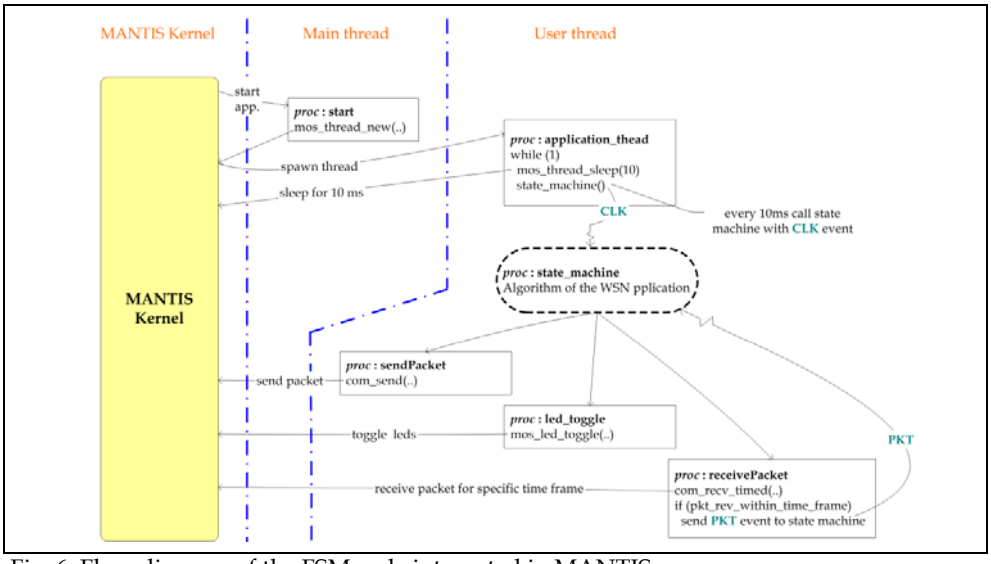


Fig. 6. Flow diagram of the FSM code integrated in MANTIS.

We can port easily the automatically generated code of the state machine in MANTIS. For this, a new thread is spawned from the *start* procedure. In the newly created thread, the state machine is called in every 10 milliseconds, as required in the algorithm. Here the *CLK* is virtually implemented by calling *mos_thread_sleep(10)* . Figure 6 shows the skeleton of the simple application implementation in MANTIS. For receiving packets, the user can use *com_recv* which waits until a successful reception of a packet by blocking the thread. But for implementing our simple application, the program needs to be in the receiving state for certain amount of time. This can be done by another API which is *com_recv_timed*. It turns on the radio in receiving mode for a certain amount of time. When it receives a packet, it calls the state machine with the incoming packet event (*PKT* event of the state machine). Implementation of other outgoing actions such as to sending a packet and switching the leds is also easy, by calling *com_send*, *mos_led_toggle* and *led_on* APIs.

7. TinyOS

The programming model of TinyOS is based on components. In TinyOS, a conceptual entity is represented by two types of components, *Module* and *Configuration*. A component implements interfaces. The interface declares signature of the *commands* and *events* which must be implemented by the provider and user of the interface respectively. Events are the software abstractions of hardware events such as reception of packet, completion of sensor sampling etc. On the other hand, commands are used to trigger an operation such as to start sensor reading or to start the radio for receiving or transmitting etc. TinyOS uses a split-phase mechanism, meaning that when a component calls a command, it returns immediately, and the other component issues a callback event when it completes. This approach is called split-phase because it splits invocation and completion into two separate phases of execution. The scheduler of TinyOS is based on an event-driven paradigm where events have the highest priority, run to completion (i.e. interrupts cannot be nested) and can preempt and schedule *tasks*. Tasks contain the main computation of an application. TinyOS applications are written in nesC which is an extension of the C language.

7.1 Application porting in TinyOS

In TinyOS, application coding uses several interfaces. The skeleton of the simple application implementation is shown in figure 7. Module *simpleAppM* uses interfaces *Boot*, *Timer* and others. When an application module uses an interface then it can issue the commands provided by that interface and it should also implement all the events that could be generated from the interface. For example, the *Boot.booted* event of the *Boot* interface is implemented in the module *simpleAppM*. Among the several interfaces available in the library of TinyOS, we listed those most frequently used for constructing simple applications.

- Initialization: *Init*, *Boot*, *Timer*
- Networking: *Send*, *Receive*, *AMSend*, *SplitControl*, *Packet*, *AMPacket*
- Visual Feedback (Leds): *Leds*

Details of the TinyOS operating system can be found in (TOS. 2000). To implement the simple application, at first a periodic timer (*CLKtimer.startPeriodic*) is initialized from the *Boot.booted* event handler. The period of the timer is set to 10 milliseconds as required in the algorithm. After initialization has been done, a timer event is generated (*CLKtimer.fired*).

Inside this event handler, the state machine is called as a *task* (implementing the *CLK* event of the state machine). The algorithm needs to be in receiving mode for specific amount of time (30 milliseconds). Hence in the *receivePacket* method, we set a one shot timer (for 30 milliseconds) and at the same time start the radio. After expiration of this timer the radio needs to be stopped (done in the event handler of *RXwindowTimer.fired*). When TinyOS receives a packet it generates an event (*Receive.receive*). Inside this event we post the task of the state machine with the incoming packet event (implementing the *PKT* event of the state machine). We used the *LowPowerListening* interface to control the radio explicitly in receiving or transmitting mode. For handling outgoing actions from the state machine, such as to send a packet, the state machine calls the *sendPacket* method. Inside this method, we at first set the radio in transmit mode and then start it. When the radio is started (it generates *Radio.StartDone* event), the method checks whether the radio is turned on for sending a packet or not. If so, we use the *AMSend.send* command of the *AMSend* interface to send the packet.

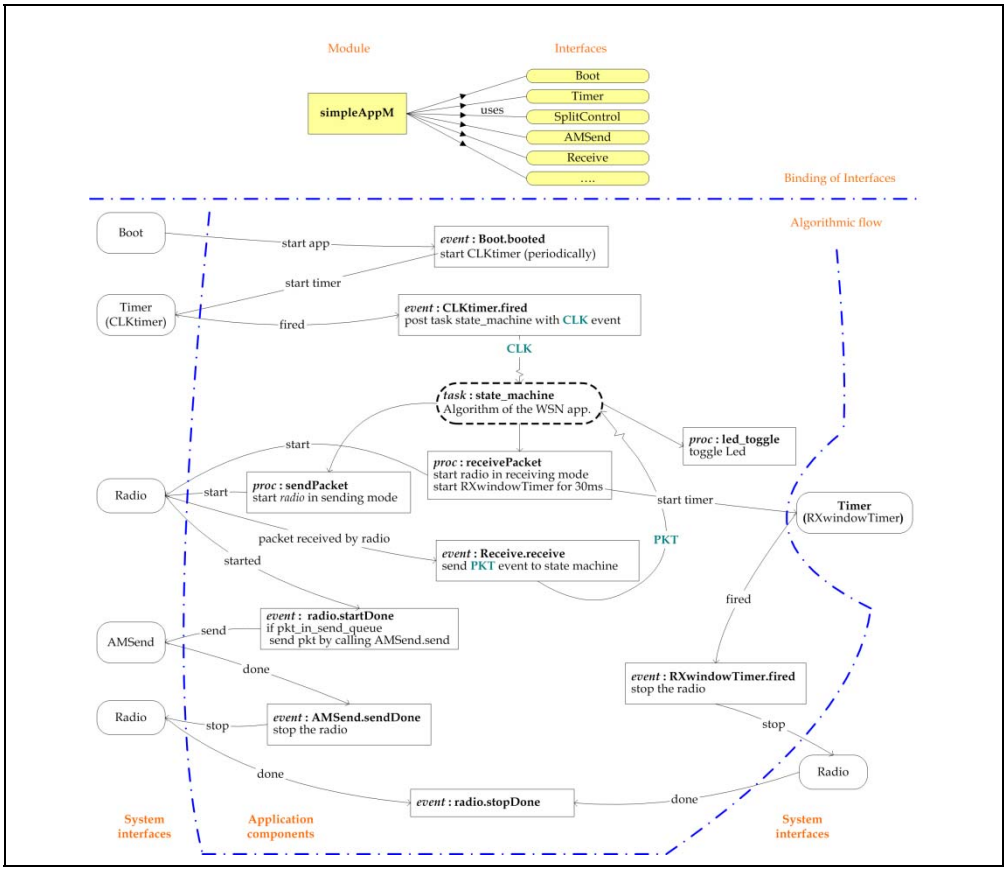


Fig. 7. Flow diagram of the FSM code integrated in TinyOS

When the packet is sent then TinyOS generates a call back event *AMSend.sendDone* which provides the status of the sending processing. Inside this event handler, we stop the radio. There are some commands in TinyOS which are qualified as *async* and do not generate callback events. We used *async* commands for switching the leds from the state machine.

8. ZigBee

ZigBee is a specification that enables reliable, cost effective, low power, wireless networked, monitoring and control products based on an open global standard. ZigBee is targeted at the WSN domain because it supports low data rate, long battery life and secure networking. At the physical and MAC layers, ZigBee adopted the IEEE 802.15.4 standard. It includes mechanisms for forming and joining a network, a CSMA mechanism for devices to listen for a clear channel, as well as retries and acknowledgment of messages for reliable communication between adjacent devices. These underlying mechanisms are used by the ZigBee network layer to provide reliable end to end communications in the network. The 802.15.4 standard is available from (IEEE. 2003).

At the network layer, ZigBee supports different kinds of network topologies such as *Star*, *Tree* and *Mesh*. The ZigBee specification supports networks with one *coordinator*, multiple *routers*, and multiple *end devices* within a single network. A ZigBee coordinator is responsible for forming the network. Router devices provide routing services to network devices, and can also serve as end devices. End devices communicate only with their parent nodes and, unlike router devices, cannot relay messages intended for other nodes. Details of the ZigBee specification can be found at (ZigBee. 2006).

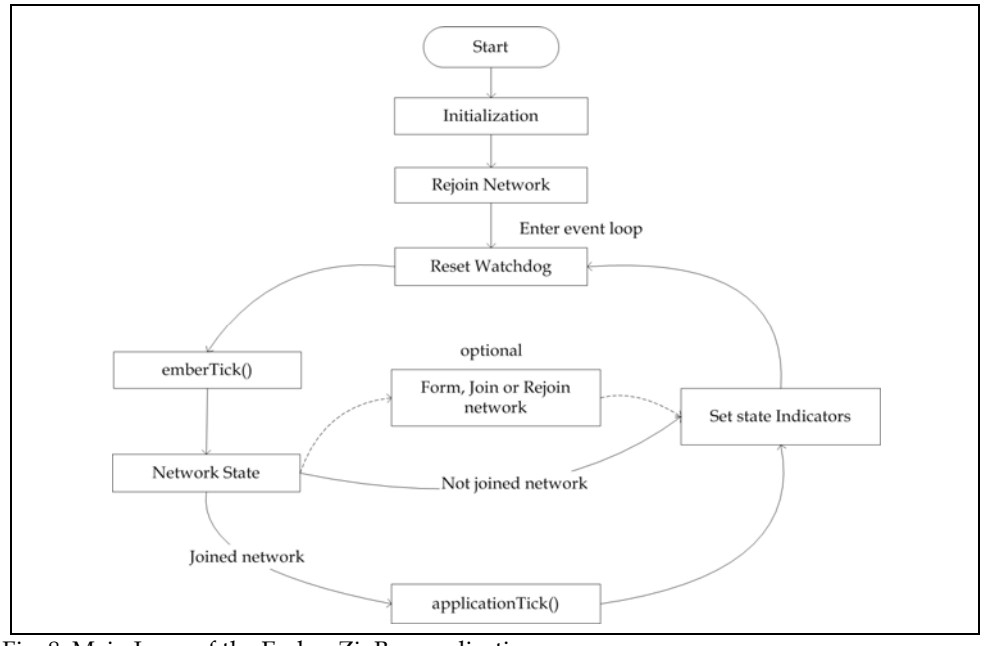


Fig. 8. Main Loop of the Ember ZigBee application

8.1 Application porting in ZigBee

Several implementations of the ZigBee stack are available on the market (such as from Texas Instruments, Ember Corporation, Freescale etc). We will describe our simple application implementation by using the Ember implementation (EMBER. 2008). The main source file of a ZigBee application must begin by defining some parameters involving *endpoints*, *callbacks* and *global variables*. Endpoints are required to send and receive messages, so any device (except a basic network relay device) will need at least one of these. Just like C, an application starts from *main*. The initialization and event loop phases (shown in figure 8) of a ZigBee application are shortly described below.

Among the initialization tasks, serial ports (SPI, UART, debug or virtual) need to be initialized. It is also important to call *emberInit()* which initializes the radio and the ZigBee stack. Prior to calling *emberInit()*, it needs to initialize the Hardware Abstraction Layer (HAL) and also to turn on interrupts. After calling *emberInit()*, the device rejoins the network if previously it had been connected, sets the security key, initializes the application state and also sets any status or state indicators to the initial state.

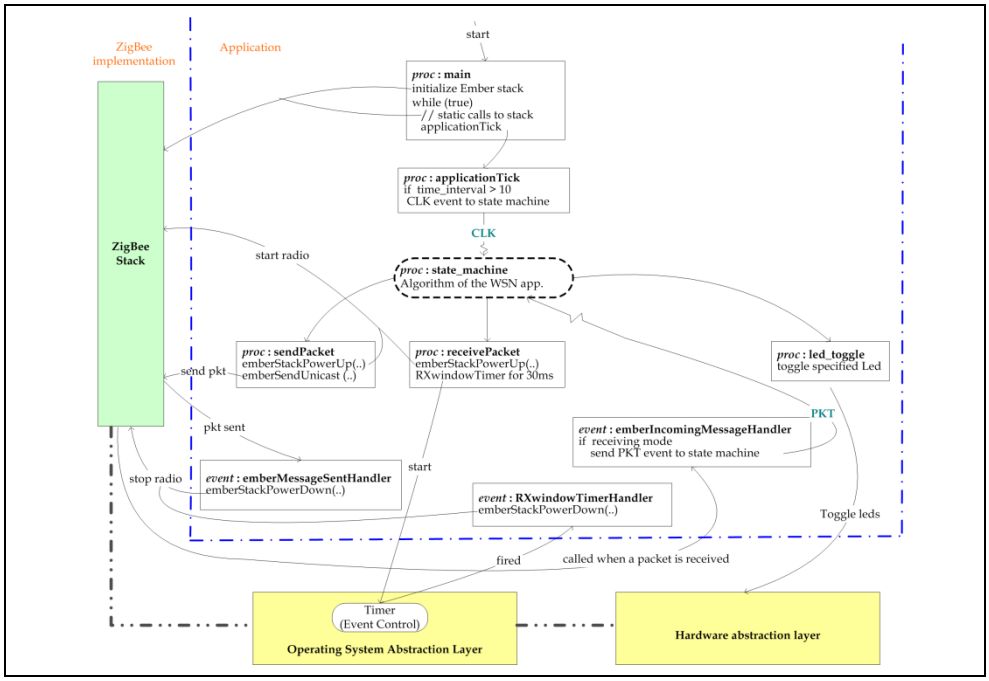


Fig. 9. Flow diagram of the FSM code integrated in ZigBee

The network state is checked once during each cycle of the event loop. If the state indicates joined (in case of router and end device) or formed (for the coordinator) network, then the *applicationTick* function is executed. Inside this function the developer will put the application code. If the network is not joined or formed, then the node will try to join or form the network. State indicators are simply LEDs but could be an alphanumeric display or some other state indicator. The function *emberTick* is a periodic tick routine that should be

called in the application's main event loop after *emberInit*. The watchdog timer should also be reset once per event loop by calling *halResetWatchdog*.
The skeleton of the simple application implementation in ZigBee is shown in figure 9. Here, the state machine is called from *applicationTick*. The state machine is called at 10 millisecond intervals, which implements the *CLK* of the state machine. When the *receivePacket* method is called from the state machine, we start the radio by calling the *emberStackPowerUp* API and then schedule an event (*RXwindowTimer*) which will generate a callback event after expiration of receiving timer (30ms). When this callback event (*RXwindowTimerHandler*) occurs, we stop the radio. In this time frame, if a packet is received by the ZigBee stack, it calls an incoming message handler function *emberIncomingMessageHandler*. Inside this function, the state machine is called with the incoming packet event (*PKT* event of the state machine). When the *sendPacket* method is called from the state machine, again we start the radio and send the packet by calling the *emberSendUnicast* API which afterward calls back the *emberMessageSentHandler* function. Inside this event handler, we stop the radio. Implementations of *led_toggle* and *led_on* methods are simple like in MANTIS and TinyOS.

9. Conclusion

We described an extensible framework for modeling, simulation and multi-platform code generation of sensor network algorithms based on MathWorks tools. We developed parameterized blocks for the *sensor node* and *communication medium* to ease the modeling and simulation of WSN applications. Portability of application between multiple platforms is an open problem, especially in the WSN domain because of the lack of a single platform standard. We presented application porting in MANTIS, TinyOS and ZigBee using a simple application. We identified a single code writing style, namely state machine-like, that can be ported easily across different platforms by just creating an API abstraction layer for sensors, actuators and non-blocking OS calls. This FSM-like code can be written by or generated from different StateChart-like or Synchronous Language models, which also makes the generation of the adaptation layer to each platform easier. The reason for choosing the MathWorks tools over, for example, TOSSIM, NS, OmNet, is that they are well known and already provide rich libraries for digital signal processing and control algorithm behavior simulation.

10. References

Abdelzaher, T. ; Blum, B. ; Cao, Q. ; Evans, D.; George, J.; George, S. ; He, T. ; Luo, L. ; Son, S. ; Stoleru, R.; Stankovic, J. & Wood, A. (2004). Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. *In the Proceedings of the IEEE International Conference on Distributed Computing Systems*, Tokyo, Japan

Almeida, V.; Vieira, L.; Vitorino, B.; Vieira, M. ; Fernandes, A.; Silva, D. & Coelho, C. (2003) Microkernel for Nodes of Wireless Sensor Networks, *In the poster session of the 3rd Student Forum SBCCI*, Chip in Sampa, Brasil.

Barry, R. 2003. FreeRTOS, A FREE open source RTOS for small embedded real time systems. http://www.freertos.org/PC/.

Bakshi, A.; Prasanna, V. K.; Reich, J. & Larner, D. (2005). The abstract task graph: a methodology for architecture-independent programming of networked sensor systems. *In the Proceedings of End-to-end, sense-and-respond systems, applications and services*, pages 19–24.

Bhatti, S.; Carlson, J.; Dai, H.; Deng, J. ; Rose, J. ; Sheth, A. ; Shucker, B. ; Gruenwald, C. ; Torgerson. A. & Han., R. (2005). MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *In the journal of MONET*, pages 563-579

Cheong, E.; Lee, E. & Zhao, Y. (2005). Viptos: A graphical development and simulation environment for TinyOS-based wireless sensor networks. *In the Proceedings of 3rd International Conference on Embedded Networked Sensor Systems*, *SenSys*, page 302

Dunkels, A.; Gronvall, B. & Voigt, T.(2004). Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors, Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, ISBN 0-7695-2260-2, pages 455--462, USA

Eker, J.; Janneck, J.; Lee, E. A. ; Liu, J. ; Liu, X. ; Ludvig, J. ; Sachs, S. & Xiong Y. (2003) Taming heterogeneity - the Ptolemy approach, *Proceedings of the IEEE*, volume 99(1), pages: 127-144

EMBER. (2001). Zigbee Wireless Semiconductor Solutions by Ember. www.ember.com.

Gay, D.; Levis, P.; Behren, J. R.; Welsh, M.; Brewer, E. A. & Culler, D. E. (2003) The nesC language: A holistic approach to networked embedded systems. *In the Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI, pages 1-11,

Gummadi, R. ; Gnawali, O. & Govindan, R. (2005). Macro-programming wireless sensor networks using kairos. *In the Proceedings of the 1st International Conference on Distributed Computing on Sensor Systems*, pages 126-140

Halbwachs, N. (1993). *Synchronous Programming of Reactive Systems*, Kluwer Academic Publishers

Levis, P.; Lee, N.; Welsh, M. & Culler, D.E. (2003) TOSSIM: accurate and scalable simulation of entire tinyOS applications. *In the Proceedings of 1st International Conference on Embedded Networked Sensor Systems*, SenSys, pages 126-137

Levis, P.; Madden, S.; Gay, D.; Polastre, J.; Szewczyk, R.; Woo, A.; Brewer, E. A. & Culler, D. E. (2004) The Emergence of Networking Abstractions and Techniques in TinyOS. *In the Proceedings of 1st Symposium on Networked Systems Design and Implementation*, NSDI, pages 1-14, 2004

Necchi, L. ; Bonivento, A. ; Lavagno, L. ; Vanzago, L. & Sangiovanni-Vincentelli, A. (2007) EERINA: an Energy Efficient and Reliable In-Network Aggregation for Clustered Wireless Sensor Networks. *In the Proceedings of Wireless Communications and Networking Conference*, *WCNC*, pages 3364-3369

Newton, R. & Welsh, M. (2004). Region streams: functional macroprogramming for sensor networks. *In the Proceedings of the 1st International Workshop on Data Management for Sensor Networks*, pages 78-87

MathWorks. (1984). MATLAB and Simulink for Technical Computing. www.mathworks.com/

MOS. (2003). MANTIS, MultimodAl NeTwork of In-situ Sensors. http://mantis.cs.colorado.edu/index.php/tiki-index.php

Mozumdar, M.M.R. ; Gregoretti, F. ; Lavagno, L.; Vanzago, L. & Olivieri, S. (2008a). A framework for modeling, simulation and automatic code generation of sensor network application, In the Proceedings of 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pages 515--522.

Mozumdar, M.M.R. ; Gregoretti, F. ; Lavagno, L. & Vanzago, L. (2008b). Porting application between wireless sensor network software platforms: TinyOS, MANTIS and ZigBee, In the Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation, pages 1145-1148.

NS-2. 2001. The Network Simulator. 2001. http://www.isi.edu/nsnam/ns

OMNeT. (1992). Community Site. http://www.omnetpp.org/

Vieira, L. F. M. ; Vitorino, B. A. D.; Vieira, M. A. M.; Silva, D. C. & Loureiro, A. O. (2005). WISDOM: A Visual Development Framework for Multi-platform Wireless Sensor Networks. *In the Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation*, *ETFA*, Catania, Italy.

RTW. (2009). Real-Time Workshop - Generate C code from Simulink models and MATLAB code. http://www.mathworks.com/products/rtw/

SF. (2009). Stateflow-Design and simulate state machines and control logic. http://www.mathworks.com/products/stateflow/

TOS. (2000). TinyOS Community Forum, An open-source OS for the networked sensor regime. http://www.tinyos.net/

ZigBee. (2006). ZigBee Alliance. http://www.zigbee.org/.

6

VAN Applied to Control of Utilities Networks. Requirements and Capabilities.

Javier Silvestre-Blanes, Víctor-M. Sempere-Payá
and Teresa Albero-Albero
Instituto Tecnológico de Informática (ITI)
Universidad Politécnica de Valencia (UPV)
Spain

1. Introduction

Industrial communication networks have played a key part in the success and evolution in the concept of CIM (*Computer-Integrated Manufacturing*) and in their more recent evolution MES (*Manufacturing Execution System*) and ERP (*Enterprise* Resource Planning). These systems provide a means of communication for industrial applications of a distributed nature, through the use of sensors, controllers and intelligent devices capable of exchanging and processing information. Although there were precursors to current industrial networks in the area of instrumentation such as CAMAC (*Computer Automated Measurement and Control*) or GPIB (*General Purpose Instrumentation Bus*), these types of networks really began to appear in the 1980s, with MAP (*Manufacturing Automation Protocol*) and its variants can generally be considered as the first industrial networks. MAP appeared in the manufacturing plants of General Motors as a consequence of the growing cost and complexity of interconnection between machines. This difficulty arose principally because of the incompatibility between the many different proprietary systems in use at the time, and led to the development of standards designed to eliminate this incompatibility problem (See (Sauter, 2005a) for a more detailed description of the history of fieldbuses and the evolution of their standardization process).
The introduction of these networks led to a change of paradigm in the development of automated industrial systems, allowing the development of systems that were both distributed and decentralized. The use of these networks has also had a great influence on what is called the "automation pyramid". Although various proposals and numbers of levels are given today (Sauter, 2005b), initially within the CIM hierarchy at the network level, there were considered to be three: factory networks, cell networks, and fieldbuses. Of these, only the last was expected to have special capabilities, different from what was traditionally expected of office networks; that is, the capability to perform in harsh environments, and that users of this exchange of information were processes and not humans (Decotignie & Pleineveaux, 1993). However, from a network point of view, the most significant differences are in:

- The time requirements, which become stricter as we go lower on the pyramid.
- The volume of information exchanged, which becomes less as we go lower on the pyramid.
- The frequency of information exchange, which becomes greater as we get closer to the process.

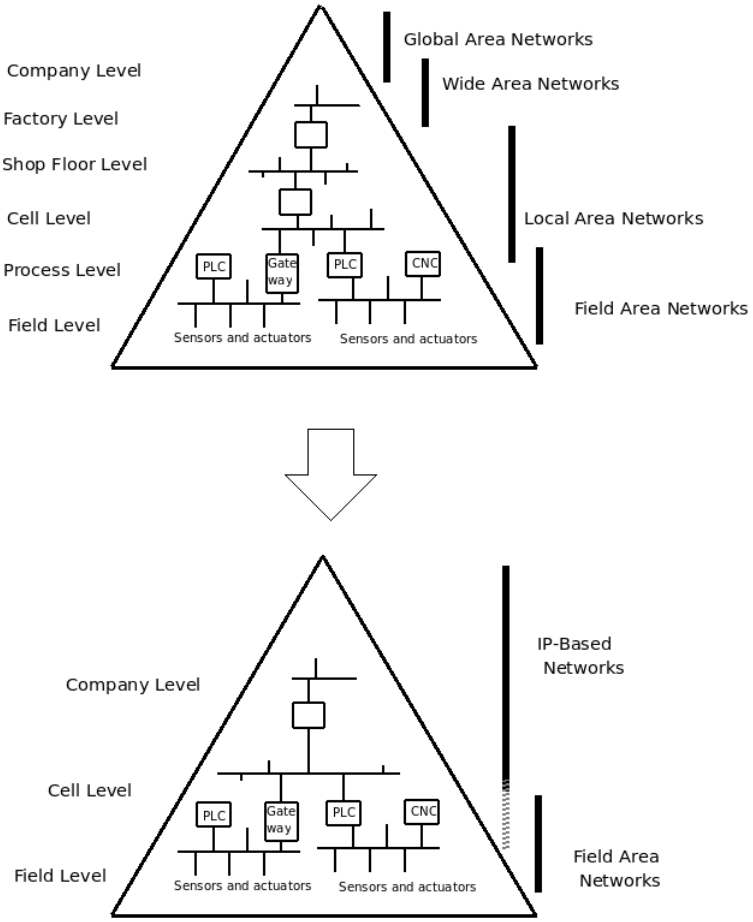


Fig. 1. Evolution of the CIM pyramid

However, this distinction corresponds with the initial concept of the CIM pyramid. Currently, the introduction of Ethernet in the different levels of the pyramid (even the lower level) and the use of internet have reduced the number of layers to only three (Sauter, 2007), as can be seen in Fig 1. This has affected the design of new networks and applications, which now offer new types of services. From the point of view of the application domain, although industrial networks were initially only applied in the lower levels of the pyramid,

fieldbus technology influenced all applications domains. Currently, these networks are classified into six different categories (Thomesse, 2005), whose principal characteristics are shown in table 1.

		requirements		
Application Domain	**Distances**	**Time**	**Safety**	**Dependability**
discrete manufacturing	LAN	High/medium	Not specified	
process control	LAN	Very high	Not specified	
Embedded	PAN	Higher/lower	Major constraint-none	
Transportation	MAN-WAN		Very high	
Building Automation	LAN,MAN	low	high	
Control of Utilities Networks	MAN, WAN	medium	Medium	

PAN: Personal Area Networks **LAN**: Local Area Networks
MAN: Metropolitan Area Networks **WAN**: Wide Area Networks

Table 1. Fieldbus Applications Domains

In each of them, there are different requirements and constraints in terms of time, synchronization, distances, safety and dependability. However, although there may be principles in common between application domains, each individual application can have its own particular requirements. For example, in embedded systems domains the requirements are completely different for the control of a motor or a vehicle's brakes from those of a coffee machine; or in Transportation systems, where there are also important requirement differences between the management of a railway network, and highway monitoring applications.

The first three categories, due to similarity in distances covered, are considered to belong to the LAN/PAN domain. The first application domain (*discrete manufacturing*) is characterized by the existence of a stable state of products between two operations, which allows a process to be divided into various subprocesses independent from each other. Dependability is connected with productivity. The temporal requirements for a machine or process are quite high, but the traffic between different processes can be considered to be asynchronous. The second application domain (*process control*) involves continuous processes, so there are not stable states between two processes. The temporal requirements are greater than the requirements in the same machine of the previous domain, and also cover a wider area, since these requirements must also be satisfied between different machines. Safety is usually more stringently controlled (for example in the chemical industry), and dependability usually demand redundancy. In the third (*embedded systems*) the distances are very short, and so must be considered as PANs. When used in vehicles of some type, they have the greater requirements, but for many other types of embedded system these requirements are very relaxed.

The other application domains, belong to the MAN/WAN domain. The *Transportation systems* domain covers (still following the Thomesse classification) the management of railway networks, remote control of urban traffic, the monitoring of railways, etc., and therefore safety, dependability and availability are crucial. In the *building automation* area the applications are more relevant to data acquisition and supervision than to control, this being often very simple. There is more extended use here, with the typical state information, of multimedia streams. Another difference is the use of a higher number of devices and controllers, which makes them very complex systems. Reliability is also required, but with lower demand. In the ***control of utilities networks*** the applications consist of remote

monitoring and control of very large networks for the distribution of water, gas, or electricity. The networks are no longer really LANs, as the operators are in ***central control rooms*** (CCR) for operation and maintenance organization. The traffic consists of status variables and events, the data rate depending on the complexity of the system considered. The networks have the same dependability roles as a fieldbus in a factory, the difference being in the distances covered. For example, power line protocols are used in electrical networks, radio waves are often used to connect very remote stations, and it is also now a preferred domain for Internet use.

The necessities when working with geographically distributed plants require the use of ***heterogeneous networks***, consisting of local and wide area, and wired and wireless communication systems operated by different authorities. (Sempere et al., 2006). In an automation environment, these heterogeneous networks are denominated ***Virtual Automation Network*** (VAN) (Neuman, 2003a)(Neumann, 2007) since the classical requirements demanded of fieldbuses must be satisfied by a network that is currently composed of many different networks. A VAN is "a heterogeneous network consisting of wired and wireless local area network, the internet, and wired or/and wireless telecommunications systems" (Neuman, 2003b). However, it is usually only applied to discrete manufacturing applications (Balzer et al., 2008), and not to other fieldbus application domains which are type MAN/WAN.

In Fig. 2 we can see a typical VAN network for control of utilities in a big city. There are a wide range of remote nodes: remote stations for environment, for control (pumps, gates, levels), for control/monitoring by multimedia, mobile stations, etc. Each remote station can be considered as an autonomous entity that controls a particular aspect of the application. This entity uses fieldbuses with more or less strict real time requirements depending on the type of station, in order to perform the service desired in each area (automation islands). However, to improve the working and use of the system, facilitate maintenance, improve responses in situations of risk and alerts, increase the quantity of information available in the control center etc., VAN must provide the capacity to communicate information between nodes and the central using a wide range of solutions.

In this chapter, the requirements that this type of applications make on VAN networks, the different type of technologies that are normally employed and the capacities that they can offer are analyzed. Public networks and Global Wireless networks are analysed in (Sempere et al., 2003) and in (Sempere et al., 2004; Albero et al., 2005) respectively, so in this chapter an special focus is done in WiMAX networks, since they have enormous potential in this area, an area which up to now has been studied very little. Also, the real implementation of a VAN network for the purification network of Valencia City is presented and a video can be seen in extension 1, which shows the real capabilities of this kind of network.

This chapter is organized as follows. In section II the services supported by the VAN infrastructure for control of utilities networks is analyzed. In section III we review the networks employed in factory automation, putting special emphasis on MAN networks and their use in a VAN environment. In section 4, we present results obtained using WiMAX 802.16-2004 networks as a VAN infrastructure in a utilities network control application.

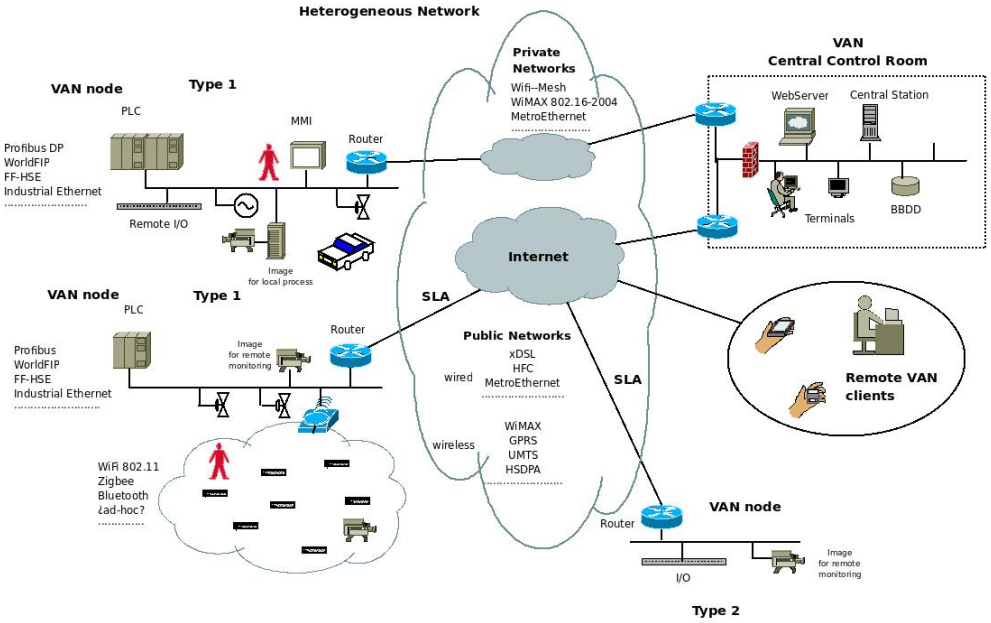


Fig. 2. Architecture of a Classical Control of Utility Network

2. Services

As previously mentioned, in each of the fieldbus application domains, there are a wide variety of requirements generally, and control of utilities network and in particular the VAN concept are no exception. One important difference is in the function of the services destinations; basically users and processes.

The services offered to a user of a VAN are divided into media services and alert services. In the opposite direction, it is the sending of orders by the client in order to gain information or to act on the process. In the case of alert services the volume of information is very low. Normally we are dealing with a few octets which give information on a particular state or alarm in the installation. The user receives the alarm because he is subscribed to this type of alarm service. In the case of alert services, the source of the alert will always be control equipment in a VAN node. The receiver may be in the central control room, in a VAN node or in a remote VAN client. The maximum delay between production and reception of the alarm is 1s. The media services typically offered will be images or video streaming for the supervision of the installation. The source is a camera installed in a VAN node (see Fig.1 image for remote monitoring). Images or streaming are offered by camera and sent to a client located in the VAN node itself, in another VAN node, in the VAN central control room or may even be sent to a remote VAN client. The maximum delay for receiving supervision images or video streaming is 3s.

A user can send requests or orders to obtain information or to act on a process. The user requests to receive images or video streaming, but not alerts, as this is a service that the user is subscribed to and does not need to request. If the user is located in the VAN central

control room, he will make the request through an image order from the CCR. On the other hand, if the user is a remote VAN client, the information will be offered via a web page. In this last case, there must be access security guarantees. Another type of order is those that allow a user to parameterize the installation. These orders do not allow a user to change the state, and furthermore, remote orders are not permitted for security reasons. Safety of personnel must be guaranteed (functional safety), an operator who is not in the VAN itself will not be able to execute an order because it is not known whether it could affect the safety of the personnel working in this VAN node.
The services offered in a VAN network are divided into alert and control services, and media services. For media services, in the case that the source is a camera installed in a VAN node offering images of local processes (see Fig.1), these images are not offered to a user but to a process. The requirements here are different those of supervision. The maximum delay permitted for supervision images is 3s. Concerning alert and control services, this type of traffic has traditionally been divided into four groups:

- Best effort service: allows basic connectivity without QoS guarantees. There is no difference between traffic flows.
- Soft Real Time (RT) service: also called differentiated service or soft QoS. To offer this service, some traffic has preferential treatment over other traffic, being offered greater bandwidth, a lower loss rate, greater speed, etc. This achieved by classifying the traffic along with the use of QoS tools. The reaction time permitted is between 10 ms and 100 ms.
- Hard Real Time service: denominated guaranteed service or hard QoS. To offer this type of service, all the resources of the network are reserved for particular traffic. The reaction time permitted is 10 ms.
- Isochronous: these services are used when there are strict bandwidth requirements as occurs with particular audio and video services. That is to say that there are applications that require the continuous sending of information at defined intervals. The reaction time permitted is 1 ms.

These temporal requirements are satisfied by different types of fieldbuses in PAN and LAN type applications that can be found in VAN nodes, but in VAN applications the requirements must be at least one order of magnitude greater or even completely discarded, as with isochronous services. In applications supported by VAN infrastructures, and in particular in the domain of utilities network control, each VAN node will use local control systems in order to be able to guarantee isochronous and/or hard real time behavior, but can use the VAN infrastructure to communicate events, receive orders (such as changes in control policies, etc.) and synchronize actions with other VAN nodes.

3. Networks

The communication between control devices inside each VAN node is outside the scope of this chapter. However, there are solutions which provide isochronous Real-Time between the distributed processes inside. An interesting review can be found in (Thomesse, 2005), and based on Ethernet solutions in (Decotignie, 2009). In this section, there is a brief description of the most interesting technology making up heterogeneous networks which form part of the VAN infrastructures in a metropolitan installation such as that found in Control of Utilities Networks.

3.1 Public Networks

Public networks are defined as networks which are publicly owned, however, the infrastructure is usually operated, in part or completely, by private companies. (service providers). Public administration owns the network of infrastructures, and as such must guarantee access to the communication network at high speed to the whole population. Nowadays, the most common internet access technologies are xDSL (Digital Subscriber Line), both asymmetrical (ADSL) and symmetrical (SDSL). The formal agreement between Internet provider and the client is commonly denominated the Service Level Agreement (SLA), specifying the level of service, mainly the bandwidth and availability guarantee. However, IPv4 networks are only capable of offering best effort services, for which reason their use in factory automation is fairly limited. Networks of this type have been used for monitoring systems (Sempere et al., 2003) and there are various proposals for using them with real time services (Torrisi et al. 2007) (Balzer et al. 2008).

3.2 Wired Networks

The management and operation of distributed installations metropolitan environment are based on a collection of heterogeneous networks, mobile networks, fixed wire, coaxial and fiber-optic networks, etc. which operate in in a MAN environment. There is evidence of the growing use of METRO ETHERNET and CARRIER ETHERNET as technologies in telecommunications networks on the part of providers and operators. The reason for the growing use of this technology is clearly that when generating and receiving information in extreme formats, using Ethernet means the transport has the same format, the benefits of which are evident: efficiency and simplicity.

Metro Ethernet Forum (MEF. See http://metroethernetforum.org) defined the attributes that Metro and Carrier Ethernet must have independently of the solution system used: SDH (Synchronous Digital Hierarchy), OTN (Open Transport Network), HFC (Hybrid Fiber Coaxial), OF (Optical Fiber), WDM (Wavelength Division Multiplexing), WiMAX, Bridging, MPLS (Multiprotocol Label Switching), etc. These attributes defined by the MEF are: standardized services, scalability, security and robustness, quality of service and management (MEF, 2006). Two connectivity services were initially defined, E-Line and E-LAN (MEF, 2004) which were later broadened with E-Tree (MEF, 2008). All Ethernet services can be defined within these categories. E-Line is that which provide Ethernet Virtual Connection (EVC) point to point between two UNIs (Unit Network Interface). An E-Line service can provide a symmetric bandwidth to send information in whatever direction and without any type of quality of service (best effort) at 10 Mbps between two UNIs. An E-LAN provides multipoint to multipoint connectivity connecting two or more UNIs. E-Tree services provide Ethernet Virtual Connection (EVC) point-to-multipoint. In all cases, best effort services are provided, but if it were necessary to offer more sophisticated services, communications could be carried out with certain guarantees, offering different speeds depending on the direction of transmission. To achieve this, it would be possible to combine two of four Traffic parameters defined in a Bandwidth Profile service attribute (Kasim, 2008).

3.3 Wireless Networks

Wireless networks have received a great deal of attention in recent years, and their use today is fairly widespread in PAN y LAN environments. In the area of factory automation, the mobility and flexibility of these types of networks offer interesting advantages and applications, and a great deal of effort has recently gone into solving the various problems inherent in an open, unstable medium such as this (Willing et al. 2005; Matkurvanov et al., 2006; Cena et al. 2008). It is now common to see these networks incorporated in different standards, although several proprietary solutions have also been developed, such as the ABB wireless sensor, denominated WISE (Scheible et al. 2007).

Within wireless networks, there are also PAN, LAN, MAN and WAN classifications. The principal characteristics and difficulties in this type of network when compared to wired networks are:

- Problems related to security are accentuated in wireless networks. This is a shared communication medium which can be accessed by anyone. The privacy of information must be guaranteed, and for this the most common solution is information encryption (Crow et al. 1997).
- Interference and reliability. Interference in wireless communications is common due to being a shared medium. A typical problem is that of hidden terminals, which occurs when there are two nodes within communication range of a third node but not of each other. The typical solution for this is coordination of terminals for the transmission of information (RTS/CTS, Request to Send/Clear to Send).
- Frequency allocation. So that the different nodes in a network can communicate with each other, they must operate on the same frequency.
- Mobility. One of the principal advantages of wireless networks is mobility. However, this often means a network topology that is changing, and where links between nodes are created and lost dynamically.
- Throughput. Due to physical limitations and the fact that the bandwidth available in a wireless interface is less than with a wired interface, transmission rates are lower than in wired networks. To support multiple transmissions simultaneously, spread spectrum techniques are frequently employed.

Another important characteristic leading to significant differences in this type of network is whether there is the need for a Line of Sight (LOS) between the entities that establish a wireless communication, or not (NLOS).

3.3.1 wPAN

Wireless Personal Area Networks (WPAN) interconnect devices in a small area. 802.15 is the wireless standard defined by IEEE for WPANs. The first version, 802.15.1, is based on the Bluetooth specifications (lower layers) and is completely compatible with Bluetooth 1.1. The standard 802.15.1 was approved in 2002 (802.15.1-2002) by IEEE and in 2005 an updated version was introduced, 802.15.1-2005. This standard allows interconnection of devices at distances of a few cm (≈ 10 cm) up to 10 meters. The IEEE Working group continued improving the Bluetooth standard. Two categories of 802.15 are currently proposed: the low rate 802.15.4 (TG4) and high rate 802.15.3 (TG3). The IEEE 802.15.4 standard (IEEE, 2003), approved in 2004 and promoted by the ZigBee Alliance, has been developed to enable applications with relaxed bandwidth and delay requirements, where the emphasis is device battery lifetime maximization. Devices can be powered by batteries for months or even

years. These applications will be run on platforms such as sensors. The IEEE 802.15.3 defines the PHY y MAC levels for high speed WPANs (15 – 55 Mbps). With the IEEE 802.15.3a standard, there was an attempt to improve the physical layer of UWB for its use in applications which work with multimedia applications; however, after several years of deadlock, the IEEE 802.15.3a task group was dissolved in 2006.
In table 2 the main properties of these networks are summarized. Although they are used as sensor networks in industrial environments, the coverage area does not allow their use as VAN networks.

Network	Band	Channels	power	Distance	Throughput	No. elements
Bluetooh – class 1	2.4MHz ISM	79	100mW	≈100m	1Mbps	7
Bluetooh – class 2	2.4MHz ISM	79	2.5mW	≈10		7
Bluetooh – class 3	2.4MHz ISM	79	1mW	≈1		7
Ultra-Wideband	3.5GHz		20-40 mW	10 m	50-480Mbps	
802.15.4	868-868.6MHz	1	200-500 μW	10-100 m	20,100,250 Kbps	65000
	902-928MHz	10		10-100 m	40-250 Kbps	65000
	2.4MHz ISM	16		10-100 m	250Kbps	65000

Table 2. Wireless PAN

The Task Group 4 (TG4), presented a wide range of applications at the 802 meeting in 2001 (Gutiérrez et al. 2001). These applications include industrial monitoring and control. It is important to highlight that industrial monitoring applications which use this protocol must not be operating with critical information, must accept high latency and do not need constant updating. However, there are now several proposals in existence which aim to support real time applications. These are based on the Guaranteed Time Slot (GTS) mechanism (JunKeun et al. 2007) and the use of low superframe orders (Koubba et al. 2006)

3.3.2 wLAN

In Wireless Local Area Networks (WLAN) there are technologies based on HiperLAN (High Performance Radio LAN) a group of the standard group ETSI (European Telecommunications Standards Institute) and Wi-Fi standardized under IEEE 802.11 series. This standard works on unlicensed ISM band, using 2.4 GHz in 802.11/b/g, and 5 GHz for 802.11/a/n. There are 11 channels (although this may vary from country to country) with a bandwidth of 22 MHz per channel for the standards IEEE 802.11 b/g and approximately 20 MHz for the standard 802.11a, and a separation value between channels higher than (5MHz). Because of this, non consecutive channels are usually used. As well as the traditional problems with wireless networks, we must also consider the possibility of multiple interference from other devices and equipment. With a transmission power of 100mW, this technology provides NLOS communication with coverage's of between 100 and 400 meters, and throughputs of between 11 (802.11b) and 300 Mbps (802.11n)[1]. The use

[1] According to the march 2009 DRAFT the standard 802.11n must work at 2.4 GHz and at 5GHz, although as the band 5GHz used by 802.11a is less used, there is a general feeling that 802.11n should not be permitted to include the option of working in the 2.4 GHz band. Regarding the Mbps, according to the DRAFT, it must be capable of working at 600 Mbps.

of this type of network in automation environments has been studied a great deal, due to the characteristics mentioned, although for the coverage range, only in LAN environments. For example, in (Rauchhaupt, 2002) and in (Willing, 2003) the use of WiFi in Profibus is analysed. In (Willing, 2005)(Willing, 2008)(De Pellegrini et al., 2006) the authors carry out a comprehensive study on the use of wireless networks in industrial applications. In (Brevi et al. 2006) the authors evaluated the use of 802.11a in a real industrial environment.

3.3.2 wMAN

When speaking of wireless networks with a MAN type coverage, we are basically speaking of trunk networks and of WiMAX. The first of these are not standardized and are based on proprietary technology. WiMAX networks, with coverage's of up to 8-50 Km (depending on whether they are LOS or NLOS), are a clear alternative for the development of networks for son tele-operation and tele-supervision (Cardeira, 2006; Silvestre et Al. 2007). WiMAX activities started in August 1998, and it was the IEEE 802 group who led the formation of the Working Group 802.16 Broadband Wireless Access (BWA) Standards in 1999. The necessity of NLOS (Non Line of Sight) links moves the frequency bands from 10-66 GHz to 2-11 GHz. The first standard, 802.16a in 2001, was improved upon by the standard 802.16d, which is normally known as 802.16-2004. The standardization of 802.16e in 2005 adds mobility support to the family (Li, 2007). Other standards that will be a choice in the future for this kind of application will be IEEE 802.22 Wireless Regional Area Network or IEEE 802.20. The current standardization process and their different characteristics are summarized in table 3. IEEE 802.16 is optimized for point to multipoint (PTMP) configurations, where there is a base station (BS) and several subscriber stations (SS). Later amendments also allow for mesh network architecture. Of the three air interfaces, the OFDM (Orthogonal-frequency division multiplexing) is suitable for NLOS and is the more extended due to a lower peak to average ratio, faster FFT (Fast Fourier Transform) calculation, and less stringent requirements for frequency synchronization compared to OFDMA (OFDM access) (Ghosh, 2005). It offers a flexible burst-type frame structure with fixed frame duration, and the duplexing is provided by means of TDD (Time Division Multiplexing).

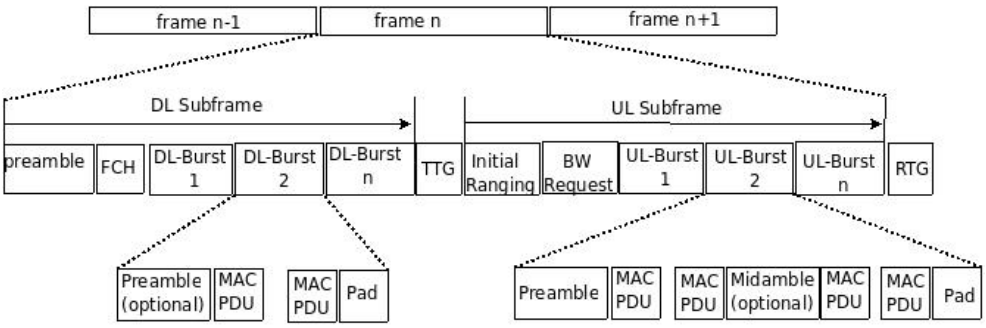


Fig. 3. IEEE 802.16 MAC frame in TDD mode (Hoymann, 2005)

In Fig. 3 the structure of an IEEE 802.16 frame is shown with Time Division Multiplexing (TDD). As can be seen, the BS has the capability for a full schedule of the traffic BS to SS, in the DL subframe. Also, in the UL subframe, there is a mechanism for bandwidth request.

However, the BS also has the possibility to schedule each burst, so it can map the QoS demands in the frame structure in a centralized way.

Network	**Band**	**Power**	**Distance**	**Throughput**	Network Architecture
802.16a	10-66GHz		50Km	32-134 Mb/s	PTP, LOS
802.16d 802.16-2004	2.5- 2.69 GHz 3.4-3.6GHz	3W (SS) 60W (BS)	50Km LOS 7.4Km NLOS	Up to 75 Mbps	PTP, PTMP, NLOS
	5.725-5.850GHz	1 W			
802.16e 802.16-2005	2-6GHz		2km	63Mb/s downlink 28Mb/uplink	PTP, PTMP, mobile

Table 3. Wireless MAN

QoS provisioning in 802.16 is based on Bandwidth grant services. For the downlink flow the BS has information for a correct scheduling. For uplink flow, the BS has to schedule the traffic based on the information provided by SS. There are different types of services (Cicconetti et al, 2006):

- UGS (Unsolicited Grant Service). This provides fixed size transmission at regular time intervals without the need for request or polls. It is adequate for constant bit rate traffic such as VoIP. (Strict delay requirements. Offset upper bounded by the tolerated jitter)
- rtPS (Real-Time polling service). This provides transmission at regular time intervals, where the BS offers the SS periodic request opportunities to indicate the required bandwidth. It is adequate for variable bit rate (VBR) traffic such as MPEG video (less strict delay requirements. Parameters: *minimum reserved traffic rate, maximum latency*
- nrtPS (Non-Real-Time polling service). This type is used for delay-tolerant data service with a minimum data rate. SS can use contention requests and unicast request opportunities will be offered to SS regularly.
- BE (Best Effort). Similar to nrtPS, does not provide bandwidth reservation or regular unicast polls.
- ErtPS (Extended Real Time polling service). This is only applicable to 802.16e (Li, 2007). Provides a service similar to UGS and rtPS. Offers unsolicited unicast grants, but with a dynamic bandwidth allocation.

4. Experiments and results

In this section, there is an analysis of the working of WiMAX networks as support for VAN communications in control of utilities networks applications. Fig. 4 shows the distribution of the stations that use WiMAX as a communications network to give support to the services that the VAN network has to offer to a Control of Utilities Networks application. The equipment works in TDD mode, with OFDM modulation, in the 5.4 GHz band (5.470-5.725 GHz), with adaptive modulation and channels of 10 MHz. As can be seen in Fig. 4, the testbed scenario presents a wide range of WiMAX scenarios, with point to point and multipoint communication, with LOS and NLOS links of between 0.5Km a 2.5Km. The fig. also shows the SNR of the links, which affects the type of modulation to be used and the

maximum bit rate that can be achieved. To analyze the channel, interarrival packet time was measured (Katabi & Blake, 2002) (Varga & Kún , 2005) with different periods, message sizes and sending by unicast and multicast, obtaining also the Packet Error Rate (PER) obtained from the point of view of the application.

The first of the tests consisted of a ***multicast*** transmission from the Central Control Room (**CCR**) to the rest of messages with periods (T) of 50, 100 and 1000 ms. and payloads (C) of 100, 1000, 4000 and 10000 bytes. These values were chosen to evaluate the capacity to offer the services defined in section 2, and represent bandwidth requirements ranging from 800bps to 1.6 Mbps, which could saturate some links. The second of the tests represented a ***unicast*** transmission from the VAN nodes to the CCR with periods of 50 ms. and payloads of 100, 1000, 4000 and 10000 bytes.

Table 4 shows the PER obtained in each of the multicast and unicast experiments. The table shows how in the majority of cases, when the bandwidth requirement is less than 640kbps, a PER less than 1% was obtained, which is acceptable for the application. However, for C=10000B, the PER obtained is excessive even for T of 1000 ms. Of these values, the greatest degradation is seen in the link between CCR and VAN node 2, reaching a PER of 55.6%. This is due to the available bandwidth that the PTMP has to share between VAN 2, 4 y 6, and the worst SNR of this link which means that it has to work with modulations that reduce the channel bandwidth. Another reason is the double traffic that VAN 2 has to carry, as it makes the link between the CCR and VAN 3. This being the most affected node, in the best cases it has double the PER rates compared to the bridging node. Of the rest of the values obtained, it is worth highlighting the good performance of the PTP link between the CCR and VAN6, as well as that of the other nodes communicating with the PTMP link, that is to say VAN 4 y VAN 5.

Concerning the communication from the VAN nodes to the CCR, it is evident that this means a higher overload of the channel, which increases congestion in the links and leads to a higher PER, although this remains acceptable for rates lower than 640Kbps. Although the PER is widely used as a measure for the characterization of channels in a wireless environment, circumstances sometimes arise that affect the channel temporarily and which affect this value and the working of the applications that use this infrastructure. The distribution of lost packets over time affects the fulfilment of deadlines the availability of the channel, as can be seen in Fig. 5. Thus, although VAN 2 has the highest PER, in no case did it record the loss of two consecutive packets, guaranteeing delivery of at least one of the two packets transmitted every 200 ms and therefore high availability. VAN 4 gave a higher PER, however, a number of small bursts of packets were lost, reaching up to 8 consecutive, meaning that there was a loss of availability in the channel of around a second. In VAN 5, there was only one burst of lost packets, but it was of 97 consecutive, at approximately the same moment in time as in VAN 4, reaching around 10 s of non availability.

Fig. 4. WiMAX station distribution in a VAN control of utilities network application

	T=50	multicast					T=100	multicast			
C	**VAN2**	**VAN3**	**VAN4**	**VAN5**	**VAN6**	**C**	**VAN2**	**VAN3**	**VAN4**	**VAN5**	**VAN6**
100	0,2%	**1,6%**	0,3%	0,2%	0, 0%	**100**	0,1%	0,4%	0,7%	0,0%	0,0%
1000	0,3%	**1,4%**	0,2%	0,2%	0,0%	**1000**	0,2%	0,4%	**1,1%**	0,0%	0,1%
4000	0,5%	**2,2%**	0,4%	0,2%	0,1%	**4000**	0,8%	**1,0%**	**1,7%**	0,8%	0,4%
10000	**55,6%**	**93,8%**	0,7%	0,0%	0,4%	**10000**	**2,6%**	**4,2%**	**1,8%**	**2,1%**	0,7%
	T=1000	multicast					T=50	unicast			
C	**VAN2**	**VAN3**	**VAN4**	**VAN5**	**VAN6**		**VAN2**	**VAN 3**	**VAN4**	**VAN5**	**VAN6**
100	0,2%	0,2%	**1,0%**	0,0%	0,2%	**100**	**1,4%**	**1,0%**	0,3%	0,55%	**1,6%**
1000	0,1%	0,1%	**0,8%**	0,0%	0,3%	**1000**	**8,2%**	0,6%	0,0%	0,0%	**1,6%**
4000	0,6%	**1,0%**	**2,8%**	0,0%	**6,1%**	**4000**	**86,2%**	**82,9%**	**83,2%**	**81,3%**	**82,0%**
10000	**1,0%**	**3,6%**	**4,3%**	1,5%	**3,8%**	**10000**	**92,8%**	**99,2%**	**99,2%**	**99,9%**	**99,4%**

Table 4. PER obtained in the different experiments.

In the following figure the histograms of the interarrival time for different stations and combinations (C, T) are shown. Fig. 5 shows the data from a PTP link between the CRC and VAN node 6. In the consequent figs. The data is shown of a PTMP link between the CRC and VAN node 4 (Fig. 6) and VAN node 2 (Fig 7). In general, one can see how in all cases for values of T=1000 a tight spike was produced (Katabi & Blake, 2002), although this spike became slightly more spread out as C becomes greater. As we approach the throughput limits of the links with values approximate to T=50, the congestion in the channel alongside the increase in the value of C causes the spike bomb to become more spread out.

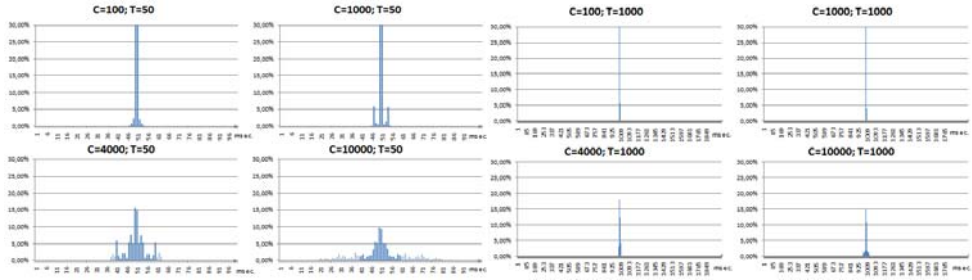


Fig. 5. Histogram of Interarrival time for station 6 and T=50 and 1000 ms.

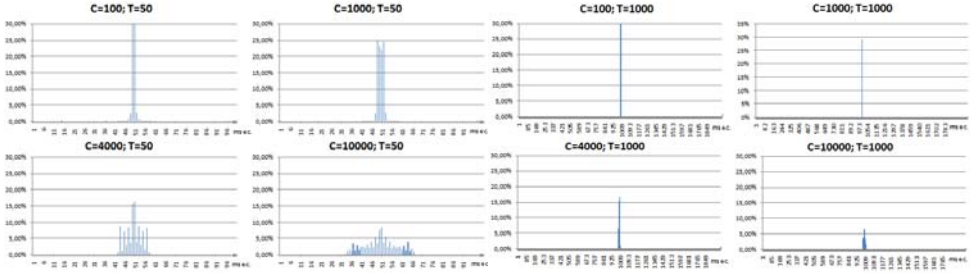


Fig. 6. Histogram of Interarrival time for station 4 and T=50 and 1000 ms.

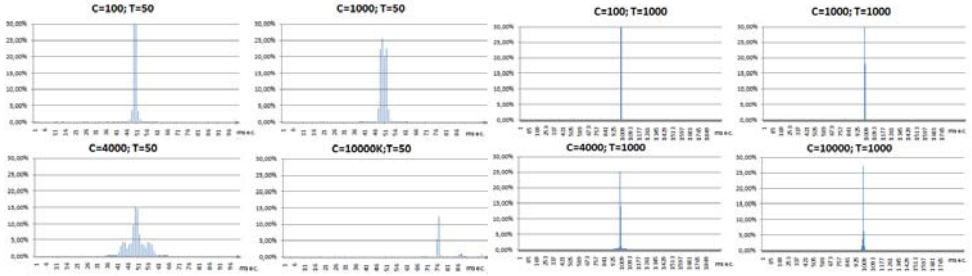


Fig. 7. Histogram of Interarrival time for station 2 and T=50 and 1000 ms.

7. Conclusions

The use of VAN infrastructures to give support to applications in the traditional areas of industrial networks , and in particular in applications with MAN requirements, as can be found in building automation, or the case studied here; utilities network control, is viable but with limitations. The use of wireless networks such as WiMAX allows proprietary networks to be quickly set up at low cost. In spite of this potential, the burst behavior they sometimes display means that the applications must take this into consideration. Thus, the control links in the VAN nodes must be based only on local information within the VAN node, using the VAN infrastructure to improve the working of a global system. Although WiMAX can form part of this VAN infrastructure in a utilities network control application and satisfy the requirements of these applications, it is important to take into consideration the interference produced in the ISM spectrum, such as the fading effects produced in wireless communication, which may produce breaks and lack of availability. These possible problems must be foreseen and considered for the higher levels of the applications so that they do not affect the performance of the system. The need to avoid congestion in the channels and the fact that the channel bandwidths change at times due to variations of the SNR mean that it is necessary to use dynamic bandwidth management mechanisms. The mechanisms must avoid congestion in the cannel, both in the normal working of the system, as well as during moments when there is degradation in the cannel and the consequent loss of bandwidth that this degradation causes.

8. Acknowledgement

We would like to thank Valencia City Council, and especially the Ciclo Integral del Agua, for the support given to this Project. This work was supported by the MCYT of Spain under the project TSI2007-66637-C02-02

9. Index to multimedia Extensions

The multimedia extensions to this publication can be found online by following the hyperlinks from www.intechweb.org

1. ControlVision. Video. Description of the purification network of Valencia city, and their automation properties and heterogeneous networks used.

10. References

Albero, T.; Sempere, V.; Silvestre, J.; Dabbas, P. (2005) Environmental Control System Based on Mobile Devices, *10 th IEEE Int. Conf. on Emerging Technologies for Factory Automation*. Catania, Italy, September. 2005

Balzer, D.; Werner, Th.; Messerchmidt, R. (2008) Public Network and Telecontrol Concepts in Virtual Automation Network, *in Proc. of 17th Wordl Congress The International Federation of Automation and Control*, Seoul, Korea, 13988-13992, IFAC, July, 2008

Brevi, D.; Mazzocchi D.; Scopigno, R.; Bonivento, A.; Calcagno R. and Rusinà F. A methodology for the analysis of 802.11a Links in Industrial Environments. *IEEE International Workshop on Factory Communications (WFCS)*, pp. 165-174. 2006

Cardeira, C.; Colombo, A.; Schoop, R. (2006) Analysis of wireless technologies for automation networking, *in Proc. of Innovative Production Machines and Systems*, 2006

Cena, G.; Valenzano, A.; Vitturi, S. (2008,) Hybrid Wired/Wireless Networks for Real-Time Communications. *IEEE Industrial Electronics Magazine*, March, 2008, 8-20

Crow B. P.; Widjaja I., Kim J. G., Sacai P. T. (1997) IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine, September 1997*.

Cicconetti, C.; Lenzini, L.; Mingozzi, E.; Eklund, C. (2006) Quality of Service in IEEE 802.16 Networks, *IEEE Network* 20(2), March-April, 2006, 50-56

De Pellegrini, F.; Miorandi, D.; Vitturi, S.; Zanella, A. (2006) On the use of Wireless Networks at low level on Factory Automation Systems, *IEEE Transactions on Industrial Informatics,* 2(2), May 2006, 129-143

Decotignie, J.D.; Pleineveaux, P. (1993). A Suarvey on Industrial Communication Networks, *Ann. Telecommunications* 48(9-10) 435- 448

Decotignie, J.D. (2009) The Many Faces of Industrial Ethernet, *Proceedings of the IEEE Industrial Electronics Magazine*, March 2009, 8-19

Ghosh, A.; Wolter, D.R. (2005) Broaenchdband Wireless Access with WiMAX/802.16 : Current performance benchmarks and Future potential, *IEEE Communication Magazine*, 129-136

Gutiérrez, José A.; Naeve, M.; Callaway, E;. Bourgeois, M.; Mitter, V.; Heile, B. (2001) IEEE 802.15.4: A developing standard for Low-Power Low-Cost Wireless Personal Area Networks. *In IEEE Network*, Vol. 15, 12-19.

Hoymann, Christian. (2005) Analysis and Performance Evaluation of the OFDM-based metropolitan area network IEEE 802.16, *Computer Networks* 49, 2005, 341-363

IEEE (2003) IEEE 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LRWPANs), IEEE, October 2003.

JunKeun, S.: Jeong-dong, R.; SangCheol, K.; JinWon, K.; HaeYong, K.; PyeongSoo, M. (2007) A Dynamic GTS Allocation Algorithm in IEEE 802.15.4 for QoS guaranteed Real-time Applications. *In IEEE Int. Symposium on Consumer Electronics* ISCE, 1-6

Kasim, A. (2008) Delivering Carrier Ethernet. Extending Ethernet Beyond the LAN. *McGraw-Hill Communications*.

Katabi, Dina ; Blake, Charles. (2002) Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times. *Technnical Report MIT-LCS-TR-828*, MIT, 2002

Koubba, A.; Alves, M.; Tovar, E. GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks *in 20th International Parallel and Distributed Processing Symposium*, 2006. IPDPS 2006.

Li, B.; Quin, Y.; Gwee, C.L. (2007) A Survey on Mobile WiMAX, IEEE Communications Magazine, December, 2007, 70-75

Matkurbanov, Pulat.; Nat, Kumoh.; Kim, Dong-Sung. (2006) A Survey and Analysis of Wireless Fieldbus for Industrial Environments, *in Proc. of Conf. SICE-ICASE,* Busan, Korea, 5555-5561, ICASE, October, 2006

Metro Ethernet Forum (2006) MEF Technical Specification, MEF 10.1, "*Ethernet Services Attributes - Phase 2*", November 2006

Metro Ethernet Forum (2004) MEF Technical Specification, MEF 6, "*Ethernet Services Definitions - Phase I*". June 2004

Metro Ethernet Forum (2008) MEF Technical Specification, MEF 6.1. "*Ethernet services definition – phase 2*". April 2008.

Neumann, P. (2003a) Virtual Automation Network – Reality or Dream, *in Proc. of IEEE Int. Conf on Industrial Technology*, Maribor, Slovenia, 994-998, IEEE, December 2003

Neumann, P. (2003b) Virtual Automation Network – Problems to be solved, *in Proc. of IEEE Int. Conf on Emerging Technology and Factory Automation*, Lisbon, Portugal, 3-6, IEEE, September, 2003

Neumann, P. (2007) Communication in industrial automation - What is going on?, *Control Engineeting Practice* 15 1332-1347

L. Rauchhaupt, Lutz. (2002) System and device architecture of a radio based fieldbus—the RFieldbus system, *in Proc. Workshop on Factory Communications Systems (WFCS)*, Vasteras, Sweden, 2002.

Sauter, T. (2005a). Fieldbus System: History and Evolution in *The Industrial Communication Technology Handbook*, CRC Press, 7.1-7.39

Sauter, T. (2005b). Integration Aspects in Automation – a Technology Survey, *in Proc. Emerging Technologies and Factory Automation*, vol. 2, 255-263

Sauter, T. (2007). The Continuing Evolution of Integration in Manufacturing Automation IEEE Industrial Electronics Magazine, 1(1), Spring 2007, 10-19

Scheible , G. ; Dzung, D. ; Endresen, J. ;Frey, Jan-Erik. (2007) Unplugged but Connected. *IEEE Industrial Electronics Magazine*, Summer, 2007, 25-34

Sempere, V. ; Albero, T.; Silvestre, J. (2003). Supervision and Control System of Metropolitan Scope Based on Public Communication Networks, *Proc. of IFAC Fieldbus Systems and Their Applications (FeT*), IFAC, Aveiro, Portugal, 2003

Sempere, V. ; Silvestre, J. ; Albero, T. (2004) Remote Access to Images and Control Information of a Supervision System Through GPRS. *Telematics Applications in Automation and Robotics*. IFAC, Helsinki, Finland, 2004

Sempere, V. ; Albero, T. ; Silvestre, J. (2006). Analysis of communication alternatives in heterogeneous network for a supervision and control system, *Computer Communications 29, 2006,* 1133-1145

Silvestre, J.; Sempere, V.; Albero, T. (2007). Wireless Metropolitan Area Networks for Telemonitoring Applications And Development, *in Proc. of Fieldbus Systems and Applications (FeT)* IFAC, Toulousse, France, November, 2007

Thomesse, Jean Pierre (2005) Fieldbus Technology in Industrial Automation, *Proceedings of the IEEE* 93(6) 1073-1101

Torrisi, N. ; Brandao, D. ; Pantoni, R.P.; Oliveira, J.F.G. (2007) Design of a communication system for integration of industrial networks over public IP networks. *In 5th IEEE International Conference on Industrial Informatics (INDIN),* 201-206

Varga, Pál. Kún, Gergelry. (2005) Utilizing High Order Statistics of Packet Interarrival Times for Bottleneck Detection. *In Proce. of the End-to-End Monitoring Techniques and Services (E2EMON) ,* 152-163

Willing Andreas. (2003) Polling-based MAC protocols for improving real-time performance in a wireless Profibus *in IEEE Transactions on Industrial Electronics*, Vol. 50, No. 4, August 2003.

Willing, Andreas; Matheus, Kirsten; Wolishz, Adam. (2005) Wireless Technology in Industrial Network, *Proceedings of the IEEE* 93(6) 1130-1151

Willing, Andreas. (2008) Recent and Emerging Topics in Wireless Industrial Communications: A Selection, *IEEE Transactions on Industrial Informatics,* 4(2), May 2008, 102-124

7

Wireless Sensor Networks for Networked Manufacturing Systems

L. Q. Zhuang, D. H. Zhang and M. M. Wong
Singapore Institute of Manufacturing Technology
Singapore

1. Trend towards the Networked Manufacturing Systems

With the continuing trend towards globalization and focusing on high value with low volume, the manufacturing system architecture is evolving from traditional centralised model to the distributed model and to the recent networked model. In the modern manufacturing environment, the manufacturing systems are in networked framework via a variety of networking communication systems integrating the heterogeneous collections of manufacturing equipment, devices and real-time information. Such networked manufacturing system monitors and controls with the clear objective of maximizing the Quality of Service (QoS) provided by the prevailing manufacturing resources and to achieve near zero down time operations.

For such networked framework for manufacturing and execution, the key issue is how to sustain networked manufacturing operations' capability in providing robust, zero-breakdown performance with various uncertainties (e.g., machine breakdown, device malfunction, sensor failure, communication delay, and data loss). The challenge therefore lies in the identification, characterization and generalization, and development of technologies to minimize their adverse impact on system performance and serviceability.

Hence from the aspect of system availability, substantial research efforts have devoted to the prognosis of equipment condition. The objective is to prolong the useful life of critical manufacturing elements by estimating the life span of components, devices and equipment. This kind of technology is widely utilized for maintenance operations with the help of the advancement of sensor and sensor fusion techniques. From the perspective of manufacturing control, the machine health information is able to reflect the prevailing health status of the equipment and influence the control decision for networked manufacturing systems. The supervisory controllers are able to take proactive measures to ensure the continual operation by executing the fall back strategies.

However from the aspect of system performance, considering the networked manufacturing systems that are connected through heterogeneous networks, hence, the decision making process is distributed to the individual control agents based on the common global goal. The collaborative decision ensures to make best choice from the consideration for all possible options. Various network models can be applied for such networked manufacturing system, particularly the wireless communication for sensory information fusion. Though such

wireless communication systems provide more flexibility for the system implementation, such infrastructure imposes a lot of other constraints and uncertainties that have a major impact on the system stability and performance. Such important technological challenges appeal to a solid theoretic methodology to properly handle such uncertainties to promise the converged collaborative decision making.
In a nutshell, in networked manufacturing environments, the hybrid model involves both the supervisory controllers and the lower-level controllers. This kind of hybrid model is to be integrated with manufacturing resource management at higher system level and machine health condition at lower equipment level. Hence, it will provide optimized control strategies based on distributed model predictive control (MPC) method for the networked manufacturing systems composed of a myriad of decentralized equipment and sub-systems. The distributed MPC involves the decomposition of the overall control optimization problem into a number of small coupled optimization problems. The allocated resources and their constraints provided by the available resource as well as the health information of the devices will be incorporated in the optimization. In the control design, network uncertainty including data losses, data disorder, random delays, and constraints in bandwidth and sensor node energy will also be considered.

2. Technologies and Standards for Networked Sensing and Control

For networked manufacturing systems, the high value manufacturing activities are more dependent on continuous real-time information from the manufacturing environment for decision making and performance optimization. For such industrial automation applications, considering the increasing requirement for more intelligent distributed control and more demanding needs of condition-based monitoring, the opportunities to apply techniques of networked sensing and control are exploding with decreasing cost of embedded processors, sensors and networking devices for such applications.
The advent of wireless communication and micro-electromechanical systems (MEMS) technologies has provided possibility to bring embedded controller, sensors and wireless communication module together as one integrated component for networked sensing and control systems, enabling remote sensing and actuation over wireless channels. Such sensors and actuators based on standard wireless interface and protocols provide new paradigm for factory automation as they converge the sensing, control, computation and communication capabilities into a single tiny node.
Hence, wireless sensor network (WSN), based on the above concept, has been proven to be one of the best platforms for networked sensing and control systems for the factory automation applications in the networked manufacturing systems. WSN is a mesh network consisting of small sensor nodes that acted as the smart layer between virtual and physical world. Integrated with capabilities of communication, computation and various sensing, each node can be imaged as an intelligent tiny device (Fig. 1) with battery energy support for distributed monitoring, estimation and control applications. With wireless communication capability, data captured by individual nodes of WSN from the observed phenomenon can be processed locally or autonomously delivered to a gateway for distributed collaborative information processing and decision making.

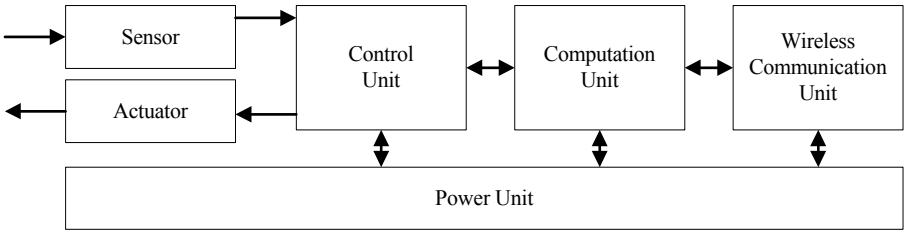


Fig. 1. Functional Blocks of Wireless Sensor Network (WSN) Node

Hence, WSN provides new paradigm for real-time control applications for military, industry and environment monitoring purpose. Applying various sensors for different industrial control and monitoring applications that are supported by the IEEE 802.15.4 communication standard, WSN has demonstrated great potentials for networked sensing and control systems. Fig. 2 shows the major categories of sensors for WSN applications and their market growth trend in future industrial automation. Many products have been available in recent years (Fig.3).

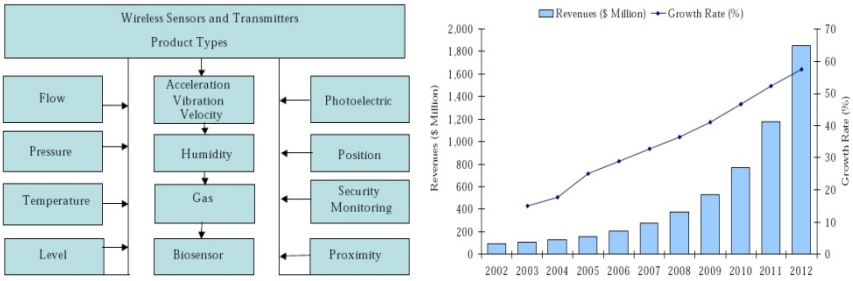


Fig. 2. Wireless Sensors and Market Trends (Source: Frost & Sullivan)

Fig. 3. MicaZ: WSN product (www.xbow.com)

The wireless communication and services have greatly enabled e-manufacturing providing more information efficiency for industrial applications at enterprise level. However, at the shop floor level, the fundamental networking technology for control information is still based on fieldbus (IEC 61158 standard) providing wired link between program logical controllers (PLC) and other physical devices such as transducers, actuators, motors and switches to form the control chain. Although, in recent years, the radio-frequency identification (RFID) technology provided electronic identification labels for object identification and asset tracking in the factory yet lacks support for sensing, information processing and actuation by its transponders. To simplify the machinery control and

monitoring in hash environments and to reduce the cost of cable installation and maintenance by using mobile device, wireless personal area network (WPAN) based on IEEE 802.15 standards become new foundation technologies in factory automations.
Standardization of WSN is one of the most important industrial drives to its commercial success for factory automation application. The standardization processes are focus in two areas: network protocol and sensor interface. The prior is described in the latest ZigBee specification which is defined on top of IEEE 802.15.4. The later is also referred as transducer electronic data sheet containing interface information connected to any kinds of sensors and this standard is defined in the IEEE 1451. The standardization helps to reduce the cost of the system deployment and shorten the cycle of development.

2.1 IEEE 802.15.4 and ZigBee

IEEE 802.15 Wireless Personal Area Network (WPAN) defines standards for short distance wireless networks including following five sub-standards: IEEE 802.15.1 for Bluetooth used for short range devices, IEEE 802.15.2 for coexistence, IEEE 802.15.3/3a for high data throughput with low power consumption in short distance which is also known as ultra wideband (UWB), IEEE 802.15.4/4a for low rate WPAN and IEEE 802.15.5 for mesh network. Especially, the IEEE 802.15.4 defines a standard for a low data rate solution with long battery life and very low complexity which can be used in factory control and monitoring. It is intended to operate in an unlicensed, 16 channels in the 2.4GHz industrial, scientific and medical radio band or 10 channels in the 915MHz or one channel in the 868MHz band (Fig. 4).

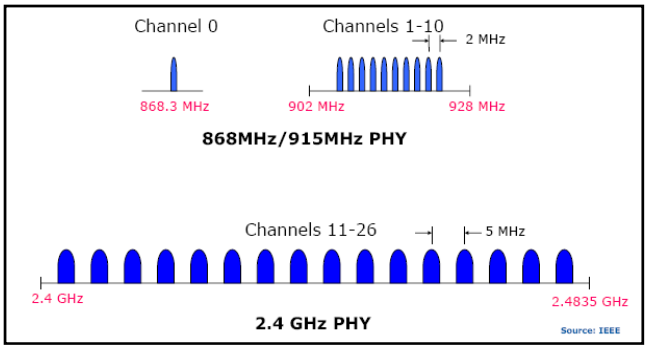


Fig. 4. IEEE 802.15.4 Channel Allocation (source: IEEE)

With the completion of standardization of the Media Access Control (MAC) Layer and Physical (PHY) Layer of 802.15.4, the industrial focus has been shifted to upper protocol layers and application profiles. The ZigBee Alliance is a group of companies which maintain and publish the ZigBee standard. ZigBee is the standard designed to address the unique needs of most real-time control and monitoring application in the factory automation. ZigBee has developed upper layers of the stack and application profile, which is shown in Fig.5. The ZigBee Alliance (www.zigbee.org) has been setup to enable reliable, cost-effective, low-power, wirelessly networked monitoring and control products based on IEEE 802.15.4. The specification defines three throughput levels: 250 Kb/s at 2.4 GHz, using 10

channels; 40 Kb/s at 915-MHz, using 6 channels; and 20 Kb/s at 868 MHz using a single channel.

Application Framework	ZigBee Device Object (ZDO)
Application Support Sub-layer (APS)	
Networking Layer (NWK)	
Data Link Layer	
IEEE 802.15.4 LLC	IEEE 802.2 LLC Type 1
IEEE 802.15.4 MAC	
IEEE 802.15.4 PHY (868/915 MHz)	IEEE 802.15.4 PHY (2400 MHz)

Fig. 5. IEEE 802.15.4 and ZigBee Stacks

2.2 IEEE 1451

As transducer is key part in the WSN used for industrial control and process monitoring, coherent open standard for sensor interface provides foundation for market adoption and successes. The standard provides seamless integration, interoperability and scalability for larger WSN to be deployed in the shop floor and coexist with existing wired control and monitoring systems.

IEEE 1451 is the family of standards for a networked smart transducer interface which provides the common interface and enabling technology for the connectivity of transducers to control devices, data acquisition systems and fieldbus. The key definition of data formats and communication protocols of Transducer Electronic Data Sheet (TEDS) have been specified in IEEE 1451.2 (1997). IEEE 1451.1 (1999) developed a smart transducer object model in frame of network-capable application processors (NCAPs) to support multiple control networks. IEEE 1451.3 (2003) extends the parallel point-to-point configuration to distributed multidrop systems. IEEE 1451.4 (2004) is an emerging standard for adding plug and play capabilities to analog transducers via a mixed-mode interface of analog and digital operating modes. IEEE 1451.5 (2007) defined the wireless communication and TEDS formats and specified sensor-to-NCAP connection for IEEE 802.11 family, IEEE 802.15 family or Ultra-wideband (UWB) connections. IEEE 1451.6 (Draft) proposed the TEDS using the high-speed CANopen network interface for measuring devices and closed-loop controllers. Fig. 6 shows the general framework of smart transducer interface of IEEE 1451 family.

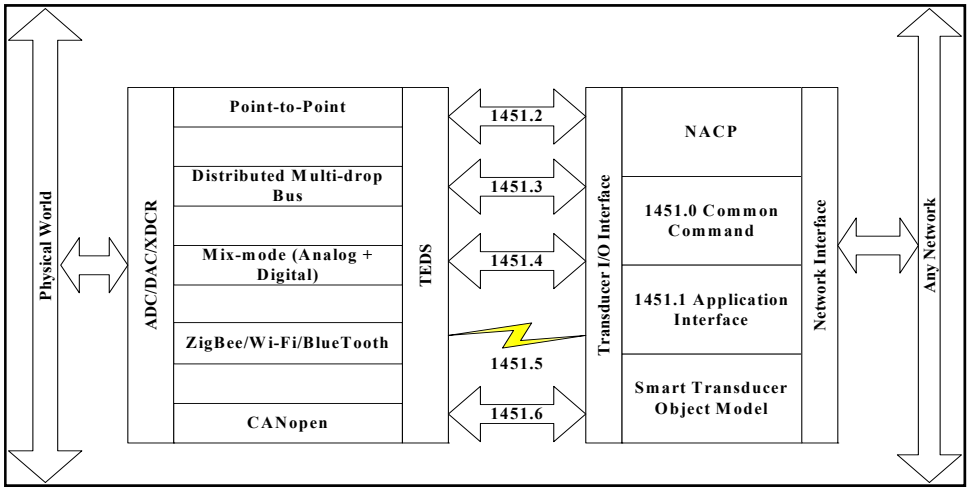


Fig. 6. IEEE 1451 Family (source: IEEE)

2.3 Standard Architecture for Condition-based Maintenance Application

As factory automation is moving towards more advanced, sophisticated and expensive machinery and devices, it calls for information exchange standard and architecture for the diagnostics and maintenance at application level. Intelligent condition-based maintenance (CBM), a maintenance philosophy for machinery and equipment, is a form of proactive maintenance that make use of sensors, sensor networks and computational intelligence techniques to efficiently forecast incipient failures and predict the remaining useful life of the equipment, based on real-time assessment of equipment condition, to perform maintenance only when there is objective evidence of need, so as to ensure near-zero downtime, and to minimize the total cost of maintenance.

Open System Architecture (OSA) for CBM has been developed and promoted by the team participants from the university, standard consortium, industry and military organization to demonstrate the system architecture that facilitates interoperability of CBM software modules. As the results, the seven functional layers are defined within the OSA-CBM development process: Data Acquisition, Data Manipulation, Condition Monitor, Health Assessment, Prognostics and a Human Interface or Presentation layer. Each layer has the capability of requesting data from any functional layer as needed and data flow will occur between adjacent functional layers.

The open architecture has finally evolved into a set of standard guidelines including ISO 13374 for Condition Monitoring and Diagnostics of Machines with four parts: General Guidelines, Data Processing, Communication and Presentation; Machinery Information Management Open Systems Alliance (MIMOSA) Open System Architecture for CBM as well as for Enterprise Application Integration (2006). Many applications have been developed under this guideline in recent years (Djurdjanovic et al., 2003; Chidambaram et al., 2005; Park et al., 2006).

3. Research Issues and Challenges

The new element in the networked control and sensing is the network communication. With each component making its own control decision locally based on own or neighbouring sensory data, they can coordinate each other and be able to achieve global targets of many industrial control and monitoring applications. The network architecture allows sensors, and other control agents such as actuators and controllers to be interconnected together, using less wiring, and requiring less maintenance than the point-to-point architecture. Such architecture also makes it possible to distribute processing functions and computational traditional loadings into several small units. Moreover, distributing control between multiple processors can make the system more flexible and fault-tolerant whereas centralized control suffers from the drawback of a single point of failure. Such networked sensing and control systems which built on sparse and unreliable networked components posed new research challenges from two aspects: control over networks and control of networks (Murray et al., 2003).

3.1 Control over Networks

For the control over packet-based communication channels, several keys issues have been addressed making networked sensing and control systems distinct from other control systems in the face of bandwidth constraints, channel fading and competition for network resources (Nair et al., 2007)

As many networked sensing and control systems are based on wireless networks, control performance tends to degrade when wireless communication channels show the characteristics of packet loss, packet delay and packet disorder; therefore communication reliability has great impacts on system stability (Nair et al., 2007). The relationship between the stability of the system and data rate of communication was explored at both Transmission Control Protocol (TCP) level and User Datagram Protocol (UDP) level (Schenato et al., 2007). The relationship between the stability of the system and data distortion was also explored to show the unstable error region by analyzing the statistical convergence properties of the error covariance matrix of sensor measurement (Liu & Goldsmith, 2004) and some results also revealed the upper bound of the expected error covariance for convergence (Elia & Mitter, 2001) as well as the critical value for the arrival rate of observations for bounded state error covariance (Sinopoli et al., 2004).

Researchers now pay more attention on efficient quantizer design for obtaining stability for such data-rate limited control system and have showed that the coarsest quantizer for quadratic Lyapunov function is logarithmic (Ishii & Francis, 2002). More significant results were reported applying sector bound approach for performance analysis of logarithmically quantization systems (Fu & Xie, 2005).

3.2 Control of Networks

For the control of networks, some basic problems have been widely explored in the research community including network congestion control, network routing strategies, transmission power management and application level performance analysis based on quality of service (QoS). These efforts have brought network protocol design into modular-based layered architecture that has evolved into seven-layer Open Systems Interconnection (OSI) model including physical, data link, network, transport, session, presentation, and application.

Due to the characteristics of nodes uncertainty, variation and limited resources such as communication channels, network bandwidth and power supply, the dynamic characteristic features of WSN infrastructure require research work for the design and development of network protocols, topology, routing, data dissemination, power scheduling, programming methods and data abstraction. These issues lead to the efforts and results for standards of WSN infrastructure that are important drives to commercial success of WSN especially for factory automation applications. These standards include IEEE 802.15.4, Zigbee and IEEE 1451.

3.3 Control and Communication Co-design

Facing the challenges from both control demand and communication provision, there is a need to take a holistic approach to both aspects for building reliable application while considering the unreliable infrastructure for the above scenarios. Hence deciding the right architecture for the convergence of communication, control, and computing becomes one of the research challenges when applying holistic view for both communication performance and control performance (Murray et al., 2003). Since classical communication theory and control theory have not shown a ready unified mathematical model for these new research challenges, there is a need to develop new approaches and techniques for optimization problems in networked sensing and control systems.

As WSN is a data-driven computation platform for factory monitoring, process control and supervisory control, optimization is needed for both control and communication performance. Such trade-off requires new design methods for the traditional layered OSI model with consideration of sensing and control objectives (Goldsmith, 2005). It requires the cross-layer consideration rather than layer by layer modulation for system optimization involving different factors from multiple layers. The cross-layer consideration is driven by requirement at application level due to the nature of WSN-based applications.

4. Optimization Techniques for WSN in Networked Sensing and Control

With emerging technology of WSN as low power pervasive computation platform for monitoring and distributed control, research on WSN in area of cross-layer design becomes more important. Comparing with the OSI-based model which is more connection oriented, with less constraints and for more general purposes of usage, WSN is task oriented, with more constraints, more data-driven features and application specific requirements. Hence, research work on WSN in area of cross-layer design becomes more important due to these unique characteristics such as distributed network management, distributed decision and energy-constraints for the individual nodes. The tradeoffs between network lifetime, node connectivity, data accuracy and network throughput require richer interactions among the physical, networking and application layers (Fig. 7). The motivating drives for cross-layer design fall into two categories: from infrastructure aspect such as prolonging the network lifetime (Hoesel et al., 2004) and from application aspect such as providing reliable data fusion for control and estimation requirement (Xiao et al., 2006).

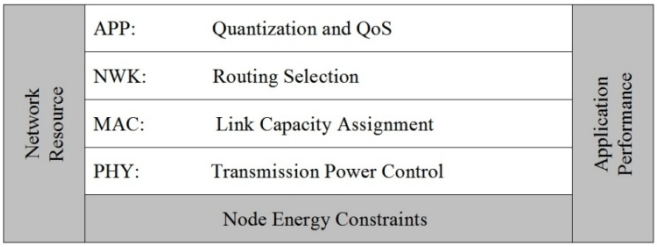


Fig. 7. Cross-layer Optimization Model

There are two directions of the research work to consider and apply cross-layer design and optimization for the sensing and control applications using WSN:

- Cross-layer design from aspect of network utility and performance optimization
- Cross-layer design from aspect of sensing and control performance

4.1 Cross-layer Design for WSN from Network Infrastructure Aspect

From infrastructure aspect, there are several issues to be considered for cross-layer design:

- Power constraints: as WSN node relies on battery power, it requires efficient power management to maximize the lifetime of the node as well as the system. Power consumption is not only related with the physical layer (PHY). It is also decided by a set of variables at different layers such as media access control (MAC) layer and networking (NWK) layer. Hence power consumption management requires joint optimization of factors across PHY, MAC and NWK layers.
- Network properties: due to the possible channel error, the node application is vulnerable to packet loss and disorder. The mobility nature brings more issues to the interference at PHY layer, access scheduling at MAC layer and network routing at NWK layer. Maximization of the network utility should be resolved by the coordination for PHY, MAC and NWK layers.

Most of the studies from infrastructure aspect have shown the benefits by joint design across NWK, MAC and PHY layers without considering QoS at application level. For example, based on the single layer energy optimization strategies for MQAM and MFSK modulation, Cui et al. demonstrated modulation optimization at physical layer with energy model considering two power status: active and sleep (Cui et al., 2003); then they extended the approach by introducing a variable-length interference-free TDMA scheme to minimize the total energy consumption by joint consideration of MAC and PHY layers that was solved by convex relaxation methods (Cui et al., 2004); and they further extended joint optimization model to the routing layer and showed results that optimization solution should go for multi-hop routing when only transmission energy was considered, but need to go for single-hop transmissions if circuit processing energy was considered as well (Cui et al., 2005). Cui summarized the above results on the cross-layer optimization in WSN with energy constraints by modulation, physical transmission and network communication routing and showed the benefits by joint design across routing, MAC, and PHY layers in his PhD dissertation.

Madan et al. proposed a cross-layer optimization model for minimizing the maximum energy consumption of any node in the network for the purpose of maximizing the network lifetime. This approach considers load balancing factor in the multi-hop model, channel

utilization based on TDMA schema as well as transmission power and transmission rate. As this non-linear problem of multi-layer optimization problem for lifetime maximization model is NP-hard, it was simplified into mixed integer convex optimization problem by convex relax using interference-free TDMA assumption (Madan et al., 2005). Their work showed a distributed algorithm which starting with a feasible suboptimal solution and finally converging to optimal solution through limited iterations.
Cross-layer approach has also been tried out in the TinyOS which is de-factor standard and open source embedded operating system for many sensor network platforms. An adaptive cross-layer framework called TinyCube provides a generic interface and a repository for the multi-layer information exchange and management (Marron et al., 2004). Under the IEEE standard 802.15.4, cross-layer between MAC and PHY has been explored using a distributed algorithm to manage the activities of sensor nodes (Misic et al., 2006).

4.2 Cross-layer Design for WSN from Application Requirement Aspect

When feedback loop of a control system is built on wireless communication channels, the communication performance has major impacts on the performance of control systems. However QoS requirements in application layer play a leading role for cross-layer optimization modelling, hence more and more work from application aspect showed the importance that cross-layer optimization should be driven by the consideration of application layer for WSN application.
Mostofi et al. analyzed tradeoffs between communication and objective tracking target, and developed algorithms to handle the optimization problem with interference between physical energy consumption and tracking accuracy of application layer for the real-time tracking applications; and they also proposed the cross-layer strategy of sharing information between the PHY layer and application layer in the design of Kalman filter for the real-time estimation based best linear unbiased estimator (BLUE) for the location estimation which demonstrated the estimation benefits and infrastructure cost saving (Mostofi et al., 2005).
Liu et al. studied the tradeoffs of data rate, time delay and packet loss in the communication link layer design. They proposed an analog soft-decoding wireless link design for robust distributed control systems (Liu & Goldsmith, 2005).
Cooperative estimation using WSN with energy-efficient method is another important research area of cross-layer optimization. Xiao et al proposed a joint optimization approach using BLUE to minimize the noise distortion while consuming minimum power of sensor nodes. In a scale signal joint estimation case, the approach tried to minimize the total transmitting power by the optimal sensor scheduling to turn off the node with higher mean square error (MSE) or lower the quantization level however still keeping overall MSE under threshold (Xiao et al., 2004). Xiao extended his work to the joint estimation problem for vector signal case, and use MSE as performance measurement as well (Xiao et al., 2008). In this case, they proposed an optimal linear decentralized estimation model with coherent media access control and resolved the problem analytically for the case of noiseless MAC and solved the noisy MAC problem using semi-definite programming (SDP). However their models only support the star networking architecture where there is only one hop from end node to fusion centre. Real-time object tracking and position estimation were widely used test bed for demonstration of applying Kalman filter on WSN. Sun et al used Kalman filter for target states estimation for the linear model with multiple packet dropouts (Sun et al. 2008).

However the above work only provided piecemeal analysis and solutions for specific cases which are focused either on control aspect or on communication aspect and the holistic view of whole stack is not presented. In the face of complex interactions required by the cross-layer approach, there is a need to build a general model to address the cross-layer issue with network utility maximization and different targets at application layer. Hence the more formal and common mathematical language is required to provide fundamental optimization modelling and techniques for cross-layer design. Such mathematical theory of network architecture is called "Layering as Optimization Decomposition" which is defined as top-down approach to design protocol stacks (Chiang et al., 2007).

4.3 Mathematics Framework Cross-layer Design and Optimization

Cross-layer optimization requires more quantitative analysis using a common language and unified mathematical framework. Mathematical decomposition techniques have emerged as a foundation for communication network maximization problems in recent years and myriad work was inspired by Kelly's model (Kelly et al., 1998). Kelly's model of optimization decomposition framework provides a common language for top-down design based on coordinating sub-problems from PHY, MAC and NWK layers. The general model is shown in Fig. 8 which considers the optimization targets at application layer as well. The model also shows general optimization targets from point of individual layer.

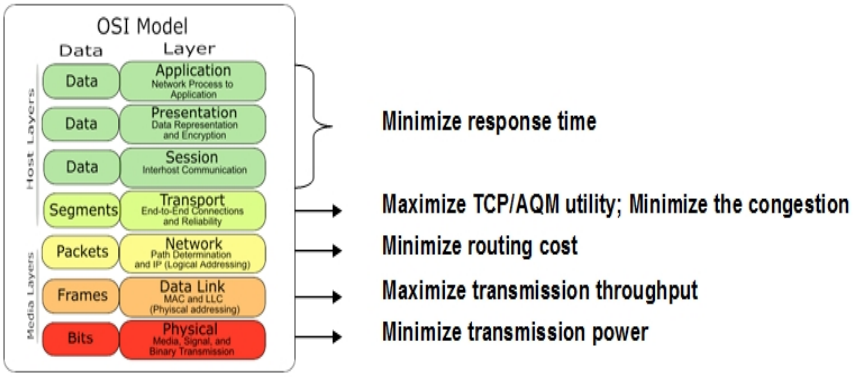


Fig. 8. Cross-layer Optimization Model

The theory proposed by Kelly used a network utility maximization (NUM) as framework to address the cross-layer issues (Kelly et al., 1998). Consider a system for TCP congestion control with routing matrix R, link capacity c, source rate vector x and x_s is source rate for source s, the utility function U_s is given in (1).

$$\max \sum_s U_s(x_s) \qquad (1)$$

$$\text{s.t. } Rx \le c$$

Such model defines NUM problem and provides solution through decomposition techniques by dividing the large optimization problem into smaller sub-problems or by

exploring the space of alternative decompositions called duality when necessary. The NUM defines the objective functions and various constraints at different layers. NUM uses primal or dual variables to indicate what need to do and uses constants values to indicate what resource can be used. It normally applies Lagrange duality theory to find optimal solution.
When a NUM formulation is given, decomposition theory is applied rather than centralized computation instead. The optimization decomposition methods can be divided into two categories (Chiang et al., 2007):

- horizontal decomposition method and
- vertical decomposition method

Horizontal decomposition method addresses issues in one layer such as TCP congestion, IP routing and MAC control through the theory of decomposition for non-linear optimization.
Vertical decomposition addresses the multi-layer optimization issues. Some existing work can be categorized into following topics (Chiang et al., 2007):

- Joint optimization of congestion control and routing
- Joint optimization of congestion control and resource allocation
- Joint optimization of congestion control and contention control
- Joint optimization of congestion control, routing and scheduling

Decomposition techniques have provided a common language for network optimization problems. These problems can be resolved by primal-dual method, interior method, quadratic programming and geometric programming method etc.
The basic principle of decomposition is to make original complex problem into independent smaller sub-problems in order to be resolved in a distributed way. Most widely used methods are classified into primal and dual decomposition that can be developed into specific distributed algorithm to resolve. More detailed techniques were discussed on optimization decomposition such as decoupling constraints, decoupling objective function and other alternative decomposition including partial decomposition, multi-level decomposition (Chiang et al., 2007).
Despite the progress for the decomposition of a generalized NUM that has been widely reported over the past few years, there are still many open research issues to be resolved such as the rate of convergence; stochastic issues at channel layer, packet level and session level; non-convexity issues etc. To draw the conclusion, cross-layer optimization and related decomposition techniques provide a top-down approach to design network protocol and allocate resource for performance optimization target. Yet few research works have involved object functions for application level targets. Hence control application with control performance and communication networks with resource constraints and optimization provides a common field for new research initiatives. The next section will show some basic ideas and basic system modelling for the control and monitoring applications in networked manufacturing systems. It shows new approaches which are different from the existing methods. The basic system modelling, simulation work, trial industrial prototype will be presented in the following section. The advantages and limitations of the current research approaches and proposal for the future work will be discussed in last section.

5. Industrial Application Case Study

Two industrial application areas are chosen as examples for system design and optimization of sensing and control using WSN. One area is manufacturing asset tracking and

management using WSN to overcome the constraints of RFID technologies which is mainly suitable for pure identification applications. Compared with RFID, WSN is able to track the objects in real-time over large areas without any additional infrastructure required. Another focus area is intelligent condition-based maintenance, a maintenance philosophy for machinery and equipment, which is a form of proactive maintenance that make use of sensors, sensor networks and computational intelligence techniques to efficiently forecast incipient failures and predicts the remaining useful life of the equipment.

5.1 Application 1: Asset Tracking in the Networked Manufacturing Systems

Though UWB or RFID is also widely applied to asset tracking application, but both technologies require power supply for their readers. For the case of some specific requirements such as monitoring the activities of maintenance workers in the cabin of aircraft in airport hangar (Fig. 9), as there is no power supply available for any maintenance activities on aircraft, hence in this situation, WSN is the best choice as both UWB and RFID readers require power supply.

Fig.9. Aircraft Hanger and Cabin

5.1.1 Network Modeling

The model consists of a global fusion centre (coordinator) for estimation, local fusion nodes or network relay nodes (routers) for data dissemination and routing and end nodes (end devices) linking with sensors for getting observation of physical parameters (Fig. 10). This architecture considers generic networking topology which includes both cluster topology and mesh topology for different application requirements with random distribution of sensor nodes.

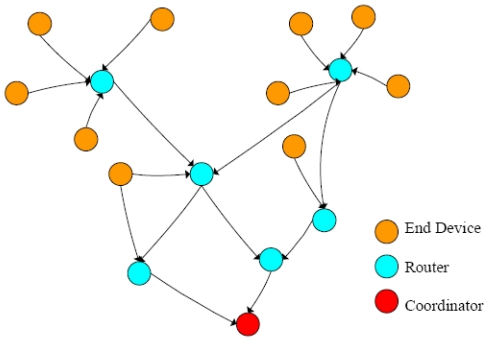


Fig.10 Topology for Networked Sensing and Control

When network routing is considered for WSN with N sensor nodes and L links between nodes, the routing matrix R is defined by an $L \times N$ matrix shown in (2).

$$\mathrm{R}_{lk}: \begin{cases} 1, & \text{if link l is in any path of source node k;} \\ 0, & \text{otherwize} \end{cases} \quad \text{where l} \in [1, \mathrm{L}] \text{ and k} \in [1,\mathrm{N}] \tag{2}$$

5.1.2 System Modeling

In the above application framework, we consider the system with N end devices making joint measurement for unknown signals x such as location. The system dynamic model is defined in (3) with zero mean Gaussian noise $\xi[k]$ at discrete time k and A is dynamic characteristics matrix for the system.

$$x[k+1] = Ax[k] + \xi[k] \tag{3}$$

Let the observation from node i $(i = 1,2,...N)$ be y_i where observation noise v_i is a Gaussian noise with zero mean and observation error covariance matrix is σ_i^2 for each node i. Suppose h is observation characterization matrix, where $h = (\sqrt{g_1}, \sqrt{g_2}, ..., \sqrt{g_n})^T$ and g_i is the channel power gain for the i^{th} node transmission. The system observation model is defined in (4).

$$y_i = hx + v_i \tag{4}$$

If wireless communication is considered between any two nodes, the channel noise is Gaussian noise with zero mean with error covariance ω_i^2. Hence the joint estimation given by best linear unbiased estimator (BLUE) for x is given by (5).

$$\hat{x} = (\sum_{i=1}^{N} \frac{g_i}{\sigma_i^2 g_i + \omega_i^2})^{-1} \sum_{i=1}^{N} \frac{\sqrt{g_i} y_i}{\sigma_i^2 g_i + \omega_i^2} \tag{5}$$

The mean square error (MSE) for the joint estimation is given by (6).

$$\mathrm{var}(\hat{x}) = (\sum_{i=1}^{N} \frac{g_i}{\sigma_i^2 g_i + \omega_i^2})^{-1} \tag{6}$$

When network congestion control is taken into consideration, we define link capacity vector $c = [c_1, ..., c_L]^T$ for all L links and source rate vector $s = [s_1, ..., s_N]^T$ for all N nodes. The following relationship (7) should be satisfied.

$$Rs \le c \tag{7}$$

Link capacity is function of the signal-to-noise ratio (SNR) denoted as (8).

$$\begin{aligned} & c_l = \log(1+\varphi_l) \\ & \text{where} \quad \varphi_l = \frac{g_{l,l} p_l}{\sum_{l \neq k} g_{l,k} p_l + \omega_l^2} \end{aligned} \tag{8}$$

The source rate shall also maintain the threshold ($\| s \| \geq S_0$) in order to keep certain quantization level and reduce the distortion while power usage should be kept to certain limitation ($\| p \| \leq P_0$) to maintain the network lifetime.

5.1.3 System Optimization

In our application framework, we try to minimize MSE error to increase estimation accuracy while keep power consumption under certain level with consideration of TCP congestion control by rate and link capacity constraints. The MSE error is a function of p if observation variance σ_i^2 is constant. The objective is to minimize the MSE $D(p)$:

$$\begin{aligned} \min \quad & D(p) \\ \text{subject to} \quad & Rs \leq c(p) \\ & \| s \| \geq S_0 \\ & \| p \| \leq P_0 \\ & s \geq 0 \\ & p \geq 0 \end{aligned} \tag{9}$$

The above optimization problem can be resolved by decomposition techniques in principle. By introducing the Lagrange multipliers (λ_0, λ_1 and λ_2), the Lagrangian can be expressed as follows:

$$L(p, s, \lambda_0, \lambda_1, \lambda_2) = D(p) + \lambda_0 (Rs - c(p)) + \lambda_1 (S_0 - \| s \|) + \lambda_2 (\| p \| - P_0) \tag{10}$$

So the problem can be decoupled into following two sub-problems for $\forall \lambda_0$, $\forall \lambda_1$ and $\forall \lambda_2$:

$$\begin{aligned} \min \quad & \lambda_0 Rs + \lambda_1 S_0 - \lambda_1 \| s \| \\ \text{subject to} \quad & s \geq 0 \end{aligned} \tag{11}$$

and

$$\begin{aligned} \min \quad & D(p) - \lambda_0 c(p) + \lambda_2 \| p \| - \lambda_2 P_0 \\ \text{subject to} \quad & c(p) = \log(1+\varphi) \end{aligned} \tag{12}$$

Problem (11) is source rate control problem and (12) is power control optimization problem. Problem (11) and (12) can be resolved in application layer and network layer respectively. λ_0, λ_1 and λ_2 are interface parameters for the cross-layer optimization.

5.1.4 Hardware and Tools

For the test bed for demonstration, JN3159 wireless micro-controller (www.jennic.com), which consists of a 16MHz 32-bit RISC CPU, 96kB RAM, 4-input 12-bit ADC, 11-bit DACs and UARTs for external sensors2, is applied for implementation. ISM free wireless communication channels on 2.4GHz are utilized, supporting both IEEE 802.15.4 and ZigBee standards with three kinds of antennas, i.e. on board ceramic, SubMiniature version A (SMA) connector and high power uFl connector for different transmission power range requirements. On such platform, link quality indication (LQI) concept defined in ZigBee standard is taken as signal strength between a pair of transmitter and receiver nodes to estimate the distance.

The initial tests focus on power saving by using less transmission power however still achieving reliable estimation results for fusion center. The experiment results for relationship between distance and LQI values using JN5139 were shown in Fig. 11. For short distance, the distance is defined as $d = (256 / LQI)^{\alpha}$ and $\alpha = 2$ in this case.

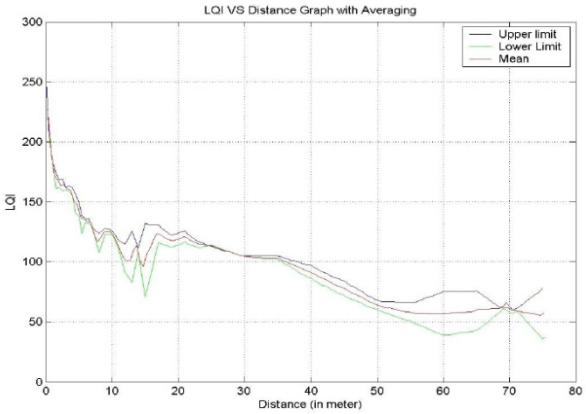


Fig. 11. LQI and Distance relationship

The system design and optimization considers the following factors:

- Transmission power of individual node
- MSE error of LQI value
- Sampling rate

5.1.5 Results

In order to improve the estimation accuracy, we design a fusion algorithm to handle multiple local estimations from a group of sensors. From the estimations of multiple nodes, we discard the LQI readings with higher MSE following the calculation formula in last section, the precision of the location estimation using a few groups of sensor nodes is improved and preliminary results are shown in the demonstration for real time object location. In Fig. 12, we demonstrate the application prototype in the lab environment and show the results on the LCD panel of coordinator node by a block indication. The working principle of estimation process is shown in Fig. 13.

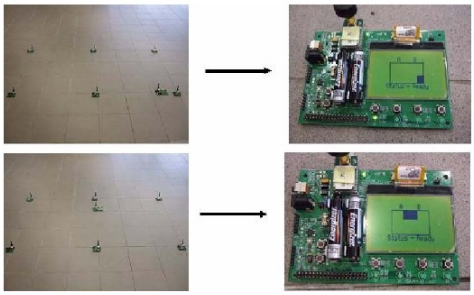


Fig. 12. Test-bed and Demonstration

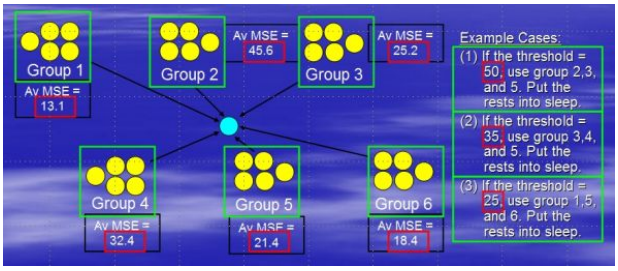


Fig. 13. Optimized Estimation Process

5.1.6 Limitations

Although we build optimization model and improve the estimation accuracy using multiple sensor fusion approach, we still encounter the scalability and mobility issues for such application. While ZigBee provides low power and standard wireless communication protocol, it still has several limitations due to the constraints of Jennic Zigbee stack implementation.

- Constraints for free broadcasting
- Limited memory segment for address recording
- Address-reading first
- Jennic Zigbee stacks default way of networking

The above limitations prevent some ad-hoc application scenarios where nodes with high mobility. It also poses challenges for application implementation for multiple target objects tracking. We are developing new mechanism in the next stage of the project.

5.2 Application 2: Condition Monitoring in the Networked Manufacturing Systems

In this scenario, the Jennic JN5139 wireless sensor evaluation board is used, which contains an analogue to digital peripheral, and can handle the data collection, data processing and transmission functions together. Incorporated with it, a small size, low power, 3-Axis ±3g, frequency bandwidth 1600Hz, iMEMS accelerometer (ADXL330) is connected to the Jennic wireless transmission module through analogue inputs. It draws power from Jennic node and need neither extra conditioning circuit nor amplifier device.

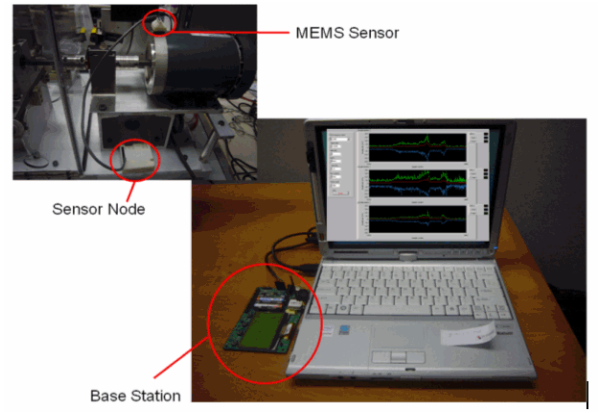


Fig. 14. Wireless Sensor for Machine Condition Monitoring

The system setup is much simpler with lower cost than conventional vibration data collection system. The system also includes the server station to display and store the data and the base-station to bridge the wireless sensor to the server. The system setup is shown in Figure 14.

5.2.1 Maximum Data Acquisition Rate

The maximum data collection rate is confined by

- Sensor node sampling rate;
- Wireless transmission rate; and
- Data transmission from base station to server PC.

Jennic on-board system timer is up to 16MHz. The ADC conversion speed could be configured through the setting of ADC clock division (250K-2MHz) and sampling holding period (2-8 clock periods). The maximum sampling rate could be 10μs. So for one channel ADC, the maximum sampling rate is about 100KHz. Consider the sensor node programming cycle time, the actual sampling rate would be less than 100KHz. For 3-axis acceleration data sampling rate should not be higher than 30KHz.

Normally, the larger proportion of user data in the package will get better transmission efficiency. In the case of Jennic SDK, the package size is up to 128 bytes, the header data size ids 44 bytes, and maximum data in the package is 84 bytes, or less than 66% of wireless transmission is used for actual testing data transfer. To reduce the rate of payload in the transmission, the combination of data set is the strategy to make good use of every package.

Another factor of wireless transmission is the topology of the system. If one node responsible on the data forwarding for other neighbor nodes, the maximum transmission rate for each node will be lower. In mesh network the case would be completed and the networking packages would also share the transmission bandwidth. For the application which request high speed data collection, star topology, and dedicate time slot for each end node is recommended.

If this required data rate is less than the baud rate of the serial communication between the

base-station and server PC, then data could be streaming to the server. When RS232 is configured at baud rate 115200bps, only about 11K bytes could be transferred in one second. The number of data sets to be transferred to server PC also depends on the data format and resolution. 2000 sets of 5 digits data could be received in continuously transfer from RS232 serial port. Our test result also shows that only less than 2% data lost for the 2000 sets/second continuously data collection and transmission to server.

5.2.2 Time Domain Signal Processing

In this test, the machine running speed is less than 3600 RPM, we set the sampling rate 5000/s for the 3 axis acceleration, statistical window size 2000.

The time domain analysis is the majority approach of the vibration signal analysis and especially suitable for on-line condition monitoring using wireless accelerometers. In order to obtain the machine performance from vibration signal in the time domain, following statistical analysis methods have been implemented on sensor node:

- Root mean square
- Peak Value
- Crest Factor
- Kurtosis
- Skewness

5.2.3 Test Cases:

High speed data acquisition is required in machine condition monitoring. With the understanding of the WSNs DAQ limitation and the nature of machine performance, different strategy could be applied for high speed DAQ. Applying above research results, following test cases were created successfully for machine condition monitoring:

Case 1: Motor Current Waveform Monitoring

The induction motor speed limit is 2850 RPM, so the current supply frequency from inverter would not be higher than 50Hz. 1KHz sampling rate is enough for current monitoring - both time domain and frequency domain. Streaming data collection is applied and reliable waveform was collected. The received data is projected in Fig. 15.

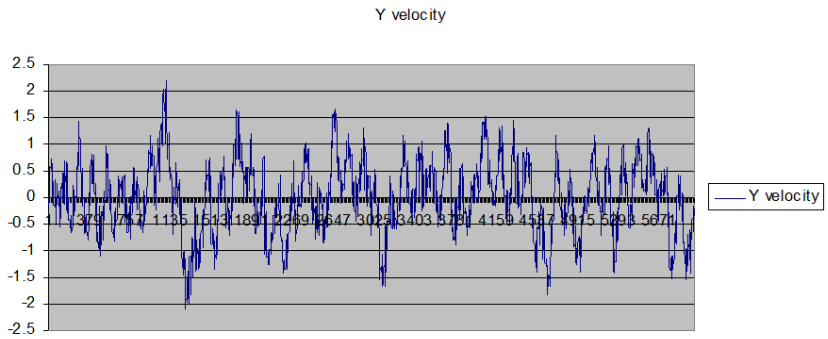


Fig. 15. Current Waveform using WSN

Case 2: Pump Vibration Monitoring

Vibration analysis is a powerful tool for the condition monitoring of rotating machinery. This especially applies to rotating equipment such as pumps. Many faults such as pump and driver misalignment, imbalance of rotating components, worn, loose or damaged parts will cause abnormal vibration of the pump. The level of vibration measured using accelerometers or velocity sensors can be used to indicate the health or integrity of the pump. According to ISO 10816, velocity peak and/or RMS value is typically used for assessing the severity of rotating machinery vibration. When velocity peak or RMS value rises above a threshold, abnormal vibration will be detected. This can be served as a pre-emptive warning for operators to warrant a detailed diagnosis and inspection to rectify the faulty pump. The results are shown in Fig. 16.

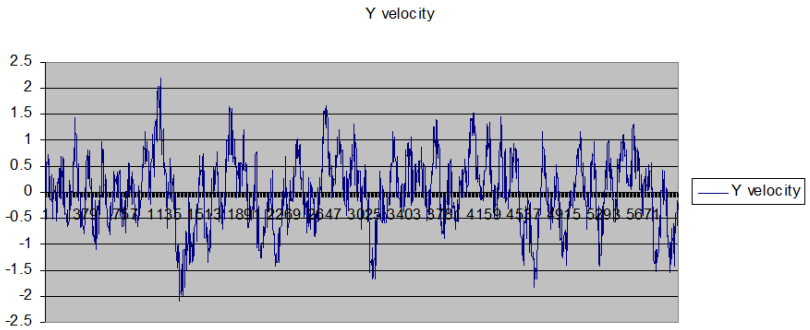


Fig. 16. Velocity Signals from WSN Board

The Jennic sensor node with MEMS accelerometer was used for on-line condition monitoring and status alerting. Using on-board time domain data processing techniques, the peak value and velocity are extracted from a window of high speed acceleration data. An integration algorithm is applied to the acceleration data.

6. Conclusion

The WSN platform has shown its great potential for factory automation applications in the networked manufacturing environment. The future work will extend above models for other types of real-time networked control systems and explore the different control or estimation objectives from these systems. Based on that, more generic model can be explored. Based on the limitation of current approach, several potential research directions are summed as follows:

- Stability Issues for Control and Communication
  The future work will also focus on stability issues for both sensing and control system and routing of network communication. Some analytical models need to address the robustness issues for control system and networking infrastructure in different time scale.
- Duality Gap Issue for Non-Convex Cases
  For many cases, even for the deterministic model for the network resource allocation, there exist situations where NUM is non-concave and constraints are not convex

functions and not separable. There is a need to set up alternative model to analyze and reduce the duality gap in order to achieve global optimal solution. It requires more efforts for such conversion which leading a non-convex problem into a convex one. Although, in theory, such approach can resolve duality gap issues however bearing risk of instability and it will be also acceptable to apply gradient programming using dual-decomposition approach leading to the suboptimal solutions however more stable in practical cases.

- Stochastic Model for Networking Data Flows
  The future work will also look into stochastic features for the system to reflect more dynamic features of packet flows in the queuing networks especially for WSN. It may require more complex techniques or algorithms to handle the coupling issues of networking constraint functions. It will incorporate stochastic network dynamics at different network protocol layers. This leads to challenging models of queuing networks however it can reveal more details of WSN at communication layers.

7. References

Chidambaram, B.; Gilbertson, D. G. ; & Keller, K. (2005). Condition-based Monitoring of an Electro-hydraulic System Using Open Software Architectures, *Proceeding of Aerospace Conference*, pp. 3532-3539, Huntington Beach , 2005.

Chiang, M.; Low, S. H.; Calderbank, A. R.; & Doyle, J. C. (2007). Layering as optimization decomposition: A Mathematical Theory of Network Architecture, *Proceedings of IEEE*, Vol. 95, No. 1, pp. 255-312, 2007.

Cui, S.; Goldsmith, A. J. ; & Bahai, A. (2003). Modulation optimization under energy constraints, *Proceedings of ICC*, Alaska, U.S.A, May, 2003.

Cui, S.; Goldsmith, A. J. ; & Bahai, A. (2004). Joint modulation and multiple access optimization under energy constraints, *Proceedings of Globecom*, Dallas, Texas, December, 2004.

Cui, S.; Madan, R.; Goldsmith, A. J.; & Lall, S. (2005). Joint routing, MAC, and link layer optimization in sensor networks with energy constraints, *Proceedings of ICC*, South Korea, May, 2005.

Djurdjanovic, D.; Lee, J.; & Ni, J. (2003). Watchdog Agent – An Infotronics Based Prognostics Approach for Product Performance Assessment and Prediction, *International Journal of Advanced Engineering Informatics, Special Issue on Intelligent Maintenance Systems*, Vol. 17, No. 3-4, pp. 109-125.

Elia, N. & Mitter, S. K. (2001). Stabilization of linear systems with limited information, *IEEE Transaction on Automatic Control*, Vol. 46, No. 9, Sep. 2001, pp. 1384–1400.

Fu, M. & Xie, L. (2005). The sector bound approach to quantized feedback control, *IEEE Transaction on Automatic Control*, Vol. 50, No.11, Nov. 2005, pp. 1698–1711.

Goldsmith, A. (2005). *Wireless Communications*, Cambridge University Press, 2005.

Hoesel, L. Von; Nieberg, T.; Wu, J.; & Havinga, P. (2004). Prolonging the Lifetime of Wireless Sensor Networks by Cross-layer Interaction, *IEEE Wireless Communications Magazine*, Vol.11, No. 6, pp. 78-86.

Ishii, H.; & Francis, B. A. (2002). Limited Data Rate in Control Systems with Networks, *Lecture Notes in Control and Information Sciences,* Vol. 275, 2002, Springer-Verlag, New York.

Kelly, F. P.; Maulloo A.; & Tan, D. (1998). Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability, *Journal of Operations Research Society*, Vol. 49, No. 3, pp. 237-252, 1998.

Liu, X.; & Goldsmith, A. (2003). Wireless communication tradeoffs in distributed control. *Proceeding of 42nd IEEE Conference on Decision and Control*, Vol.1, No.1, pp. 688–694, December 2003.

Liu, X.; & Goldsmith, A. (2004). Kalman Filtering with Partial Observation Losses, *Proceedings of CDC*, Paradise Island, pp. 4180-4186, 2004.

Madan, R.; Cui, S.; Lall, S.; & Goldsmith, A. J. (2005). Cross-layer design for lifetime maximization in interference-limited wireless sensor networks, *Proceedings of IEEE INFOCOM*, Miami, March, 2005.

Marron, P.; Lachenmann, A.; Minder, D.; Hahner, J.; Rothermel, K. ; & Becker, C. (2004). Adaptation and Cross-layer Issues in Sensor Networks, *Proceedings of ISSNIP*, Melbourne, pp. 623-628, 2004.

Misic, J.; Shafi, S.; & Misic, V. B. (2006). Cross-layer Activity Management in an 802-15.4 Sensor Network, *IEEE Communication Magazine*, Vol. 44, No. 1, pp. 131-136, 2006.

Mostofi, Y.; Murray, R.; & Burdick, J. (2005). On Dropping Noisy Packets in Kalman Filtering Over a Wireless Fading Channel, *Proceedings of ACC*, Portland, pp. 4596-4600, 2005.

Murray, R.; Astrom, K.; Boyd, S.; Brockett, R.; & Stein, G. (2003). Future directions in control in an information-rich world, *IEEE Control Systems Magazine*, Vol. 23, No. 2, pp. 20-33.

Nair, G.N.; Fagnani, F.; Zampieri, S.; & Evans, R.J. (2007). Feedback control under data rate constraints: An Overview, *Proceedings of the IEEE*, Vol. 95, No. 1, Jan. 2007, pp.108 – 137

Park, H. G.; Barrett, A.; Baumann, E.; & Narasimhan, S.; Modular Architecture for Hybrid Diagnostic Reasoners, *Proceeding of SMC-IT'06*, pp. 277-284, Pasadena, 2006.

Schenato, L.; Sinopoli, B.; Franceschetti, M.; Poolla, K.; Jordan M. I. ; & Sastry, S. S. (2007). Foundations of Control and Estimation Over Lossy Networks, *Proceedings of IEEE*, Vol.95, No. 1, Jan. 2007, pp. 163-187.

Sinopoli, B.; Schenato, L.; Franceschetti, M.; Poolla, K.; Jordan, M. I.; & Sastry, S. S. (2004) Kalman Filtering with Intermittent Observations, *IEEE Transaction on Automatic Control*, Vol.49, No. 2, pp. 1453- 1464.

Sun, S.; Xie, L.; Xiao, W. ; & Soh, Y.C. (2008). Optimal linear estimation for systems with multiple packet dropouts, *Automatica*, Vol.44, No. 5, pp. 1333-1342, 2008.

Xiao, J. J. ; Cui, S. ; Luo, Z. Q. ; & Goldsmith, A. (2004). Joint estimation in sensor networks under energy constraints, *Proc. of IEEE 1st Conf. on Sensor and Ad Hoc Communications and Networks,* pp. 264-271, 2004.

Xiao, J. J. ; Cui, S. ; Luo, Z. Q. ; & Goldsmith, A. (2006). Power Scheduling of Universal Decentralized Estimation in Sensor Networks, *IEEE Transaction on Signal Processing*, Vol.54, No. 2, pp. 413-422.

Xiao, J. J. ; Cui, S. ; Luo, Z. Q. ; & Goldsmith, A. (2008). Linear coherent decentralized estimation, *IEEE Transaction on Signal Processing*, Vol. 56, No.2, pp. 757-770, 2008.

8

Wired and Wireless Reliable Real-Time Communication in Industrial Systems

Magnus Jonsson and Kristina Kunert
Halmstad University
Sweden

1. Introduction

In modern factory automation systems, data communication plays a vital role (Moyne and Tilbury, 2007). Different nodes like control systems, sensors and actuators can communicate over a wireless or wired industrial network. The data traffic generated is often scheduled for periodic transmission, where each single message or packet must arrive in time. For this real-time communication, methods have been developed to support communication services with a guaranteed throughput and delay bound for such periodic traffic, but merely under the assumption of error-free communication. However, the possibility for errors in the transmission still exists due to, e.g., noise or interference. A node receiving sensor values from a sensor in the system might then be forced to rely upon an older sensor value from the latest period, possibly leading to inaccuracies in control loops which can compromise the functioning of the system. In safety-critical systems, redundant networks or communication channels are frequently added to cope with errors, leading to more expensive systems. In this chapter, we will describe an alternative approach where erroneous data packets are retransmitted in a way that does not jeopardise any earlier stated real-time guarantees for ordinary transmissions. Using our framework, the reliability of real-time communication can be increased in a more cost-efficient way. We describe in this chapter an overview of our framework for reliable real-time communication, while details of our approach can be found in (Jonsson & Kunert, 2008; Jonsson & Kunert, 2008b; Jonsson & Kunert, 2009). In the light of the emerging use of wireless communication (Willig et al., 2005; Willig, 2008), the framework proves to be especially useful due to the high bit error rate inherent to the wireless medium. However, the framework is naturally also attractive for wired communication systems.

The rest of this chapter is organized as follows. In Section 2, several basic retransmission schemes are described in order to introduce basic concepts in retransmission protocols, while Section 3 summarizes some of the related work in the area of real-time communication and QoS (Quality of Service) provision by the usage of retransmissions. Section 4 introduces the concept of logical real-time channels used in our framework, followed by a description of our layered approach in Section 5. The retransmission scheme is explained in Section 6, while timing and scheduling analysis are briefly discussed in

Section 7. Section 8 is dedicated to results from our simulation studies before Section 9 concludes this chapter.

2. Basic retransmission schemes

Retransmission schemes are often referred to as ARQ (Automatic Repeat reQuest) protocols and can be divided into three categories (Lin et al., 1984; Tanenbaum, 2003):

- Idle RQ (also called stop-and-wait)
- Go-back-n
- Selective repeat

The latter two variants are of sliding window type (also called Continous RQ), while the first variant is a simpler kind of retransmission scheme. An Idle RQ protocol only allows for one packet at a time to be handled. After the packet has been transmitted, the transmitting node waits for an acknowledgement packet (ACK) from the other node, or for a retransmission timer to expire. A simplified finite state machine describing the function of the sending node using Idle RQ is shown in Fig. 1. Starting in the Idle state, a state transition to the "Wait for ACK" state is made when a request to send a packet is received from the layer above. The packet is sent immediately. If negative acknowledgement (NAK) is supported, retransmission can be explicitly requested by the other node. When an ACK packet is received by the transmitting node, the node returns to Idle state.

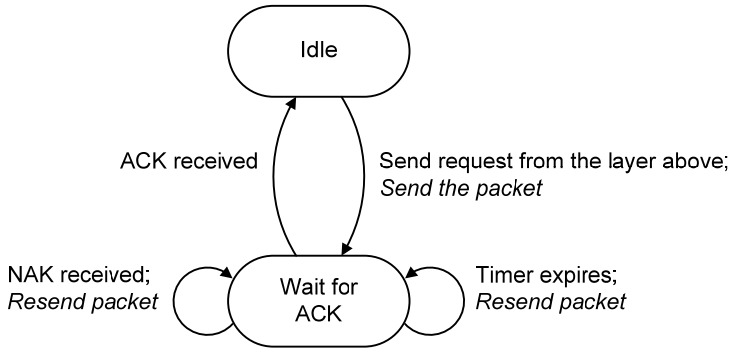


Fig. 1. Finite state machine describing the function of Idle RQ.

Idle RQ performs not very well in terms of link utilization at high bit-rates and/or long distances (large propagation delays), since a lot of the time then will be spent waiting for ACK packets. Continuous RQ protocols solve this problem by allowing several packets to be transmitted without having received ACK packets for the first packets already sent. The difference between go-back-n and selective repeat is in the retransmission strategy. When experiencing an erroneous packet transfer using go-back-n, the transmitter is restarting transmission starting with the packet after the last packet up to which all packets have been correctly transferred. In this way, even correctly transferred packets might be retransmitted. Using selective repeat instead, exclusively erroneous packets are retransmitted. In summary this means that selective repeat in most cases results in better link or network utilization, while go-back-n generally leads to simpler protocol implementations.

In order to make it possible to detect erroneous packets, a checksum or the like is always included in the packets. Furthermore, a sequence number is needed in both data packets, as an identifier of the data within the data stream, and ACK packets, to indicate which packet (or up to which packet) being acknowledged. Sequence numbers are also necessary in Idle RQ, since the possibility of duplicated data in case of a lost ACK packet might arise otherwise. Flow control, to avoid buffer overflow in the receiver, is solved in Idle RQ by only sending an ACK when the receiver is ready for the next transmission. Continuous RQ protocols use the concept of sliding window instead, where the size of the window corresponds to the receiver buffer size. This defines the width of the span of sequence numbers in which packets might be transmitted.
TCP (Transmission Control Protocol) (Postel, 1981) is probably the most well known ARQ protocol and is used in the transport layer to ensure reliable end-to-end transfer over the Internet. TCP is, however, not suitable when delay bounds and deterministic throughput guarantees need to be given. One obvious reason is the case when the same packet needs to be retransmitted indefinitely many times, and by that is delaying all other traffic. Additionally, the congestion control method included in TCP will decrease the packet transmission rate drastically whenever a retransmission is initiated. The problem with indefinite numbers of retransmissions, actually possible in all the three basic ARQ methods, can be solved by relaxing the demand on reliability just slightly and by using a truncated ARQ protocol instead, where the maximum number of retransmission attempts is limited. An example of a truncated ARQ method can be found in (Malkamäki & Leib, 2000). The usage of a truncated ARQ scheme, however, still demands real-time methods to be added, so that timely delivery of delay-sensitive traffic can be ensured.

3. Related work

The area of real-time communication is vast and well-studied, exactly as the research on the provisioning of QoS by retransmitting erroneous packets, as e.g. implemented by ARQ protocols. Although both topics have attracted interest in many years, a holistic view encompassing both approaches in order to *guarantee* timely treatment of real-time traffic and calculate the necessary delay bounds has remained relatively unstudied.
Amongst earlier work dealing with ARQ protocols for communication with real-time demands, solutions can be found which merely settle for a certain average performance, studying e.g. the average delay of delay-sensitive packets. In (Pejhan et al., 1996) it is shown, that using retransmissions is an effective way of error control for soft real-time traffic, i.e. traffic where occasionally missing the deadline is acceptable without serious implications on system performance. The article studies several different retransmission schemes used in connection with multicast protocols for real-time multimedia applications, which are typical examples of soft real-time applications. No hard real-time traffic is studied by the authors. Unfortunately, by the usage of statistical QoS parameters no deterministic calculations can be made and consequently no guarantee can be provided that no packet will miss its deadline.
The research presented in (Bilstrup et al., 2004) is suffering from a similar downside. The presented approach for a Bluetooth-based network provides a possibility of guaranteeing a certain degree of QoS. However, these guarantees are merely based on probabilistic calculations including average estimates of the quality of the wireless channel, and therefore

a deterministic guarantee of the end-to-end delay bound cannot be given. Additionally, the presented statistical calculations are based upon an Idle RQ approach, often resulting in poorer link utilization than Continuous RQ-based schemes like ours.
An approach explicitly assuming hard real-time traffic has been published by Butt (Butt, 2006). His approach is similar to ours in that it aims to decrease the bit error rate experienced by hard real-time traffic by retransmitting erroneous packets until their deadline has been reached. However, similarly to (Bilstrup et al., 2004), also this retransmission scheme is based upon Idle RQ, hazarding the consequences of decreasing system performance. By limiting the suggested solution to a packet-level description, without developing an analysis of the queuing delay or providing details on the scheduling analysis, the provision of end-to-end delay-bound guarantees is not possible. In contrast, our approach with end-to-end real-time channels makes it possible to include an end-to-end delay bound analysis (including a queuing delay analysis) and by that provide guarantees of bounded end-to-end delay.
A way of providing end-to-end real-time guarantees including retransmissions can be found in (Giancola et al., 2002). The disadvantage compared to our approach is that the network capacity is analysed by the means of flow analysis. Recent work by Fan (Fan et al., 2009) has shown that flow analysis is not exploiting the available network capacity as fully as real-time scheduling does. Our framework is combining this kind of real-time scheduling analysis with the principle of retransmissions. Additionally, the exact timing details necessary to support hard real-time communication in industrial real-time systems, and used in our detailed analysis, are not included in the approach in (Giancola et al., 2002).
A related interesting solution to provide deadline guarantees for hard real-time traffic is deadline dependent coding, which on the bit level is combining error correcting codes and ARQ (Uhlemann et al., 2000; Uhlemann & Rasmussen, 2005). While this approach is presented as a major requirement for reliable wireless real-time communication, the articles treat merely point-to-point links, and no continuation towards a complete real-time scheduling analysis framework has been targeted for.

4. Logical real-time channels

The concept of logical real-time channels (RT channels), also referred to as RTVCs (Real-Time Virtual Channels), was introduced in (Ferrari & Verma, 1990). A RT channel is an abstraction of a traffic flow over a link or a network, where resources have been allocated to guarantee a certain minimum throughput and a bounded end-to-end delay. The basic parameters of a network layer RT channel i, following the notation in (Jonsson & Kunert, 2008b) (we follow this notation in the whole chapter), are the period (or minimum inter-arrival time), $P_{N,i}$, the maximum message length in bits each period, $L_{N,i}$, and the end-to-end delay bound, $D_{N,i}$, where N denotes the network layer. A RT channel can then be defined as $\tau_{N,i} = \{P_{N,i}, L_{N,i}, D_{N,i}\}$, or, in the case of a network, as $\tau_{N,i} = \{m_{s,i}, m_{d,i}, P_{N,i}, L_{N,i}, D_{N,i}\}$, where $m_{s,i}$ and $m_{d,i}$, denote the source node and the destination node, respectively. The source nodes are bound to behave according to the traffic specifications and must not violate them by e.g. sending more often.
To be able to state a deadline guarantee for a new RT channel, the worst-case delay must be analyzed for all existing RT channels plus the new one. Only if the delay bounds for this whole set of RT channels can be met, the new channel is accepted. When this procedure is

carried out online (during run-time) it is called admission control. The admission control system needs a real-time scheduling analysis to analyze the worst-case delays, which in turn must rely upon a deterministic behaviour of the network and/or the communication equipment. A shared-medium network (for example a bus or a ring network) must have a MAC (Medium Access Control) protocol with a deterministic behaviour (Malcolm & Zhao, 1995). As an example, the CSMA/CD (Carrier Sense Multiple Access Collision Detect) method used in shared-medium Ethernet is not deterministic since indefinite numbers of collisions can occur, caused by the randomness involved when resolving collisions. When two nodes send at the same time, detect a collision and backoff before retrying to send their data, they can randomly generate the same backoff time and thereafter experience a new collision.

For a point-to-point link, or a switched network made up of switches and point-to-point links, a deterministic queuing principle (or service discipline) must be used (Zhang, 1995). Even though FCFS (First Come First Served) is a deterministic queuing principle (Fan, 2005), there are queuing principles specially developed for real-time communication. Two examples of such are delay-EDD(Earliest Due Date) (Ferrari & Verma, 1990) and jitter-EDD (Verma et al., 1991). Both delay-EDD and jitter-EDD rely upon EDF (Earliest Deadline First) (Liu & Layland, 1973) scheduling, where the packets are sorted according to their deadlines. Regarding industrial communication systems appropriate for factory automation systems, the use of EDF packet scheduling in switched Ethernet networks to obtain real-time support has been proposed (Hoang et al., 2002; Hoang et al., 2002b).

5. A layered approach to reliable real-time communication

Our proposed retransmission scheme resides in the transport layer (see Fig. 2). It relies on RT channels supported by the network layer, but which are not offering any reliability in terms of retransmissions. The transport layer can, however, rely on timely delivery of packets by the network layer.

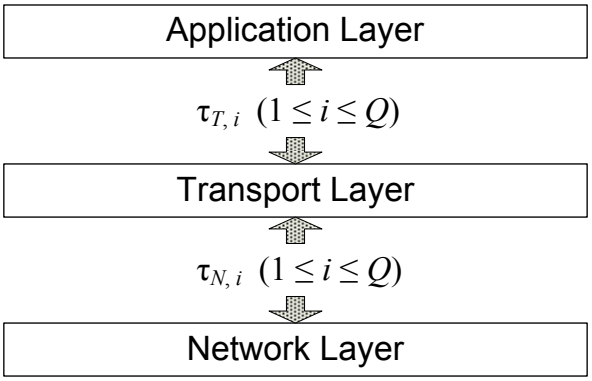


Fig. 2. The proposed retransmission scheme resides in the transport layer, relying on RT channels in the network layer and giving the service of reliable RT channels to the application layer.

The transport layer offers the service of reliable RT channels to the application layer. We define a reliable RT channel as a logical channel over which ordinary transmissions are always guaranteed to arrive in time, while retransmissions are made as long as no delay bounds of the ordinary transmissions are adventured. All parameters of a transport layer RT channel (reliable RT channel), $\tau_{T,i} = \{m_{s,i}, m_{d,i}, P_{T,i}, L_{T,i}, D_{T,i}\}$, where T stands for transport layer, can be mapped directly onto the corresponding parameters of a network layer RT channel, except for the delay bound parameter. We elaborate further on the delay bound parameter in the next section.

6. Our retransmission scheme

For the rest of the chapter, we will assume a point-to-point link over which we implement our retransmission scheme. Moreover, we assume that the packet queue in the transmitting node is an EDF queue, i.e. the packets are sorted according to their deadlines. To accomplish the possibility of having timing constraints on both ordinary transmissions and possible retransmissions, we divide the transport layer delay bound, $D_{T,i}$, for each RT channel into $T_{D\_ord,i}$ and $T_{D\_retr,i}$. The value of $T_{D\_retr,i}$ is set to the same value, D_{retr}, for all RT channels, where D_{retr} is a system parameter defining the time span allocated for retransmissions. An example is shown in Fig. 3, where a message consisting of three packets (to fit all $L_{T,i}$ bits) is transmitted. Packet three is not transferred correctly and is therefore retransmitted. Even the first retransmission attempt is incorrectly transferred and a second retransmission attempt is therefore initiated.

The second retransmission attempt is not acknowledged since no more retransmission attempts are allowed anyhow. This can of course be changed if the sending application really needs to know whether the transmission was successful or not. The ordinary packet transmissions follow the specification of the RT channel, but where the delay bound of the network layer RT channel is set to $D_{N,i} = T_{D\_ord,i}$. Network resources for ordinary transmissions are, in other words, allocated by the corresponding RT channel. Resources for retransmissions are, however, allocated through the use of special retransmission RT channels (using network layer RT channels) shared by all transport layer RT channels needing retransmissions. All retransmission RT channels will have their delay bounds set to $D_{retr,i} = D_{retr}$ (if several retransmission attempts are supported in the scope of D_{retr}, this is considered in the timing analysis), while $L_{retr,i}$ is set to the maximum packet length in the system. $P_{retr,i}$ is a system parameter indicating how often a retransmission RT channel can be used for retransmission of a packet belonging to an arbitrary ordinary RT channel.

The retransmission scheme can be seen as a truncated ARQ protocol, but where the timings of both ordinary transmissions and possible retransmissions are strictly regulated. Our retransmission scheme has not the drawbacks of Idle RQ, but allows for several consecutive transmissions without having received the ACK packets for the first transmitted packets yet. Normal flow control is not needed since RT channels are only admitted if all requirements can be guaranteed by the analysis discussed in the next section. Since the delays are bounded, it is possible to calculate the worst-case buffer population. Thereby, we can also ensure that we have enough buffer space and not admit RT channels otherwise.

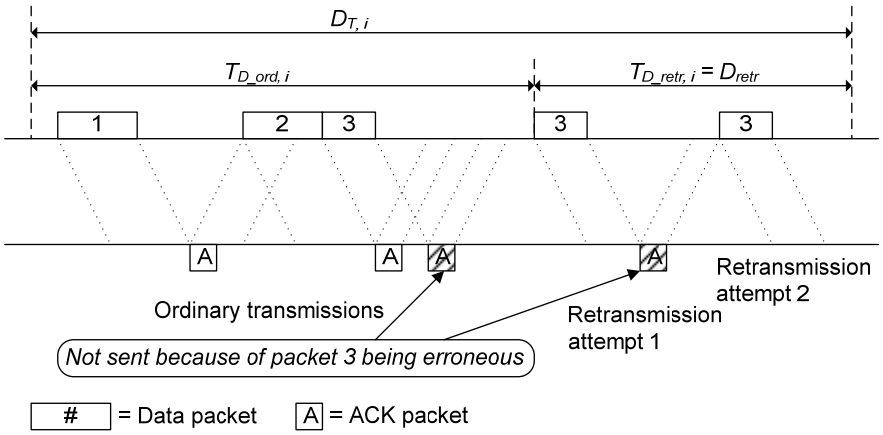


Fig. 3. The example shows how the end-to-end delay bound is divided into one bound for ordinary transmissions and one bound for a limited number of retransmission attempts.

7. Timing and scheduling analysis

The packets of a message belonging to a RT channel need to compete with packets belonging to other RT channels. As an example, the reason why packet 2 is not sent immediately after packet 1 in Fig. 3 might be that packets belonging to other RT channels, and with earlier deadlines, just arrived to the queue. To be able to analyse whether we can admit a new RT channel or not, we need to isolate the maximum message queuing delay (deadline) for each RT channel and perform a scheduling analysis for the EDF queue to see whether all such deadlines can be met. To do this we identify all other delays (constants or worst-case delays) and subtract them from the end-to-end delay bound. Examples of such delays are the propagation delay, blocking delay caused by a packet with a later deadline having to finish transmission, timing margins etc. For the retransmission RT channels, we need also to consider the maximum number of retransmission attempts per packet, $N_{attempt}$, to be supported. Detailed equations for those calculations are given in (Jonsson & Kunert, 2008b), while an extended analysis is given in (Jonsson & Kunert, 2009).
The scheduling analysis consists of two tests to be performed. The first test is to check that the aggregated utilization of all RT channels is not greater than 100%. The equation for calculating the utilization, quoted from (Jonsson & Kunert, 2008b), is:

$$U = \sum_{i=1}^{Q} \left(\frac{T_{x\_tot,i}}{P_{T,i}} \right) + \sum_{i=1}^{M} \left(\frac{T_{x\_retr,i}}{P_{retr,i}} \right) \quad (1)$$

where Q is the number of RT channels, M is the number of retransmission channels, $T_{x\_tot,i}$ is the total transmission time for all packets of a message including headers, while $T_{x\_retr,i}$ is the corresponding transmission time for a retransmission packet. In case of a successful first

test, the second test is necessary. This second test comprises the following equation quoted from (Jonsson & Kunert, 2008b):

$$h(t) = \sum_{\substack{i\in[1,Q],\\ T_{d\_ord,i}\le t}} \left(1 + \left\lfloor \frac{t - T_{d\_ord,i}}{P_{T,i}} \right\rfloor\right) \cdot T_{x\_tot,i} + \sum_{\substack{i\in[1,M],\\ T_{d\_retr,i}\le t}} \left(1 + \left\lfloor \frac{t - T_{d\_retr,i}}{P_{retr,i}} \right\rfloor\right) \cdot T_{x\_retr,i} \quad (2)$$

where $T_{d\_ord,i}$ and $T_{d\_retr,i}$ are the scheduling (queuing) deadlines for ordinary RT channels and retransmission RT channels, respectively. The $h(t)$ function is a workload function adding together the transmission times of all message instances (from different periods) of all channels for those messages which have deadline instances less or equal to t. The constraint of the second test is that $h(t)$ must be less or equal to t for any value of t. The workload function has its origin from work done by Spuri (Spuri, 1996; Stankovic et al., 1998), later adapted for communication systems (Hoang & Jonsson, 2003; Hoang & Jonsson, 2003b). Details on how the test can be reduced to a discrete number of values of t to be checked are found in (Jonsson & Kunert, 2008b; Jonsson & Kunert, 2009; Stankovic et al., 1998). If not both the utilization test and the workload test are completed successfully, the new RT channel must be rejected.

8. Simulation analysis

With the aim of demonstrating the potential of our ARQ scheme we have implemented a MATLAB simulator and conducted two types of simulations in order to evaluate two different performance parameters of the system which are influenced by our retransmission scheme. The first type of simulation studies the utilization of the link by ordinary transmissions of hard real-time traffic (without including any retransmission packets). In order to calculate the utilization parameter, the simulation implements a schedulability analysis on the RT channel level, where an admission decision for generated RT channel requests is made. If a channel's delay bound can be guaranteed, while not violating already stated guarantees, the channel will be admitted; otherwise it will be denied access to the bandwidth. We will show and evaluate both the results for the case when using no retransmissions and for the case when our retransmission scheme is used. In the second type of simulation we observe the improvement of the message error rate which can be obtained by the usage of our retransmission scheme. These observations are made on the packet level, and in order to be able to relate these results to the utilization measurements from the first simulation type, the packets studied are those transmitted on the RT channels generated in this previous simulation.

We simulated both parameters typical for wired networks and those typical for wireless ones. Assumptions in common for both networks were full-duplex links and the same bit rate in each direction for reasons of simplicity. The maximum packet length was assumed to

be 1000 bits in all cases, and the propagation delay in one direction experienced by the packets is assumed to be 1 μs. Each acknowledgement packet has a length of 100 bits, including a sufficient amount of redundant bits for error correction, resulting in a negligible error rate experienced by those packets. In order to better see the potential of our retransmission scheme, an implementation of piggyback acknowledgements was chosen as this generally leads to more efficient bandwidth utilization. The bit rates were set to be 100 Mbit/s and 10 Mbit/s for the wired and wireless links, respectively. The simulation study comprises three different cases which are presented in the following, and the traffic scenarios belonging to them are summarized in Tables 1, 2, and 3. Each traffic scenario defines four traffic classes, from which each RT channel is randomized (even distribution), and the parameters of the retransmission channel. Additionally, the number of retransmission attempts and the number of retransmission channels are indicated. In all figures the results obtained for a link without retransmissions are shown as a dashed line, while the results when using our retransmission scheme are shown by an unbroken line. In order to arrive at statistically reliable numbers, each point on the curves is, for each x-value, the average of 1000-1100 simulation runs over the length of 25-120 hyperperiods. (One hyperperiod starts at the point in time when all periods start simultaneously until they do so again.)

Case 1 - Wireless network			
Bit rate = 10 Mbit/s BER = 10^{-4} Number of retransmission attempts: 3 Number of retransmission channels: 10			
Traffic classes	*Period*	*Deadline*	*Message length*
1	20 ms	20 ms	4000 bits
2	40 ms	40 ms	4000 bits
3	80 ms	80 ms	4000 bits
4	160 ms	160 ms	4000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	10 ms	6 ms	1000 bits

Table 1. The specification parameters for the first simulated case of a wireless network.

The first case (see Table 1) evaluates the scenario of a wireless network (10 Mbit/s) with a bit error rate of 10^{-4}. All traffic classes have relatively long periods and deadlines, and the message length was set to 4000 bits, corresponding to four maximum sized packets. The ten retransmission channels have a period of 10 ms, a deadline of 6 ms and for each erroneous packet a maximum of three retransmissions was attempted. The simulation results illustrating the link utilization and the experienced message error rate are shown in Fig. 4 and Fig. 5, respectively. The results show a clear reduction of the message error rate by three orders of magnitude, while only reducing the utilization of the link by ordinary RT channels by about 12 percentage units at saturation.

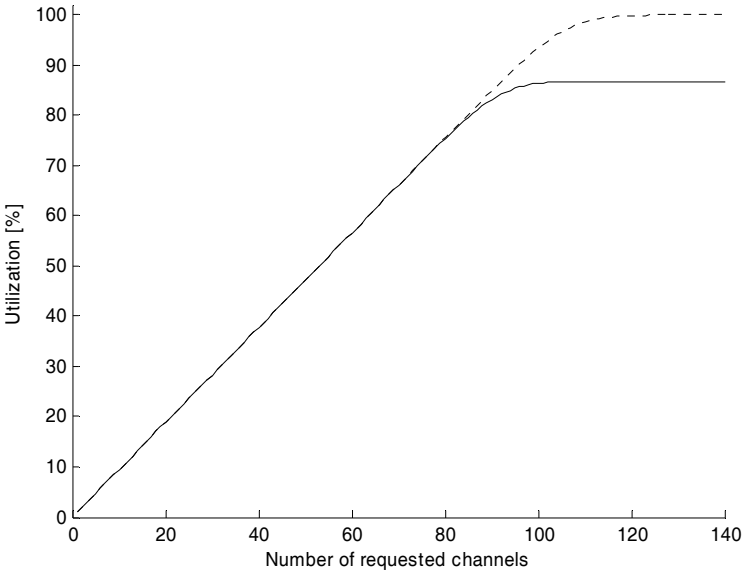


Fig. 4. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

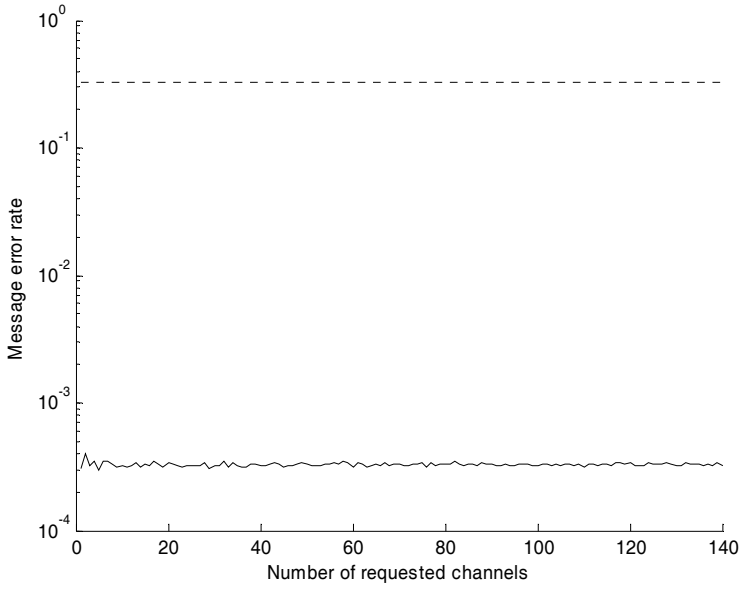


Fig. 5. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

In case 2 (see Table 2) a second wireless scenario was simulated. While still keeping the bit rate at 10 Mbit/s, the bit error rate was lowered to 10^{-5} in order to see if we could improve

this value even further by the help of our retransmission scheme. The periods and deadlines of the four different traffic classes were shortened, increasing the demand on the link, and additionally the deadline of the retransmission channel was shortened, making the scheduling of the retransmission channels more difficult. For a system with four retransmission channels and two retransmission attempts per packet, the results are shown in Fig. 6 and Fig. 7. The message error rate was improved by about four orders of magnitude, while the utilization penalty was no higher than approximately 5 percentage units. This combination of system and simulation parameters outperformed the ones from case 1 easily.

Case 2 - Wireless network			
Bit rate = 10 Mbit/s BER = 10^{-5} Number of retransmission attempts: 2 Number of retransmission channels: 4			
Traffic classes	*Period*	*Deadline*	*Message length*
1	10 ms	10 ms	4000 bits
2	20 ms	20 ms	4000 bits
3	40 ms	40 ms	4000 bits
4	80 ms	80 ms	4000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	10 ms	2 ms	1000 bits

Table 2. The specification parameters for the second simulated case of a wireless network.

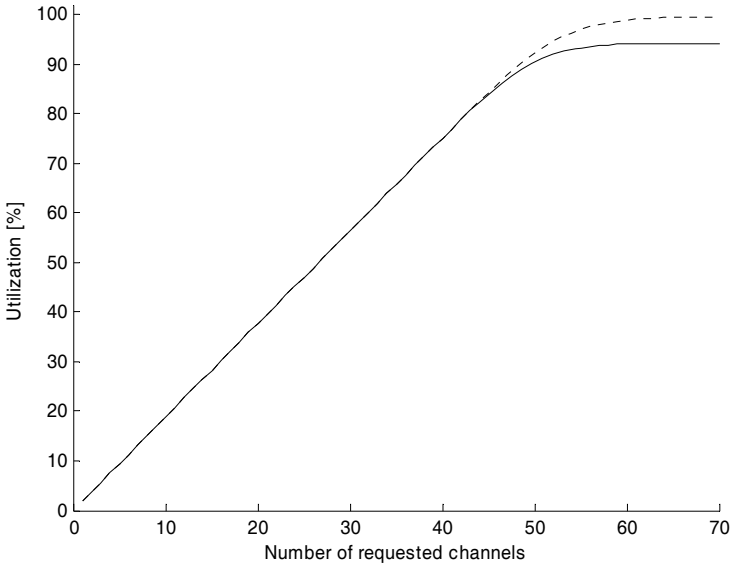


Fig. 6. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

Case 3 - Wired network			
Bit rate = 100 Mbit/s BER = 10^{-6} Number of retransmission attempts: 2 Number of retransmission channels: 3			
Traffic classes	*Period*	*Deadline*	*Message length*
1	1 ms	1 ms	20000 bits
2	2 ms	2 ms	20000 bits
3	4 ms	4 ms	20000 bits
4	8 ms	8 ms	20000 bits
Retransmission channel	*Period*	*Deadline*	*Packet length*
	1 ms	200 µs	1000 bits

Table 3. The specification parameters for the simulated case of a wired network.

In the last simulation setup, case 3 (see Table 3), the scenario of a wired network was implemented. While the bit rate was increased to 100 Mbit/s, the bit error rate was lowered to 10^{-6}. Both period and deadline of the traffic classes were shortened to values between 1 ms and 8 ms, and the packets grew to a size of 20000 bits.

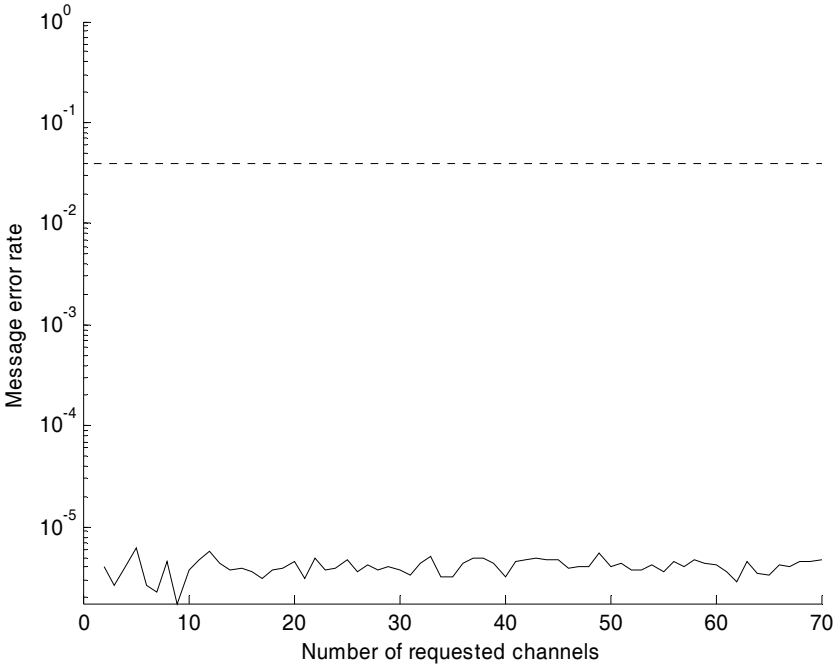


Fig. 7. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

Period and deadline of the three retransmission channels, supporting two retransmission attempts per packet, were also shortened substantially. The results obtained from this simulation setup were as positive as can be seen in the figures. Here, the utilization penalty of about 5 percentage units was able to improve the message error rate experienced by the packets by approximately four orders of magnitude (see Fig. 8 and Fig. 9).

9. Conclusions

Industrial communication systems would benefit immensely if both their demands on reliability, in terms of correctly transferred information, and real-time constraints could be considered, especially thinking about the emerging use of wireless communication technology. In this chapter, we have shown how this could be made possible by using a retransmission scheme where retransmissions are only allowed if they do not put at risk any delay bounds of ordinary transmissions by making sure they are not violated. Timing and scheduling analyses ensure the timely delivery of both ordinary transmissions and retransmissions. By the help of simulations, we have shown that a reduction of the message error rate with several orders of magnitude is possible, while just sacrificing a small fraction of the bandwidth. Future work includes investigating further how to enhance the framework for complex networks instead of a point-to-point link.

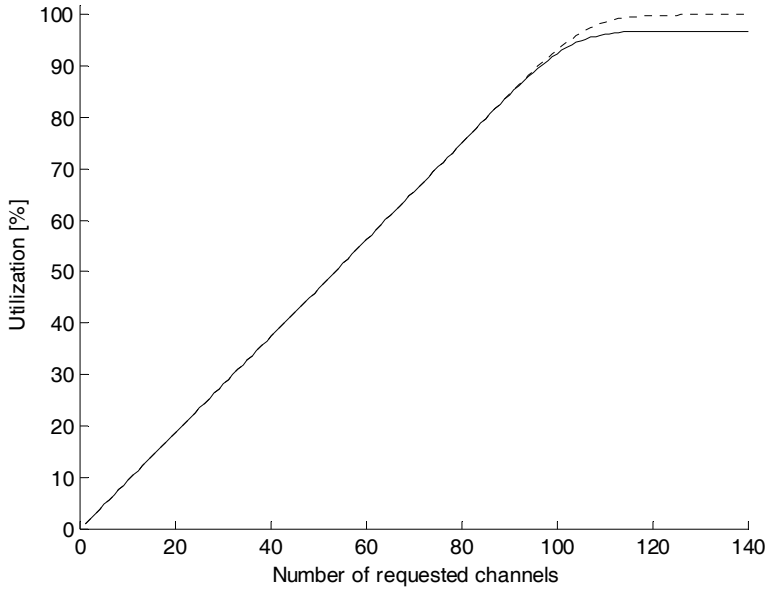


Fig. 8. Simulation results showing the link utilization both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

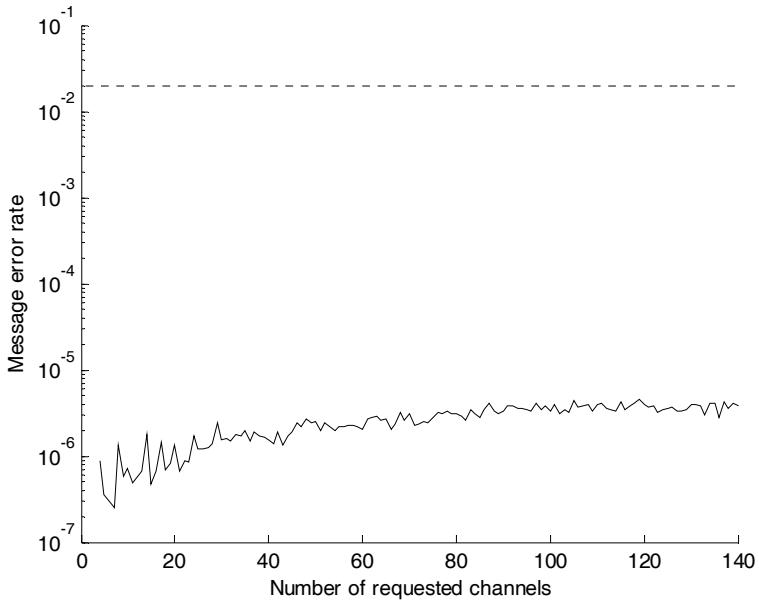


Fig. 9. Simulation results showing the experienced message error rate both using no retransmission scheme (dashed line) and using our retransmission scheme (unbroken line).

10. References

Bilstrup, U.; Sjöberg, K.; Svensson, B. & Wiberg, P.-A. (2004). A fault tolerance test enabling QoS in a Bluetooth piconet, *Proceedings of the 3rd International Workshop on Real-Time Networks (RTN 2004)*, June 2004, pp. 33-36.

Butt, M. M. (2006). Provision of guaranteed QoS with hybrid automatic repeat request in interleave division multiple access systems, *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems (ICCS 2006)*, Oct. 2006, pp. 1-5.

Fan, X. & Jonsson, M. (2005). Guaranteed real-time services over standard switched Ethernet, *Proceedings of the 30th Annual IEEE Conference on Local Computer Networks (LCN 2005)*, Sydney, Australia, Nov. 2005.

Fan, X.; Jonsson, M. & Jonsson, J. (2009). Guaranteed real-time communication in packet-switched networks with FCFS queuing, *Computer Networks*, Vol. 53, No. 3, Feb. 2009, pp. 400-417.

Ferrari, D. & Verma, D. C. (1990). A scheme for real-time channel establishment in wide-area networks, *IEEE Journal of Selected Areas in Communications*, Vol. 8, No. 3, Apr. 1990, pp. 368-379.

Giancola, G.; Falco, S. & Di Benedetto, M. G. (2002). A novel approach to error protection in medium access control design, *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Networks (MWCN 2002)*, Sept. 2002, pp. 337-341.

Hoang, H.; Jonsson, M.; Hagström, U. & Kallerdahl, A. (2002). Switched real-time Ethernet with earliest deadline first scheduling - protocols and traffic handling, *Proceedings of the Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2002) in conjunction with the International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Fort Lauderdale, FL, USA, April 2002.

Hoang, H.; Jonsson, M.; Kallerdahl, A. & Hagström, U. (2002b). Switched real-time Ethernet with earliest deadline first scheduling - protocols and traffic handling, *Parallel and Distributed Computing Practices*, Vol. 5, No. 1, 2002.

Hoang, H. & Jonsson, M. (2003). Switched real-time Ethernet in industrial applications - asymmetric deadline partitioning scheme," *Proceedings of the 2nd International Workshop on Real-Time LANs in the Internet Age (RTLIA 2003) in conjunction with the 15th Euromicro Conference on Real-Time Systems*, Porto, Portugal, July 2003.

Hoang, H. & Jonsson, M. (2003b). Switched real-time Ethernet in industrial applications - deadline partitioning, *Proceedings of the Asia-Pacific Conference on Communications (APCC 2003)*, Penang, Malaysia, Sept. 2003, pp. 76-81.

Jonsson, M. & Kunert, K. (2008). Reliable hard real-time communication in industrial and embedded systems, *Proceedings of the 3rd IEEE International Symposium on Industrial Embedded Systems (SIES)*, Montpellier, France, June 2008.

Jonsson, M. & Kunert, K. (2008b). Meeting reliability and real-time demands in wireless industrial communication, *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, Hamburg, Germany, Sept. 2008.

Jonsson, M. & Kunert, K. (2009). Towards reliable wireless industrial communication with real-time guarantees, *IEEE Transactions on Industrial Informatics*, Vol. 5, No. 4, Nov. 2009, pp. 429-442.

Lin, S.; Costello, D. & Miller, M. (1984). Automatic-repeat-request error-control schemes, *IEEE Communications Magazine*, Vol. 22, No. 12, Dec. 1984, pp. 5-17.

Liu, C. L. & Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment, *Journal of the ACM*, Vol. 20, No. 1, pp. 46-61, Jan. 1973.

Malkamäki, E. & Leib, H. (2000). Performance of truncated type-II hybrid ARQ schemes with noisy feedback over block fading channels, *IEEE Transactions on Communications*, Vol. 48, No. 9, Sept. 2000, pp. 1477-1487.

Malcolm, N. & Zhao, W. (1995). Hard real-time communication in multiple-access networks, *Real-Time Systems*, Vol. 8, No. 1, 1995, pp. 35-77.

Moyne, J. R. & Tilbury, D. M. (2007). The emergence of industrial control networks for manufacturing control, diagnostics, and safety data, *Proceedings of the IEEE*, Vol. 95, No. 1, Jan. 2007, pp. 29-47.

Pejhan, S.; Schwartz, M. & Anastassiou, D. (1996). Error control using retransmission schemes in multicast transport protocols for real-time media, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 3, June 1996, pp. 413-427.

Postel, J. (1981). Transmission control protocol - darpa internet program protocol specification, *IETF RFC 793*, Sept. 1981.

Spuri, M. (1996). Analysis of Deadline Scheduled Real-Time Systems, *Technical Report RR No. 2772, INRIA*, France, 1996.

Stankovic, J. A.; Spuri, M.; Ramamritham, K. & Buttazzo, G. C. (1998). *Deadline scheduling for real-time systems - EDF and related algorithms*, Kluwer Academic Publishers, Boston, MA, USA, 1998.

Tanenbaum, A. S. (2003). *Computer Networks*. Fourth edition, Prentice Hall, Upper Saddle River, NJ, USA, 2003, ISBN 0-13-066102-3.

Uhlemann, E.; Wiberg, P.-A.; Aulin, T. M. & Rasmussen, L. R. (2000). Deadline dependent coding - a framework for wireless real-time communication, *Proceedings of the 7th International Conference on Real-Time Computing Systems and Applications (RTCSA 2000)*, Cheju Island, South Korea, Dec. 2000, pp. 135-142.

Uhlemann E. and Rasmussen, L. K. (2005). Incremental redundancy deadline dependent coding for efficient wireless real-time communications, *Proceedings of the 10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, Catania, Italy, Sept. 2005, Vol. 2, pp. 417-424.

Verma, D ; Zhang, H. & Ferrari, D. (1991). Guaranteeing delay jitter bounds in packet switching networks, *Proceedings of TriComm'91*, Chapel Hill, NC, USA, Apr. 1991, pp. 35-46.

Willig, A.; Matheus, K. & Wolisz, A. (2005). Wireless technology in industrial networks, *Proceedings of the IEEE*, Vol. 93, No. 6, June 2005, pp. 1130-1151.

Willig, A. (2008). Recent and emerging topics in wireless industrial communications: a selection, *IEEE Transactions on Industrial Informatics*, Vol. 4, No. 2, May 2008, pp. 102-124.

Zhang, H. (1995). Service disciplines for guaranteed performance in packet-switching networks, *Proceedings of the IEEE*, Vol. 83, No. 10, Oct. 1995, pp. 1374-1396.

9

A perspective on the IEEE 802.11e Protocol for the Factory Floor

Lucia Lo Bello, Emanuele Toscano
and Salvatore Vittorio
University of Catania
Italy

1. Introduction

In recent years, wireless technologies have been identified as a very attractive option for industrial and factory automation. Some of the benefits of these technologies are mobility support, reduced cabling and installation costs, reduced danger of breaking cables and less hassle with connectors. Typical application classes where wireless technologies might be used are closed-loop control involving mobile subsystems, coordination among mobile robots or autonomous vehicles, health monitoring of machines, tracking of parts and many more. Moreover, wireless technologies may be useful in factory automation systems where there is the need to dynamically connect new components to an already deployed wired communication system that cannot be reached (easily and/or reliably) with a cable.
On the other hand, wireless solutions and products available today on the market are considered unsuitable for implementing distributed control applications and systems, in particular when real-time and/or reliability constraints are key issues. In fact, wireless networks are influenced by a number of factors that have to be carefully analysed before using them at the device level of factory automation systems. As an example, wireless transmissions are quite sensitive to electro-magnetic noise, obstacles, multi-path fading, etc. Industrial environments are generally very hostile and multiple types of noise may cause transmission errors, that can significantly affect the robustness. Moreover, as the wireless channel is non-deterministic and time-varying, the classical deterministic performance measures such as the worst-case transmission times should be replaced by probabilistic assessments.
IEEE 802.11 is the most widely used wireless technology and several research works exist that study its suitability for the industrial environment and try to enhance the support provided to industrial traffic. Moreover, the interest of the industrial community regarding this protocol has noticeably grown since the IEEE 802.11e amendment has been incorporated into the published IEEE 802.11-2007 standard. In fact, such an amendment uses the same IEEE 802.11 physical layer, but provides two novel channel access mechanisms that support Quality of Service (QoS), namely, EDCA and HCCA. This Chapter aims at giving a summary of the many works in literature, to provide a broad overview on the current results regarding the IEEE 802.11e protocol in industrial communication, with a special

focus on the support provided to real-time traffic in factory automation. While Section 2 presents a summary of the IEEE 802.11e protocol, Sections 3 and 4 give an overview of analytic performance models and of the most relevant admission control algorithms currently in literature, respectively. In Section 5 some improved scheduling algorithms for HCCA are presented. In addition, as analytic models prove that the EDCA performance strongly depends on the protocol parameters, some mechanisms that control those parameters to improve the performance of the EDCA traffic will be discussed as well. Section 6 discusses several case studies of using 802.11e for factory automation.

2. The IEEE 802.11 standard

The 802.11 standard defines the Media Access Control (MAC) and Physical (PHY) layers specifications for wireless LANs. Three different physical layer specifications were originally defined, namely, Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS) and Infrared (IR), with a nominal data rate up to 2 Mbps. Today such PHY specifications are obsolete, while three different specifications are currently in use, defined in 802.11a (IEEE 802.11a, 1999), 802.11b (IEEE 802.11b, 1999) and 802.11g (IEEE 802.11g, 2003). While 802.11a operates in the 5 GHz band supporting a data rate up to 54 Mbps, both the 802.11b and 802.11g physical layers operate in the license-free 2.4 GHz ISM (Industrial, Scientific and Medical) band, supporting data rates up to 11 Mbps and 54 Mbps, respectively.

The IEEE 802.11 protocol defines two different WLAN architectures, i.e., Basic Service Set (BSS) and Independent Basic Service Set (IBSS). In a Basic Service Set, some wireless stations (STAs) are associated to an Access Point (AP) and all the communications take place through the AP. Conversely, in an Independent Basic Service Set, STAs can communicate directly to each other, provided that they are within each other's transmission range. This kind of architecture allows the formation of a wireless ad-hoc network in the absence of any network infrastructure.

At the MAC layer, the IEEE 802.11 protocol defines two different access mechanisms, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The former access mechanism is mandatory and provides distributed channel access based on a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). A node willing to transmit a data frame performs the carrier sensing to determine whether the channel is idle or busy. After sensing an idle channel for a duration equal to a Distributed Interframe Space (DIFS) time, the node can transmit. If a collision occurs, the node starts a back-off procedure to avoid other collisions. The back-off procedure starts a random timer, whose value is uniformly chosen in the [0,CW] interval, that is decremented at each slot time after the channel is sensed idle for a DIFS time. If the medium becomes busy during this back-off process, the station pauses its back-off timer until the channel is free again. The next transmission attempt will occur when the back-off timer expires. In particular, at the first transmission attempt, the CW is set to the minimum Contention Window size, CWmin. After each unsuccessful transmission, the CW is increased, using the equation

$$CW_{new} = 2 \times (CW_{old} + 1) - 1, \quad (1)$$

until it reaches the maximum Contention Window size, CWmax. Conversely, after each successful transmission the CW value is reset to CWmin. The PCF mechanism is optional and divides time in superframes, providing centralized channel access through polling within a contention-free period (CFP) of the superframe. While DCF is designed to support only best effort traffic, PCF may be used for time bounded services. However, in PCF the AP has to contend for channel access at the beginning of each CFP. This leads to unpredictable jitters in the start time of the CFP, that make it difficult to support periodic time-constrained traffic that is typical of industrial applications. Moreover, in PCF the duration of transmissions by the polled station is unknown. To provide a better support for time constrained applications some extensions of the basic MAC protocol are provided by the IEEE 802.11e specifications.

2.1 The IEEE 802.11e amendment

The 802.11e standard specification (IEEE 802.11, 2007) was published by the IEEE Task Group E to provide mechanisms to enhance the current 802.11 MAC, so as to provide the support of Quality of Service requirements. In IEEE 802.11e, the DCF remains the principal access method when any QoS support is not needed while, as in IEEE 802.11, the PCF is optional. However, an additional coordination function is introduced, called a Hybrid Coordination Function (HCF), which in turn introduces two additional MAC modes, i.e., Enhanced Distributed Channel Access (EDCA), and HCF Controlled Channel Access (HCCA), the former being an enhancement of Distributed Coordination Function (DCF), while the latter being an enhancement of the Point Coordination Function (PCF).

The IEEE 802.11e protocol defines a Contention Period (CP) and Contention Free Period (CFP), during which the EDCA and the HCCA operate, respectively. These two periods appear alternately, and their durations are configurable according to the application requirements. While the presence of the CFP is optional, as it is only needed when HCCA is used, a minimum CP is always needed, as EDCA is also used to request resources for HCCA traffic. A CFP is delimited by a Beacon frame at the start and by a CF-End frame at the end.

The frame exchange mechanism of IEEE 802.11e is similar to that of IEEE 802.11, but some novel features have been introduced, e.g., the possibility to attach either an ACK or a poll for a specific QoS Station (QSTA) to a previously received frame (piggybacking). In this way the number of frames to be exchanged is reduced and the overhead is decreased. In IEEE 802.11e, the time interval during which a particular non-AP station has the right to access the channel is called transmission opportunity (TXOP). A TXOP is characterized by a starting time and a maximum duration. Two different types of TXOPs exist: EDCA TXOP and HCCA TXOP. The former is obtained by means of the EDCA during the CP, while the latter is assigned by the Hybrid Coordinator (HC) during either the CFP or a special part of the CP, called Controlled Access Phase (CAP).

A minimum interval between the transmission of consecutive frames is defined, namely the Inter Frame Space (IFS). While different IFS types exist in the IEEE 802.11 protocol to prioritize different kinds of frames, in IEEE 802.11e IFSs are also used to prioritize channel access of data frames belonging to different service categories, by introducing an Arbitration IFS (AIFS).

2.2 HCF Controlled Channel Access (HCCA)

Similarly to PCF, HCCA provides polled access to the wireless medium. But unlike PCF, QoS polling can take place during CP and scheduling of packets is based on the traffic requirements. In particular, when a station (QSTA) wants to associate with a certain Basic Service Set (BSS), it specifies its requirements during the so-called TSPEC negotiation. The QoS stations (QSTAs) send QoS reservation requests using a special QoS management frame, called Traffic Specification (TSPEC). The TSPEC frame contains the set of parameters that define the QoS characteristics (such as mean data rate or delay bound) of the particular traffic they are willing to transmit. An Admission Control Unit (ACU) in the QoS Access Point (QAP) is in charge of admitting or rejecting a new stream based on both the resource available and scheduling values.

The time between two consecutive beacon frames is called a super-frame. It is divided into a contention free period (CFP) and a contention period (CP) as shown in Fig. 1.

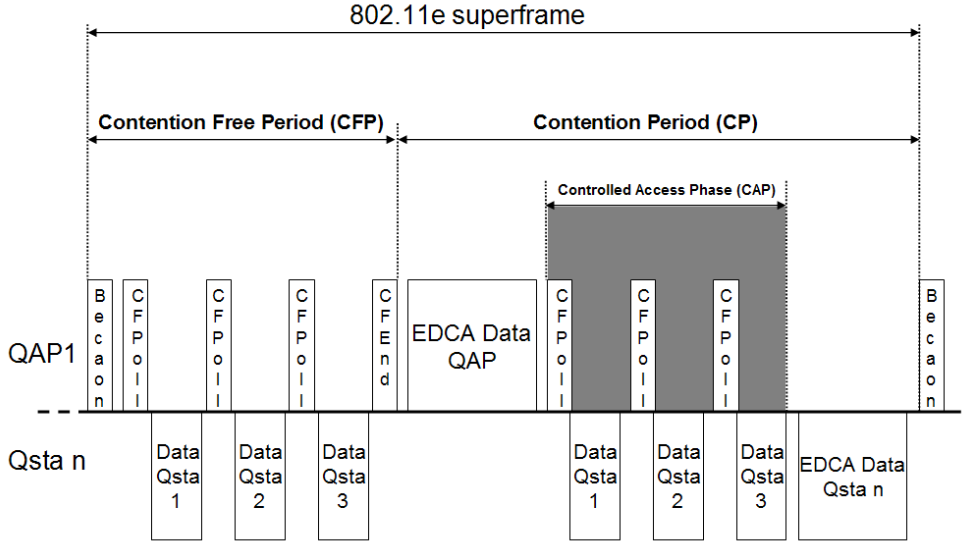


Fig. 1. 802.11e Super-frame

During CFP, the hybrid coordinator (HC) controls the access to the channel by polling its associated stations through CF-Poll messages, according to their QoS requirements. The HC will typically reside within an 802.11e AP.

HCCA can be used not only in the CPF but also in CP together with EDCA, i.e., the HC can poll for data even during the CP. The part of the CP accessed using polling is called Controlled Access Phase (CAP). The presence of a CAP is possible thanks to the fact that, before accessing the channel, each station has to wait for an interframe space. As the interframe space of the HC is shorter than those of other stations, the HC has priority over any QSTA and it can allocate TXOP either to itself whenever it has a frame to send or to some other stations allowing them to deliver as many data frames as can be accommodated into the time allocated by the TXOP. As the HC (Hybrid Coordinator) allocates the TXOPs

through a polling, a scheduler is needed to select the order in which stations are to be polled. The IEEE 802.11e standard does not impose a mandatory HCCA scheduling algorithm, but it offers a reference scheduler that respects a minimum set of performance requirements, on the basis of the mean data rate, nominal MAC Service Data Unit (MSDU) size and either maximum service interval or delay bound information provided by the TSPEC.

2.3 Enhanced Distributed Channel Access (EDCA)

The EDCA mode extends the IEEE 802.11 Distributed Coordination Function (DCF) (IEEE 802.11, 2007) by differentiating traffic into four Access Categories (ACs): AC_BK (background category), AC_BE (best-effort category), AC_VI (video category) and AC_VO (voice category), where AC_VO is the highest priority category, while AC_BK is the lowest.
To manage the different ACs, EDCA implements in each node a dedicated transmit queue and an independent back-off entity for each AC. Each queue works as an independent DCF station and uses its own parameter set, including the Arbitration Inter-Frame Space (AIFS), the minimum Contention Window size (CWmin), the maximum Contention Window size (CWmax) and the Transmission Opportunity limit (TXOPlimit), that is the time duration a station may transmit after winning access to the medium. Similarly to an 802.11 DCF node, each AC starts a back-off timer after sensing an idle channel for a duration equal to an AIFS length. However, while in DCF all nodes have the same probability to access the channel, in EDCA the AIFS depends on the AC, so that its duration is shorter for the higher priority ACs, which thus have a higher probability of accessing the channel than the lower ACs (Fig. 2) .

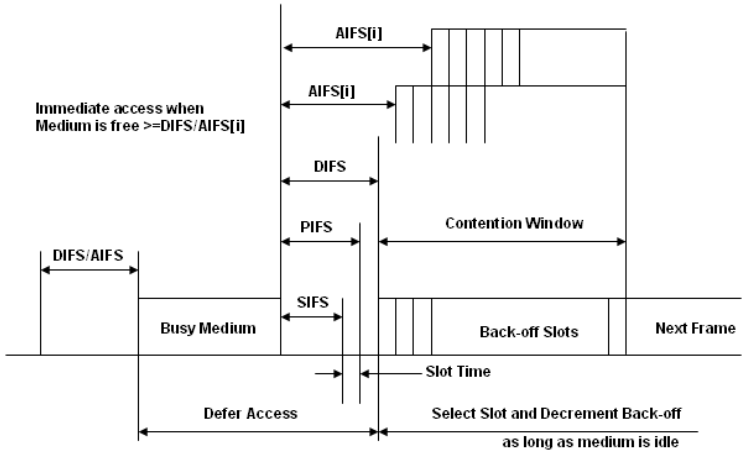


Fig.2. Inter Frame spaces in the EDCA mechanism, redrawn from (IEEE 802.11, 2007).

2.4 EDCA vs. HCCA in industrial environments

The nature of the two MAC protocols (EDCA and HCCA) is very different.

The HCCA protocol is the most suitable for real-time traffic, as the polling-based scheme can support time constrained applications, even though only statistic guarantees can be provided, due to the non-deterministic nature of the channel. HCCA is a centralized protocol, so the Access Point can adopt suitable scheduling and admission control algorithms to allocate at the MAC layer the timeslots needed to meet the station requirements.
On the other hand, EDCA is a protocol which provides differentiated services to different access categories, in which it is not possible to allocate bandwidth to the applications. Therefore, EDCA cannot be adopted in safety-critical systems, and not even for applications with tight timing constraints. But, in many practical situations in industrial systems, the occurrence of a missing frame does not affect human safety or the safety of production equipment. In such cases it is sufficient to ensure that the transmission deadlines are met most of the times. Hence, a number of works exist in literature that study the performance of the EDCA protocol in different situations (e.g. saturation and non-saturation). Such studies can be useful in implementing some admission control algorithms that allow to meet the time requirements in a probabilistic way.

3. Analytic models of IEEE 802.11e

3.1 Analytic models of EDCA

Several theoretical EDCA performance analyses exist in literature. All of them are implicitly or explicitly derived from analytic models of the 802.11 DCF.
As the theoretical analysis can potentially be complex, a proper set of assumptions is needed to obtain a simple yet accurate analytic model. Most of the models assume that every station has always backlogged data ready to be transmitted in its buffer at any time (in saturation). Other models release the saturation assumption. As a result, we categorized these works in two groups, i.e., models "under saturation" and "under non-saturation" conditions.

3.1.1 Analytic model under saturation condition.

Three major analytic models have been proposed to analyse the performance of CSMA/CA DCF scheme in saturation.

1. Bianchi (Bianchi, 2000) proposed an analytical evaluation, where the behaviour of a single station is modelled with a simple Discrete Time Markov Chain (DTMC). In this analysis, the author makes the assumption of constant and independent collision probability of a packet transmitted by each station, regardless of the number of retransmissions already suffered. The outcome of the Markov model is the stationary probability that a station will transmit in a generic slot time. Then, the saturation throughput is expressed as a function of such a stationary probability, by studying the events that may occur in a generic slot time.
2. Calì et al. (Calì et al., 1998 & 2000) use the renewal theory to analyse a p-persistent variant of DCF, which differs from the standard protocol only in the selection of the back-off interval. In particular, the p–persistent IEEE 802.11 protocol samples its back-off interval from a geometric distribution with parameter p, while in the standard a binary exponential back-off is used. This model is able to quantify the maximum achievable throughput as a function of the protocol parameters, e.g., the contention window size. Moreover, the authors showed by means of simulations

that the p–persistent protocol closely approximates IEEE 802.11 when the average back-off interval is the same.

3. Tay and Chua (Tay & Chua, 2001) make some simplifications in the 802.11 MAC and model it with an average value mathematical model to evaluate DCF back-off procedure and to calculate the average number of interruptions that the back-off timer experiences. This model assumes slot homogeneity, i.e., constant collision probability at an arbitrary back-off slot, to provide closed-form expressions for the collision probability and the saturation throughput, thus facilitating the analysis of the dependence of system performance on the protocol parameters. The model validation showed that, even though this model omits many system details, it still achieves good accuracy.

These models are used and extended (especially Bianchi's work) to model the EDCA scheme and to include its extra features.

Xiao (Xiao, 2004 & 2005) modified the Markov chain in (Bianchi, 2000) to model resource sharing by different access categories and to analyse the CW size differentiation in the EDCA mechanism. This model is able to derive saturation throughput, delays and frame dropping probabilities of the different priority classes. It differentiates the minimum back-off window size, the back-off window-increasing factor and the retransmission limit of the different access categories. However, IFS-based priority is not taken into account.

The work in (Robinson & Randhawa, a & b 2004) keeps the two dimensional Markov chain, but takes into account that different contentions can experience different collision probabilities because the probability that a transmission collides is a function of the size and composition of the set of competing stations. Moreover, they perform an analysis on back-off and transmission after a collision occurs.

In (Kong et al., 2004), the authors took into account AIFS differentiation via a 3-dimensional DTMC. They introduced a separate one-dimensional Markov chain to be used along with Bianchi's model, so that the model can reflect the back-off and access procedures accurately. Moreover this model takes into account the back-off timer freeze that occurs when a station is deferring, and the virtual collision policy. They analyze the throughput performance of differentiated service traffic and propose a recursive method capable of calculating the mean average delay.

In (Inan et al., 2008), the authors analysed the saturation throughput of EDCA by means of a DTMC which models AIFS and CW differentiation between the ACs in the constant transmission probability assumption. Moreover they showed that the slot homogeneity assumption does not lead to accurate performance prediction when the saturation assumption is released.

Another analytical model based on a three-dimensional DTMC is proposed in (Tao & Panwar, 2004 & 2006), where the third dimension models the state of back-off slots between successive transmission periods. The fact that the number of idle slots between successive transmissions can be at most the minimum of AC-specific AIFS values is also considered. This model is able to compute the maximum sustainable throughput and service delay distribution for each priority class when the network is under heavy load.

In (Hui & Devetsikiotis, 2004, 2005 & 2006), the authors combined several major analytic approaches, including Bianchi's Markov model, Calì's p-persistent CSMA model and Tay's average-value model for DCF analysis, into one average model, to borrow the strengths from these models and compose a unified performance model of EDCA. In particular, they

use renewal theory to express the throughput, but assume constant packet-transmission probabilities for different stations taking into account the differentiation of both AIFS and CWmin/CWmax. Another assumption is that each packet-transmission probability only depends on a unique collision probability.
In (Lin & Wong, 2006) the authors extended the work in (Tay & Chua, 2001) and used the mean value analysis to evaluate saturation throughput in 802.11e networks. The authors argue that the use of multi-dimensional Markov chains and other non-linear equations lead to high computational complexity. Thus, although some of these models provide good accuracy, the high computational complexity make them unsuitable for real-time control functions such as on-line admission control in IEEE 802.11e networks. Their approach is simpler, and yet it maintains good accuracy.

3.1.2 Analytic model under non-saturation condition.

In non-saturation models, the saturation assumptions are released basically following two principal methods; (1) modelling the non-saturated condition with Markov analysis, (2) using queuing theory (Kleinrock, 1975) to evaluate certain performance figures through average or Markov analysis.
To perform Markov analysis under non-saturated conditions slot homogeneity is assumed and the model in (Bianchi, 2000) is extended with necessary extra Markov states and transitions. Similar extensions of (Bianchi, 2000) for the non-saturated analysis of 802.11 DCF are proposed in (Duffy et al., 2005) and in (Alizadeh-Shabdiz & Subramaniam, 2004 & 2006). These extensions assume a buffer size for the MAC layer of only one packet. However, such an assumption is shown to lead to significant performance prediction errors for EDCA in the case of larger buffers. An extension of the model provided in (Alizadeh-Shabdiz & S. Subramaniam, 2004) is given in (Cantieni et al., 2005), where station buffers are assumed infinitely large and the MAC queue is assumed empty with a constant probability, regardless of the back-off stage in which the previous transmission took place. In addition to perform throughput analysis, this model is also able to estimate MAC delay.
In (Engelstad & Osterboused, 2006), a DTMC model is used to perform delay analysis for both DCF and EDCA, considering the same queue utilization probability as in (Cantieni et al., 2005). While most of the analytical models dealing with delay performance of IEEE 802.11 focus on the prediction of only the mean delay due to the MAC functioning, this model provides analytical predictions of the total delay, which also includes the time spent in queue. Therefore this model can be used by higher layer protocols and applications that are interested in the overall performance of the transmission, including queuing delay.
Several models use queuing theory to carry out 802.11(e) performance analysis in non-saturated conditions. These models need some independent analysis for the calculation of some figures such as collision and transmission probabilities. In (Tickoo & Sikdar, a & b 2004), each 802.11 node is modelled as a discrete time G/G/1 queue to derive the service time distribution. However, such a model is based on the assumption that certain quantities in non-saturated conditions can be approximated well using in-saturation analysis. In (Chen et al., 2006) both G/M/1 and G/G/1 queue models are used on top of the model provided in (Xiao, 2005), which only considers CW differentiation. The use of M/G/1 queuing model together with a simple non-saturated Markov model to calculate necessary quantities is analysed in (Lee et al., 2006). In (Foh & Zukerman, 2002) a framework to analyze the performance of DCF under statistical traffic based on Markov chains is presented, where the

number of contending nodes is modelled as an M/Ej/1/k queue. Such work is extended in (Tantra et al., 2006) to take into account EDCA service differentiation. This work analyses both the throughput and delay performance of EDCA mechanism under statistical traffic. However, such an analysis is valid only when all nodes have a MAC queue size set to one packet.

3.2 HCCA

Currently only a few analytic works on HCCA exist in literature. In (Rashid et al., 2006) the authors introduced a novel analytic queuing framework that allows the performance of HCCA to be analysed when supporting Variable Bit Rate (VBR) traffic applications. Additionally such a framework is useful to support the design of HCCA scheduler and admission control mechanisms.
In (Ghazizadeh et al., 2009), the authors presented a priority queuing model to analyse medium access in the HCCA by making use of an MAP/PH/1 queue with two types of jobs which are suitable to support a wide range of traffic streams.

4. Admission Control for EDCA

The EDCA does not implement any mechanism to provide guarantees to real-time traffic. Nevertheless, several works in the literature, e.g., (Moraes et al., 2006), (Mangold et al.,b 2002) and (Gu & Zhang, b 2003), proved that the protocol can provide a good level of QoS in terms of delay, jitter and bandwidth under low workload conditions. As a result, limiting the number of traffic flows that can gain access to the channel so as to limit the total workload in the network makes it possible to obtain satisfactory performance.
The selection of which traffic can be admitted without compromising the performance of previously admitted traffic is the task of Admission Control.
The IEEE 802.11e standard (IEEE 802.11, 2007) specifies a mechanism to implement an admission control in EDCA scheme. The admission control applies to any AC when the associated Admission Control Mandatory (ACM) field is activated (in the beacon frame). In that case, all STAs that wish to use this AC must send to the QAP an Add Traffic Stream (ADDTS) request containing the TSPEC of the new flow. After receiving an ADDTS request, the Admission Control, on the basis of the received TSPEC and the current system state, decides whether to:

i) admit the new flow with the requested TSPEC,
ii) suggest a different TSPEC that might be admitted by the network, or
iii) reject the flow.

Once a decision is taken, the Admission Control sends an ADDTS response packet to the corresponding STA, notifying of the decision. When the station receives the response, it decides whether the TSPEC suggested by the AP satisfies or not its traffic requirements. If not, the STA is allowed to schedule another ADDTS request after a period of time. If both the QAP and the QSTA accept, the flow becomes active. When the flow finishes, the QSTA must send a Delete Traffic Stream (DELTS) packet to allow the QAP to release the resources used by this flow.
While the standard describes the message exchange to implement admission control, it does not provide any mechanism to determine how to decide which traffic should be admitted to maintain a desired QoS level. However, in the past few years much research has focused on

admission control in EDCA. Basically, the existing EDCA admission control algorithms can be classified into two categories: measurement-based admission control and model-based admission control. In the measurement-based schemes, admission control decisions are made on the basis of the continuously measured network conditions such as throughput and delay. On the other hand, the model-based schemes use some analytic model to evaluate certain performance metrics to assess the status of the network.

4.1 Measurement-Based Admission Control

In the Distributed Admission Control (DAC) presented in (Xiao & H. Li, a & b 2004), the QAP announces the transmission budget within the beacon frame. The transmission budget is the additional amount of time available for either a new flow or existing flows to increase their transmission time per AC in the next beacon interval. To calculate such a budget, the QAP counts the amount of time occupied by the transmissions of each AC during each beacon interval. Then, the QAP calculates the transmission budget for each AC by subtracting the occupied time from the transmissions of the relevant AC. Moreover, each station determines an internal transmission limit per AC within each beacon interval on the basis of the successfully used transmission time during the previous beacon period and the transmission budget announced from the QAP. When the transmission budget for an AC is used up, a new QoS flow will not be able to obtain any transmission time, and existing QoS flows will not be able to increase their transmission time.

Based on the DAC scheme, (Xiao et al., 2004) proposed a two-level protection and guarantee scheme. The purpose of the first-level protection is to protect each existing AC_VO and AC_VI flows from new and existing QoS flows, while the purpose of the second-level protection is to protect QoS flows from best-effort traffic.

For the first-level protection, two enhancements, i.e., *tried-and-known* and *early-protection*, are introduced in the original DAC scheme. The *tried-and-known* is an enhancement of DAC mechanism and is only applied to new flows. Each station keeps track of the throughput and delay performances obtained in previous beacon intervals. If the average throughput and/or delay do not meet the flow requirements, the flow will be rejected. The *early-protection* mechanism prevents two or more new flows from being admitted during the same beacon interval when the budget is below a certain threshold. While this first-level protection is able to protect each existing QoS flow from other QoS flows, it cannot protect QoS flows from best-effort traffic, as this kind of traffic does not need admission control. However, too much best-effort traffic can degrade the performance of QoS flows since many collisions might occur. Therefore, a second-level protection is introduced that dynamically tunes the EDCA parameters of the QSTAs. The basic idea is to increase the initial contention window size and interframe space for best-effort traffic when the number of active stations is large so the number of collisions decreases and the bandwidth available increases. This is controlled by the QAP, that sends such parameters within the beacon frame.

The Threshold-Based Admission Control, proposed in (Gu & Zhang, a 2003), works in both ad hoc and infrastructure modes, so each station needs to measure the traffic condition on the wireless link and decide whether to transmit or reject traffic flows of a specific AC on the basis of network-load. The authors proposed two different methods to assess the network condition and to implement the admission control:

1. **Using relative occupied bandwidth:** T_{Busy} is defined as the amount of time (within a given time period, e.g., a beacon interval) the wireless medium is busy, while the

B_{occu} is the relative occupied bandwidth and it indicates at what percentage the wireless medium is being used. The lower and upper thresholds for B_{occu}, to maintain a desired QoS can be obtained from simulations or/and calculations, and are indicated as B_{lo} and B_{up}, respectively. Defining "*active AC*" as each previously admitted AC and "*inactive AC*" as each AC that has been refused, the criteria to admit data flows can be summarized as follows:

- $B_{occu} \leq B_{lo}$: Admit the inactive AC with the highest priority during the next period of T .
- $B_{lo} \leq B_{occu} \leq B_{up}$: No action taken.
- $B_{occu} \geq B_{up}$: Stop the transmission of the lowest active AC during the next period of T.

2. **Using average collision:** In this case, instead of using the relative occupied bandwidth, a new parameter, that is the average collision ratio during a fixed sampling period, is employed for admission control. The average collision ratio is defined as $R_c = N_c / N_t$, where N_c is the number of collisions that have occurred (estimated through the number of retransmissions) and N_t is the total number of transmissions. Similarly, there are two thresholds: the lower threshold R_{lo} and the upper threshold R_{up}. The criteria for admission control is the same as that in the case of using relative occupied bandwidth, with the difference that the new R_c, R_{lo} and R_{up}, parameters are used in place of B_{occu}, B_{lo} and B_{up}, respectively.

Although this threshold-based admission control is very easy to implement, the threshold values are difficult to set.

A different approach is the HARMONICA scheme (Zhang & Zeadally, 2004), which works only in BSS mode. The AP periodically samples the link-layer quality indicator (LQI) parameters, which include drop rate, end-to-end delay and throughput, to dynamically adjust the channel access parameters CWmin, CWmax and AIFS for each traffic class. A simple and flexible technique to avoid congestion is proposed, which employs two different adaptation algorithms:

1) The relative-adaptation algorithm adjusts the relative differences between the channel access parameters of different classes to obtain QoS guarantee.
2) The base-adaptation algorithm synchronously adapts the channel access parameters of all the classes (increase all or decrease all) to achieve high channel utilization.

Whenever a new real-time application requires admission, HARMONICA will select the traffic class i that best matches its QoS requirement and then execute an admission control process. The decision of admission control is based on the throughput requirement ($Req_{throughput}$) of the flow and the monitored LQI parameters. The basic idea is to check whether it is possible to take the required bandwidth from the best effort class ($BE_{throughput}$) while guaranteeing a minimal bandwidth (BE_{Min}) for the best-effort traffic class. In particular, in order to admit a new QoS flow, three requirements need to be satisfied:

1. The relative adaptation has reached a stable state;
2. $BE_{throughput} - Req_{throughput} \geq BE_{Min}$;
3. The bandwidth in $BE_{throughput}$ can be "translated" into Class i without loss.

Results provided in (Zhang & Zeadally, 2004) show that, by dynamically adjusting channel access parameters, it is possible to simultaneously match the QoS requirements, maximally utilize network resources and guarantee a minimal bandwidth for the best-effort traffic.

4.2 Model-Based Admission Control

A model based approach is proposed in (Pong & Moors, 2003), where admission control is performed on the basis of the predicted achievable throughput for each flow. The estimation of the achievable throughput is based on the two-dimensional Markov Chain model proposed in (Bianchi 2000), the authors extend such a model so as to estimate the throughput of flows belonging to different ACs, each with different channel access parameters and to incorporate the concept of TOXP into the model. The model takes the following as inputs: EDCA parameters (such as CWmin and the back-off stage), the payload of data packets and some statistics on the channel (such as successful transmission probability, collision probability, idle probability).

The admission control is performed by the AP. Before transmitting its data, a station has to make a request to the admission controller (AP) to obtain the desired bandwidth. If the total traffic (already admitted plus the new flow) exceeds the bandwidth limit, then the request is rejected, otherwise it is accepted. In the latter case ACU searches the best parameters for CWmin and TXOP duration (if used), given the required bandwidth of the new stream.

A Contention Window Based admission control is proposed in *(Banchs et al.,2003)*. The aim of this approach is to adjust the CW values of the different stations so as to reach the goals of admission control. Consider a set of n IEEE 802.11e nodes which is operating with a contention window set $\{CW_1,...,CW_n\}$, that satisfies the throughput requirements $\{R_1,...,R_n\}$ for all the stations. Denoting r_i as the actual throughput experienced by the station i, r_i is greater than R_i for each i=1,...,n. When a new station (n + 1) with a throughput requirement of R_{n+1} wishes to join the network, firstly a new contention window set $\{CW_1^{'},...,CW_n^{'},CW_{n+1}^{'}\}$ is calculated using the analytical model proposed in (Banchs et al.,2003), then the same model is used to compute the throughput. If the resulting throughput meets the requirements, the station (n + 1) is accepted and the new contention window set is distributed to all the stations. Otherwise, the station (n+1) is rejected.

5. Improving real-time traffic support in IEEE 802.11e

Real-time behaviour is one of the main concerns in industrial networking. This section will discuss the most relevant proposals to enhance real-time traffic support in IEEE 802.11e, in both HCCA and EDCA modes.

5.1 Scheduler and ACU for HCCA

In the HCCA reference scheduler, both Service Interval and TXOP are fixed values that are recomputed each time a new Traffic Stream (TS) arrives. Several simulation studies, such as (Casetti et al., 2005) and (Yang et al., 2008), show that the reference scheduler is inefficient, especially when dealing with variable bit rate (VBR) traffic. The reason is that all the TSs, although having different characterizations, are polled with the same period and are granted the same transmission time. As a result, the TXOPs are often overestimated and a significant amount of bandwidth may be wasted.

Several works in literature propose new scheduling algorithms to improve the HCCA performance for either constant bit rate (CBR) or VBR traffic. As most of the traffic in industrial applications is CBR, the former algorithms are more suitable with industrial applications. On the contrary, the latter are more suitable for multimedia applications (Voice

and Video). The most relevant algorithms for CBR traffic are Real-Time HCCA and Group Sequential Communication (GSC).
The Real-Time HCCA (RTH) algorithm (Cicconetti et al, a 2007) is designed to provide real-time support to traffic flows with a given capacity and period. This technique is composed by some offline and online activities. The most complex activities, i.e., the *admission control* and *timescale computation*, are performed offline, while the *enforcement procedure* is performed online.
In the admission control phase, the QAP executes the admission control algorithm each time a QSTA requests to transmit a new TS. Using the TSPEC parameters contained in such requests, the QAP calculates two other parameters for each new TS: the capacity C_i and the period T_i. C_i is the minimum capacity needed to comply with the TS requirements and is calculated as

$$C_i = [\frac{(R_i \cdot T_i)}{N_i}] t_{N_i} \tag{2}$$

where the R_i is the mean data rate, N_i is the nominal SDU size, t_{Ni} is the nominal transmission time. The period T_i is set equal to the delay bound D_i if D_i is smaller than $[N_i/R_i]$, otherwise is set to $\lfloor (R_i/N_i) \cdot D_i \rfloor \cdot N_i/R_i$. Then, the QAP verifies the schedulability of the traffic using the test introduced in (Baker, 1991) in the multi-programmed environment, treating the non-preemptability of transmissions as a critical section. In this case, the minimum critical section bi for a T_{Si} is $b_i = t_{Ni} + t_{Pi}$, where t_{Pi} is the poll time for uplink TSs .
When TS_i is in a critical section, and is thus scheduled instead of the highest priority TS_j, TS_i is said to block TS_j. So, the blocking time for a TS_i is the maximum critical section duration of TSs with a period longer than TS_i, i.e., $B_i = \max_{j>i}\{b_j\}$ considering that the TSs are sorted by increasing period duration, i.e., $i>j \rightarrow T_i \geq T_j$. The schedulability analysis produces the following sufficient condition to determine the set of n schedulable TSs and which has an O(n) computational complexity:

$$\frac{B_i}{T_i} + \sum_{j \leq 1} \frac{C_j + \pi_j + t_{P_j}}{T_i} \leq 1 \qquad \forall i: \ 1 \leq i \leq n \tag{3}$$

The admission control procedure produces also a set of parameters describing each TS, composed by $[i,t_i,TXOP_i]$ tuples, where i is the index of the next QSTA which can access the medium, t_i is the polling time and $TXOP_i$ is the duration of its transmission. The admitted TSs are scheduled in EDF-order.
The online enforcement activity only consists in reading the next entry in the timetable, $[i,t_i,TXOP_i]$, waiting until time t_i and then granting a TXOP of duration $TXOP_i$ to TSi.
The GSC mechanism presented in (Viègas et al., 2007) is a technique to reduce the polling overhead of the HCCA scheme. The authors propose to grant the channel access to RT-stations using a virtual token passing. Stations are classified in two different categories: Real-time and Non Real-time. The GSC procedure groups all the RT stations in a Sequential Group (SG), and identifies them as a vector $L=\{GI_1,GI_2,\ldots,GI_{np}\}$, where GI_i is the station identifier and *np* is the number of RT stations. Each station maintains a local Sequence

Counter (SC) that is the image of the distributed variable (SC). A station having the identifier GI captures the virtual token when SC is equal to GI_i.
The GSC mechanism works as follows: at the beginning of the CFP, the HC sends a beacon frame and the stations set their NAV, while the SC counter is set to 1. If an RT station is ready to send a frame and the SC is equal to its GI, it is authorized to transmit the real-time message. Once the message has been sent, the other stations listen to the channel and at the first slot-time after the SIFS they increment the SC counter. On the contrary, if a station does not have any frame to send when the SC is equal to its GI_i, then the SC value will be incremented in all the RT stations after a slot-time. This way, the token is implicitly passed to the next RT-stations, until the end of the RT section of L.
The timing analysis provided in (Viègas et al., 2007) shows that the GSC mechanism guarantees a smaller polling overhead than the reference scheduler.
There are several works in literature where new schedulers are investigated to improve the performance of VRB traffic.
In (Grilo et al., 2003), the SETT-EDD scheduling is proposed, that uses the Earliest Due Date policy on the basis of the Estimated Transmission Times and the minimum Service Interval (mSI). The minimum service interval (mSI), i.e., the minimum time that must elapse between two consecutive TXOPs, is specified directly in the TSPEC, while the transmission time is estimated through the maximum burst size given by the TSPEC. Using the arrival time of packets to serve SDUs according to the Earliest Due Date algorithm, the SETT-EDD is shown to perform better than the reference scheduler for the scheduling of VBR traffic.
The Wireless Timed Token Protocol (Cicconetti et al., b 2007) is based on the *Timed Token Protocol* (TTP) (Grow, 1982), which is a token passing scheme employed as a MAC protocol for ring-based networks.
The token circulation is ruled by the *Target Token Revolution Time* (TTRT), which is a parameter indicating the reference round duration: its value is computed by the QAP according to the TSPEC values negotiated by the QSTA during the admission control phase. This mechanism provides CBR streams with a fixed capacity, while a minimum reserved rate is allocated for VBR flows.
The Feedback Based Dynamic Scheduler (Boggia et al., 2007) assigns dynamically the TXOP according to queue length estimation, while SI remains fixed. This technique uses the optional piggybacking mechanism to exchange information about the queues together with a discrete time model to estimate the length of uplink TS queues. The duration of the scheduled TXOPs depends on the requirements estimated by the HC during each period.

5.2 Improving EDCA performance

Several works in literature proved with simulations (Vittorio et al., a 2007), experimental measurements (Narbutt & Davis, a & b 2007) and with analytic models that the EDCA parameters (CWmin, CWmax, AIFS, TXOP) strongly affect the performance of the protocol. Some research works also investigated the effect of EDCA TXOPs on the performance of 802.11e under a saturated scenario. In both (Mangold et al., a 2002) and (Suzuki et al., 2006) the performance analysis is carried out through simulation. In (Tinnirello and S. Choi, 2005) the efficiency of burst transmissions with block acknowledgements is analyzed.
As a result, recent literature presented some mechanisms to dynamically change these parameters on the basis of the channel condition.

In (Sawaya et al., 2005), the authors proposed an adaptive back-off procedure for EDCA that, instead of using the standard back-off procedure, sets CW[AC] of the back-off timer directly to the most adequate value, calculated on the basis of the congestion level. As in the standard EDCA, the CW is incremented whenever a station fails to transmit. In fact, the most effective CW values are close to CWmax when the channel congestion is high, while they are close to CWmin when the channel is not congested. The congestion level is assessed through a parameter called *ratio* and defined as

$$ratio = weight \times \frac{CW_{Current} - CW_{\min}}{CW_{\max} - CW_{\min}} \tag{4}$$

where *weight* is a constant reflecting the accuracy of the channel estimation. The shorter the time between the last transmission and the current estimation, the greater the value of the weight. The new value of the CW for the next transmission is calculated as

$$CW_{new} = ratio \times (CW_{current} - CW_{\min}) + CW_{\min} = weight \times \frac{(CW_{Current} - CW_{\min})^2}{CW_{\max} - CW_{\min}} + CW_{\min} \tag{5}$$

This CWnew value is used as the back-off timer and is shown to be effective in adapting performances to the varying network condition.
In (Romdhani et al., 2004) the authors propose an adaptive mechanism to find the most appropriate values of the back-off parameters for each AC. While the standard EDCA scheme resets the CW[AC] value to the CW_{min}[AC] value after each successful transmission, the proposed scheme assesses channel status by taking into account the estimated collision rate, f_{curr}^{j}, whose value is computed as

$$f_{curr}^{j} = \frac{E(collions_j[p])}{E(data\_sent_j[p])} \tag{6}$$

where $E(collisions_j [p])$ is the number of collisions experienced by the station p at step j, and $E(data\_sent_j [p])$ is the total number of packets that have been sent in the same period. To minimize the bias against transient collisions, an Exponential Weighed Moving Average is used to smoothen the estimates values, i.e.,

$$f_{avg}^{j} = (1-\alpha) \times f_{curr}^{j} + \alpha \times f_{avg}^{j-1} \tag{7}$$

where α is a constant in (0,1) that determines the memory of the estimation.
To maintain the priority-based discrimination between different classes, a Multifactor Factor MF[AC] (on the formula) is defined for each class. This factor is limited to 0.8 and is defined as

$$MF[i] = \min((1 + (AC \times 2)) \times f_{avg}^{j}, 0.8) \tag{8}$$

Using this formula, the highest priority is given to the CW parameter with the smallest MF values.

After each successful transmission the CW value is updated as

$$CW_{new}[AC] = \max(CW_{\min}[AC], CW_{old}[AC] \times MF[AC]) \quad (9)$$

while after an unsuccessful transmission (collision) of a packet belonging to class AC the CW value is updated as

$$CW_{new}[AC] = \min(CW_{\max}[AC], CW_{old}[AC] \times PF[AC]) \quad (9)$$

where PF is a parameter proposed in a draft version of the IEEE 802.11e standard to calculate the CW_{new} value, and it is used by the authors to ensure that high priority traffic has the smallest CW values.

In (Vittorio et al., b 2007) a new mechanism is proposed, called a Contention Window Adapter (CWA). Instead of setting the CW to an ideal value within the fixed [CWmin, CWmax] range defined by the standard, which is shown to be inappropriate in many network load conditions, the CWA adjusts the range of the current CW (i.e. the values of CWmin and CWmax) of all the ACs, on the basis of the workload information, which is estimated on the basis of the retransmission count. Although the CWA is based on empirical rules, it is shown to improve significantly the performance of the Real-Time flows, which are mapped into the AC_VO category. A similar approach is followed in (Vittorio et al., 2008), where a Contention Window Fuzzy Controller (CWFC) uses fuzzy logic in place of the simple empirical rules of the CWA to dynamically find the most appropriate CW range on the basis of the network status, estimated in terms of both global throughput and local retransmissions count.

The Virtual Token Passing-CSMA (VTP-CSMA) architecture has been proposed in (Moraes et al., 2007) to handle the timing constraints of real-time traffic when real-time devices share the wireless channel with devices that are not time constrained. Two different types of stations are considered: RT (Real-Time) and ST (Standard Station). A traffic separation mechanism (TSm) is realized, so that, when a collision occurs, the ST stations select a random back-off interval according to the access category, while the RT stations (that transmit their traffic at the highest priority of EDCA) set both the CWmin and CWmax parameters to zero. This way, if two or more RT stations simultaneously contended for the medium access, they would continuously collide and discard the frame. To avoid this problem, the VTP-CSMA serializes the transmission of the RT stations. At the design phase of the network each RT station is assigned a progressive number, which is used to pass a token between the RT stations. In particular, RT stations continuously listen to the wireless channel and, by counting the number of the elapsed time-slots, each station knows whether the token belongs to it or not. Once a station obtains the token, if it has an RT message to transfer, it can transmit it. When it finishes transmitting, the token will pass to the next station. However, if a station obtains the token while not having any frame to transmit, the token will immediately pass to the next RT station (with the subsequent number).

6. The IEEE 802.11e standard in industrial environments: case studies

Several works in literature analyze the suitability of IEEE 802.11e for industrial communications by means of either simulations or real measurements in case-study

scenarios emulating real industrial applications. In the following some relevant works will be shown, categorized in two classes, i.e., case studies using HCCA and EDCA, respectively.

6.1 The IEEE 802.11e HCCA standard in industrial scenarios

In (Karanam et al., 2006) and in (Trsek et al., 2006) the authors evaluated the performance of the HCCA in an protocol industrial automation network with real-time requirements by means of a simulation case study using the network simulator OPNET and compared the results with those obtained with EDCA in terms of latency in various scenarios. The authors assessed the performance of the HCCA protocol in two industrial scenarios. In the former only real-time traffic is present, while in the latter there are both real-time and non real-time traffic. An extended 802.11b (11 Mbps) model of OPNET was used, featuring the reference scheduler described by the IEEE 802.11e standard, hence with TXOP and SI values equal for all the Traffic Streams (TSs).

Two types of TS were considered: downstream and upstream. A PLC is connected directly to an AP through a real-time Ethernet network and generates cyclic data to update the outputs of the remote I/O devices. The I/O nodes sent cyclic data of their inputs to PLC through the WLAN. The traffic flow from node to PLC is defined as upstream and from PLC to node as downstream.

Simulations were performed with a growing number of stations, i.e., from 2 to 80 for EDCA and from 2 to 100 for HCCA. As the maximum number of clients in HCCA is constrained by the chosen service interval (SI), the SI was increased according to the growing number of stations. The size of the packets was set to 40 byte for both TSs.

Results showed that for a small number of stations the delays experienced by HCCA and EDCA are similar, as there is a small number of contentions for the medium. However, when the number of the stations exceeds 25, the EDCA delay increases exponentially due to the large number of collisions and retransmissions. Similar results were obtained maintaining a fixed number of stations while increasing the network load.

The authors conclude that HCCA is more suitable than EDCA for the support of industrial traffic, as EDCA becomes inefficient and unreliable when there is either a large number of stations or high network load, while HCCA remains more predictable and reliable.

6.2 The IEEE 802.11e EDCA standard in industrial scenarios

In (Moraes et al., 2006) the authors assess, by means of simulations, an industrial scenario consisting of an open communication environment (OCE), where the traffic from RT stations share the same communication medium with generic multimedia (voice and video) and background traffic from a set of standard (ST) stations. Basically two simulation scenarios are analysed: the small population scenario, which considers the case of 20 stations (10-RT; 10-ST), and the large population scenario, which extends the small population scenario to 50 stations (10-RT; 40-ST). Each station operates at orthogonal frequency division multiplexing (OFDM) PHY mode and the PHY data rate is set to 36 Mbps. Each RT station generates 1 packet every 2 ms with a 45 bytes data payload, while the load offered by ST stations ranges from 5% to 95% of the PHY data rate (36 Mbps).

The results showed that the RT stations are able to transfer significantly more packets containing VO traffic than ST stations, even though the same access category is used. Such improvement is due to the TXOP concept introduced in the 802.11e amendment that defines

the time interval during which a station is able to transfer a burst of packets from the same access category, after winning the medium access. Consequently, an RT station will be able to transfer a higher number of packets than an ST station using the same access category (VO), because RT-VO packets are shorter than ST-VO packets.
When increasing the number of stations contending for the medium access, there is a degradation of the QoS for large population scenarios.
The authors showed that real-time traffic transferred by RT stations has an average packet delay slightly worse than the voice traffic transferred by ST stations, although RT stations were able to transfer more messages than ST stations.
The authors conclude that the default values of the EDCA parameters are not able to guarantee the timing requirements of industrial communication when the AC_VO class is used to support real-time traffic in shared medium environments and other types of traffic are also present.
In (Cena et al., 2008) the authors performed an in-depth evaluation of the performance achievable by EDCA in industrial environments. The Authors provide a perspective on requirements and characteristics of the traffic typically found in industrial control applications. Four different traffic categories are defined:

- Urgent asynchronous notifications (alarm, *RT0*);
- Process data sent on a predictable schedule (periodic,*RT1*);
- Process data sent on a sporadic schedule (*RT2*);
- Parameterization service (*NRT*).

RT0 traffic is related to either alarms that are generated spontaneously by devices (failure/error notifications) or asynchronous time-critical commands sent by the application master. RT1 traffic consists of process data characterized by real-time requirements that are generated in a predictable way (periodic traffic). The authors simulate the access to channel of this traffic as a TDMA scheme where the transmission is organized as a repeated communication cycle of fixed duration. Within each cycle, each station sends periodic frame in its assigned slot(s) (e.g. using synchronization). RT2 traffic is similar to RT1 traffic but it is generated in an unpredictable way (aperiodic). Finally, NRT traffic is related to network operations with no particular real-time requirements (e.g., remote configuration, management and diagnostics).
The authors mapped the RT0 on AC_V0 (highest priority), the RT1 on AC_VI, the RT2 on AC_BE and the NRT on AC_BK. The scenario evaluated is composed of 20 stations, 10 of them, defined as “stations under test”, that produce a specific kind of traffic and 10, defined as “the interfering stations”, that generate low priority traffic. The performance evaluated is the response time, defined as the time elapsed between the transmission request issued at the sender and the receiving time at the intended destination.
The work presents many results obtained by several simulations with different scenario settings. In general, the Authors show that EDCA (enhanced through TDMA techniques to exploit the knowledge about predictable traffic) can be considered a suitable solution for industrial applications, as long as safety and/or time critical requirements are not a primary issue. In fact, the average performance resembles closely those achievable with the currently existing fieldbus networks, but, compared to fieldbuses, WLANs exhibit a noticeably lower degree of determinism.

7. Conclusions

This chapter addressed the case for wireless networks in automation and the significant efforts currently made by a large community of researchers, from both academia and industry, to investigate suitable solutions to adapt the IEEE 802.11e standard to the industrial communication needs on the factory floor.
This chapter provided an overview of current literature concerning the use of IEEE 802.11e in industrial environment, focusing on real-time performance of both EDCA and HCCA mechanisms. The limits of such protocols have been discussed and some notable works to improve their real-time performance have been presented. Such works can be used and combined to improve the support for real-time industrial traffic. As an example, studies on the EDCA admission control algorithms that limit the workload in a wireless network might take advantage of some analytic models predicting the performance of the protocol from the workload and the protocol parameters to provide probabilistic guarantees.
Finally, this chapter discusses the results from case studies that analyse the performance of IEEE 802.11e networks in realistic industrial scenarios.
Despite the significant effort of researchers, there are still some open issues concerning the introduction of wireless local area networks (WLANs) in the factory floor. The most relevant is how to achieve performance guarantees while using an unreliable and non-deterministic wireless channel. Other open issues are: the integration with pre-existing wired networks, so as to form hybrid architectures that are still able to meet the performance requirements; the support for mobility and handover under real-time and reliability constraints; security and privacy of industrial communications; scalability of real-time wireless networks. All these issues are currently object of notable research efforts.
Among these efforts, there is the Flexible Wireless Automation in Real-Time Environments (flexWARE) collaborative project, funded by the European Commission under the 7FP. This project aims at providing real-time communication on the factory floor with wireless local area networks (WLANs), with a special focus on security, flexibility and node mobility.
The outcome of the flexWARE project will be a turnkey system that can overcome the restrictions of the state-of-the-art wireless real-time systems, which are bounded to a single cell, rather than a multiple cell network covering the whole factory, and will define a platform that fulfils the requirements of flexible wireless communications. In the flexWARE architecture, the wireless infrastructure is integrated with a real-time backbone network that can be used to connect different nodes spread over the entire factory floor. Moreover, such an infrastructure can transparently switch between access points. In addition, it can provide time synchronization, location awareness and security. All these features are offered without compromising on the real-time feature of the whole system.

8. References

IEEE 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard (Sept. 1999).

IEEE 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer Extension in the 5 GHz Band, Supplement to IEEE 802.11 Standard (Sept. 1999).

IEEE 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard (June 2003).

IEEE Std 802.11TM, IEEE Standards for information Technology, 2007.

Alizadeh-Shabdiz, F. and Subramaniam, S. (2004). "Analytical Models for Single-Hop and Multi-Hop Ad Hoc Networks", *Proceedings of the First International Conference on Broadband Networks*, pp. 449 – 458, ISBN: 0-7695-2221-1, IEEE Computer Society Washington, DC, USA.

Alizadeh-Shabdiz, F. and Subramaniam, S. (2006). "Analytical Models for Single-Hop and Multi-Hop Ad Hoc Networks," *Mobile Networks and Applications,* Vol. 11 , Issue 1, pp. 75–90, ISSN:1383-469X.

Banchs, A.; Perez-Costa, X. and Qiao, D. (2003). "Providing throughput guarantees in IEEE 802.11e wireless LANs," in *Proc. the 18th International Teletraffic Congress(ITC-18)*, Berlin, Germany.

Baker, T.P.(1991). "Stack-based scheduling of real-time processes". *Journal of Real-Time Systems*, Vol. 3, No. 1, pp. 67-99, ISSN: 1573-1383, Springer Netherlands

Bianchi, G. (2000). "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communications*, Volume 18, Issue 3, pp. 535–547.

Boggia, G.; Camarda, P.; Grieco, L. A. and Mascolo, S. (2007). "Feedback-Based Control for Providing Real-Time Services with the 802.11e MAC". *IEEE/ACM Transactions on Networking*, 15(2):323–333. Volume: 15, Issue: 2 pp. 323-333, ISSN: 1063-6692, San Francisco, CA, USA.

Calì, F.; Conti, M. and Gregori, E. (1998). "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement," in *Proc. IEEE Infocom'98.* Vol. 1, pp: 142 – 149.

Calì, F.; Conti, M. and Gregori, E. (2000). "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Netw.*, Vol. 8, Issue 6, pp. 785–799.

Cantieni, G. R.; Ni, Q.; Barakat, C. and Turletti, T. (2005). "Performance Analysis under Finite Load and Improvements for Multirate 802.11," *Comp. Commun.*, Vol. 28, Issue 10, pp. 1095–1109.

Casetti, C.; Chiasserini, C.-F.; Fiore, M. and Garetto, M. (2005). "Notes on the inefficiency of 802.11e HCCA", in *IEEE Vehicular Technology Conference VTC 2005.* Vol. 4, pp. 2513–2517.

Cena, G. Bertolotti, I.C. Valenzano, A. Zunino, C. (2008). "Industrial applications of IEEE 802.11e WLANs", *IEEE International Workshop on Factory Communication Systems, 2008. WFCS 2008.* .pp. 129-138, ISBN: 978-1-4244-2349-1, Dresden.

Chen, X.; Zhai, H.; Tian, X. and Fang, Y. (2006). "Supporting QoS in IEEE 802.11e Wireless LANs", *IEEE Trans. Wireless Commun.*, Vol. 5, Issue 8, pp. 2217 – 2227.

(a) Cicconetti, C.; Lenzini, L.; Mingozzi, E. and Stea, G (2007). "Design and Performance Analysis of the Real-Time HCCA Scheduler for IEEE 802.11e WLANs". Computer Networks, Vol. 51 , Issue 9, pp. 2311-2325, ISSN:1389-1286, Elsevier North-Holland, Inc. New York, NY, USA.

(b) Cicconetti, C.; Lenzini, L.; Mingozzi, E. and Stea, G (2007). "An efficient cross layer scheduler for multimedia traffic in wireless local area networks with IEEE 802.11e HCCA". ACM Mob. Comput. and Commun. Vol. 11 , Issue 3, pp. 31 – 46, ISSN:1559-1662, New York, NY, USA.

Duffy, K.; Malone, D. and Leith, D. J. (2005). "Modeling the 802.11 Distributed Coordination Function in Non-Saturated Conditions," *IEEE Commun. Lett.*, Vol. 9, Issue 8, pp.715– 717.

Engelstad, P. E. and Osterbo, O. N. (2006). "Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF," in *IEEE International Conference on* Communications, Vol. 2, pp. 552 – 559, ISSN: 8164-9547, ISBN: 1-4244-0355-3, Istanbul.

flexWARE project. Link: http://www.flexware.at/

Foh, C. H. and Zukerman, M. (2002). "A New Technique for Performance Evaluation of Random Access Protocols," in *IEEE International Conference on Communications,* Vol. 4, pp. 2284 – 2288, ISBN: 0-7803-7400-2.

Ghazizadeh, R.; Fan, P. and Pan, Y. (2009). "A Priority Queuing Model for HCF Controlled Channel Access (HCCA) in Wireless LANs," *I. J. Communications, Network and* System Sciences, 2009, Vol. 1, pp. 1-89.

Grilo A., Macedo M., and Nunes M, (2003). "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks", *IEEE Wireless Communications*, Vol. 10, Issue: 3, pp. 36- 43, ISSN: 1536-1284 .

Grow, R. (1982). "A timed-token protocol for local area networks", In I. Electronic Conventions, editor, Proc. Electro/82, number Paper 17/3 in Token Access Protocols, El Segundo, Calif.

(a) Gu, D. and Zhang, J. (2003). "A new measurement-based admission control method for IEEE 802.11 wireless local area Networks," *Mitsubishi Electric Research Laboratory*, Tech. Rep. TR-2003-122.

(b) Gu, D. and Zhang, J. (2003). "QoS Enhancement in IEEE 802.11 Wireless Area Networks," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 120-24. ISSN: 0163-6804.

Hui, J. and Devetsikiotis, M. (2006). "Metamodeling of Wi-Fi Performance," in *Proc. IEEE ICC '06, 2006. ICC '06. IEEE International Conference on Communications,* Vol. 2, pp. 527-534, ISSN: 8164-9547, ISBN: 1-4244-0355-3, Istanbul.

Hui, J. and Devetsikiotis, M. (2004). "Performance Analysis of IEEE 802.11e EDCA by a Unified Model," in *Proc. IEEE Globecom '04,* Vol. 2, pp. 754 – 759, ISBN: 0-7803-8794-5.

Hui, J. and Devetsikiotis, M. (2005). "A Unified Model for the Performance Analysis of IEEE 802.11e EDCA," *IEEE Trans. Commun.* Vol. 53, Issue: 9, pp. 1498-1510, ISSN:0090-6778.

Inan, I.; Keceli, F. and Ayanoglu, E. (2008), "Analysis of the 802.11e Enhanced Distributed Channel Access Function", *CoRR* abs/0704.1833.

Karanam, S. P.; Trsek, H. and Jasperneite, J. (2006). "Potential of the HCCA scheme defined in IEEE802.11e for QoS enabled Industrial Wireless Networks," in *Proc. 2006 IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 227-230, ISBN: 1-4244-0379-0.

Kleinrock L. (1975), "Queueing Systems" . *John Wiley and Sons*.

Kong, Z.; Tsang, D. H. K.; Bensaou, B. and Gao, D. (2004). "Performance Analysis of the IEEE 802.11e Contention-Based Channel Access," *IEEE J. Select. Areas Commun.*, Vol. 22, Issue: 10, pp. 2095–2106, ISSN: 0733-8716.

Lee, W.; Wang, C. and Sohraby, K. (2006). "On Use of Traditional M/G/1 Model for IEEE 802.11 DCF in Unsaturated Traffic Conditions," in *Proc. IEEE WCNC '06*. Vol. 4, pp. 1933-1937, ISSN: 1525-3511, ISBN: 1-4244-0269-7, Las Vegas, NV.

Li, B. and Battiti, R. (2004). "Analysis of the IEEE 802.11 DCF with Service Differentiation Support in Non-Saturation Conditions,", *Quality of Service in the Emerging Networking Panorama,* Volume 3266, ISBN: 978-3-540-23238-4.

Lin, Y. and Wong, V. W. (2006). "Saturation Throughput of IEEE 802.11e EDCA Based on Mean Value Analysis," in *Proc. IEEE WCNC '06*, Vol. 1, pp. 475-480, ISSN:1525-3511, ISBN: 1-4244-0269-7, Las Vegas, NV.

(a) Mangold, S.; Choi, S.; May, P. and Hiertz, G. (2002). "IEEE 802.11e - Fair Resource Sharing Between Overlapping Basic Service Sets," in *Proc. IEEE PIMRC '0,* Vol. 1, pp. 166- 171, ISBN: 0-7803-7589-0.

(b) Mangold, S.; Choi, S.; May, P. and Hiertz, G. and Stibor, L. (2002). "IEEE 802.11e wireless LAN for quality of service,", *Proceedings of the European Wireless*, Vol. 1, pp. 32-39, Florence, Italy.

Moraes, R.; Portugal, P. and Vasques, F. (2006). "Simulation Analysis of the IEEE 802.11e EDCA Protocol for an Industrially-Relevant Real-Time Communication Scenario", *IEEE ETFA'06*, pp. 202-209, ISBN: 0-7803-9758-4, Prague, Czech Republic.

Moraes, R.; Vasques, F.; Portugal, P. and Fonseca, J.A. (2007). "VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks". *IEEE Trans. Industrial Informatics*, Vol.3 ,Issue: 3, pp. 215-224, ISSN: 1551-3203.

(a) Narbutt, M. and Davis, M. (2007). "Experimental tuning of AIFSN and CWmin parameters to prioritize voice over data transmission in 802.11e WLAN networks". *Proceedings of the IWCMC 2007*, pp.140-145, ISBN:978-1-59593-695-0, New York, NY, USA.

(b) Narbutt, M. and Davis, M. (2007). "The capability of the EDCA mechanism to support voice traffic in a mixed voice/data transmission over 802.11e WLANs - an experimental investigation,", *32nd IEEE Conference on Local Computer Networks (LNC 2007)*, pp.463-470. Dublin, Ireland.

Pong, D. and Moors, T. (2003). "Call admission control for IEEE 802.11 contention access mechanism," in *Proc. IEEE GLOBECOM'03*, vol. 1, pp. 174–178, ISBN: 0-7803-7974-8, San Francisco.

Rashid, M. M. and Hossain, E. (2006). "Queuing analysis of 802.11e HCCA with variable bit rate traffic," *IEEE International Conference on Communications*, Vol. 10, pp. 4792 - 4798, ISSN: 8164-9547, ISBN: 1-4244-0355-3 , Istanbul.

(a) Robinson, J. W. and Randhawa, T. S. (2004). "Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function," *IEEE J. Select. Areas Commun.*, Vol. 22, Issue: 5, pp. 917–928, ISSN: 0733-8716.

(b) Robinson, J. W. and Randhawa, T. S. (2004). "A Practical Model for Transmission Delay of IEEE 802.11e Enhanced Distributed Channel Access", *Proc. IEEE PIMRC '04,* Vol. 1, pp. 323- 328, ISBN: 0-7803-8523-3.

Romdhani, L.; Ni, Q. and Turletti, T. (2004). "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks," *Wireless Commun. and Mobile Comp.* Vol. 2, pp. 1373-1378, ISSN: 1525-3511, ISBN: 0-7803-7700-1, New Orleans, LA, USA.

Sawaya, J.; Ghaddar, B.; Khawam, S.; Safa, H.; Artail, H. and Dawy, Z. (2005) "Adaptive Approach for QoS Support in IEEE 802.11e Wireless LAN," *IEEE International Conference on WiMob*, Vol. 2, pp. 167- 173, ISBN: 0-7803-9181-0.

Steigmann, R. and J. Endresen, R. (2006). Introduction to WISA - Wireless Interface for Sensors and Actuators, *White Paper ABB*.

Suzuki, T.; Noguchi, A. and Tasaka, S. (2006). "Effect of TXOP-Bursting and Transmission Error on Application-Level and User-Level QoS in Audio-Video Transmission with 802.11e EDCA," in *Proc. IEEE PIMRC '06*, pp. 1-7, ISBN: 1-4244-0329-4, Helsinki.

Tantra, J. W.; Foh, C. H.; Tinnirello, I. and Bianchi, G. (2006). "Analysis of the IEEE 802.11e EDCA Under Statistical Traffic," in *Proc. IEEE ICC '06*, Vol. 2, pp. 546-551, ISSN: 8164-9547, ISBN: 1-4244-0355-3. Istanbul.

Tao, Z. and Panwar, S. (2004). "An Analytical Model for the IEEE 802.11e Enhanced Distributed Coordination Function," in *Proc. IEEE ICC '04,* Vol. 7, pp. 4111- 4117, ISBN: 0-7803-8533-0.

Tao, Z. and Panwar, S. (2006). "Throughput and Delay Analysis for the IEEE 802.11e Enhanced Distributed Channel Access," *IEEE Trans. Commun.*, Vol. 54, Issue: 4, pp. 596- 603, ISSN: 0090-6778.

Tay, J. C. and Chua, K. C. (2001). "A Capacity Analysis for the IEEE 802.11 MAC Protocol," *Wireless Netw.*, Vol. 7, Issue 2, pp. 159 – 171, ISSN:1022-0038, Kluwer Academic Publishers.

(a) Tickoo, O. and Sikdar, B. (2004). "Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks,", *Proc. IEEE Infocom '04,* Vol. 2, pp. 1404- 1413, ISSN: 0743-166X , ISBN: 0-7803-8355-9.

(b) Tickoo, O. and Sikdar, B. (2004). "A Queueing Model for Finite Load IEEE 802.11 Random Access MAC," in *Proc. IEEE ICC '04,* Vol. 1, pp. 175- 179, ISBN: 0-7803-8533-0.

Tinnirello, I. and Choi, S. (2005). "Efficiency Analysis of Burst Transmissions with Block ACK in Contention-Based 802.11e WLANs," in *Proc. IEEE ICC '05,* Vol. 5, pp. 3455-3460, ISBN: 0-7803-8938-7.

Trsek, H.; Jasperneite, J. and Karanam, S. P. (2006). "A Simulation Case Study of the new IEEE 802.11e HCCA mechanism in Industrial Wireless Networks,", *Proc. 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2006)*, pp. 921-928, ISBN: 0-7803-9758-4 Prague, Czech Republic

(a) Vittorio, S.; Kaczynski, G. and Lo Bello, L. (2007). "Improving the real time capabilities of IEEE 802.11e through a Contention Window Adapter", *RTAS'07_WIP*, pp.64-67, Bellevue, USA.

(b) Vittorio, S. and Lo Bello, L. (2007). "An approach to enhance the QoS support to real-time traffic on IEEE 802.11e networks", Proc. of the 6th Intl Workshop on Real Time Networks (RTN 07), Pisa 2007, http://rtn2007.loria.fr.

Vittorio, S. Toscano, E. Lo Bello, L. (2008). "CWFC: A contention window fuzzy controller for QoS support on IEEE 802.11e EDCA", IEEE International Conference on Emerging Technologies and Factory Automation, 2008. ETFA 2008, pp. 1193-1196, ISBN: 978-1-4244-1505-2, Hamburg.

Viegas Jr., R.; Moraes, R.; Guedes, L. and Vasques, F. (2007) "GSC: A Real-Time Communication Scheme for IEEE 802.11E Industrial Systems". In *Proceedings of the 7th IFAC International Conference on Fieldbus Systems and their Applications (FET-2007)*, pp. 111-118, Nov 7-9, Toulouse, France.

Xiao, Y. (2004). "An Analysis for Differentiated Services in IEEE 802.11 and IEEE 802.11e Wireless LANs," *Proc. IEEE ICDCS '04*, pp. 32 - 39, ISBN:0-7695-2086-3.

Xiao, Y. (2005). "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," *IEEE Trans. Wireless Commun.*,Vol. 4, Issue: 4, pp. 1506- 1515, ISSN: 1536-1276.

(a) Xiao, Y. and Li, H. (2004). "Evaluation of distributed admission control for the IEEE 802.11e EDCA," *IEEE Commun. Mag.*, vol. 42, no. 9, pp. S20–S24, ISSN: 0163-6804.

(b) Xiao, Y. and Li, H. (2004). "Voice and video transmissions with global data parameter control for the IEEE 802.11e Enhance Distributed Channel Access," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 1041–1053, ISSN:1045-9219.

Yang, G.; Choi, J.; Oh, S. and Lee C. (2008). "An efficient scheduling and admission control algorithm for IEEE 802.11e WLAN", *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 12-19, ISBN:978-1-59593-993-7, Suwon, Korea.

Zhang, L. and Zeadally, S. (2004). "HARMONICA: enhanced QoS support with admission control for IEEE 802.11 contention-based access," in *Proc. IEEE RTAS'04*, pp. 64–71, ISSN: 1080-1812, ISBN: 0-7695-2148-7, Toronto, Canada.

10

Wireless Sensor Networks in Industrial Automation

Marko Paavola and Kauko Leiviskä
University of Oulu, Control Engineering Laboratory
Finland

1. Introduction

Wireless sensor networks (WSN) are gaining the ground in all sectors of life; from homes to factories, from traffic control to environmental and habitat monitoring. Monitoring seems to be the key word. Wireless systems can take control actions, too and in this way they compete e.g. with existing process automation systems or with conventional home automation.

WSN consist of nodes. A node in the sensor network includes a microcontroller, data storage, sensor, analogue-to-digital converters (ADC), a data transceiver, controllers that tie the pieces together, and an energy source. The nodes connect to each other using different architectures depending on the applications and surrounding environment. Several architectures, usually called network topologies, are possible: star, cluster-tree and mesh. In different topologies, sensor nodes can act as simple data transmitters and receivers or routers working in a multi-hop fashion (Aakvaag et al., 2005).

Energy is the limiting resource in WSN. A simple microcontroller may operate at 1 mW/10 MHz. When most of the circuits are turned off (in standby/sleep mode), the power consumption is typically about 1 μW. The amount of energy needed to communicate wirelessly increases rapidly with distance and obstructions further attenuate the signal. WSN radios' energy consumption is about 20 mW and their range is typically tens of meters. The network minimizes the energy consumption by eliminating communications or turning off the radio, when communications do not occur. There are several possibilities: local processing of data in nodes, communicating, only if something of interest occurs, data aggregation, compression and scheduling, assigning certain tasks for special nodes and turning off radio, when uninteresting packet is received (Culler et al., 2004).

Recently, the use of WSN in industrial automation has gained attention. The proposed and already employed technologies vary from short-range personal area networks to cellular networks, and in some cases, even global communications via satellite are applied. In industrial environments, the coverage area of WSN as well as the reliability of the data may suffer from noise, co-channel interferences, and other interferers (Low et al., 2005). For example, the signal strength may be severely affected by the reflections from the walls (multi-path propagation) (Werb & Sexton, 2005), interferences from other devices using ISM bands (Low et al, 2005), and by the noise generated from the equipments or heavy

machinery (Werb & Sexton, 2005). In these conditions, it is important to maintain data integrity for operation-critical data, for example alarms (Low et al., 2005). All these factors set a special emphasis on automation design and the fact that WSN are technically challenging systems, requiring expertise from several different disciplines, emphasizes this. Additionally, requirements for industrial applications are often stricter than in other domains, since the system failure may lead to loss of production or even loss of lives. (Low et al., 2005); (Werb & Sexton, 2005).
This Chapter discusses wireless sensor networks in industrial automation, focusing especially on performance issues, both in the design phase and during actual operation. The Chapter will proceed as follows: Section 2 introduces industrial applications. Moreover, a demo system, developed by Control Engineering Laboratory, University of Oulu, is presented as an example. Section 3 concerns the protocols and standards in the industrial WSN. In Section 4, the interferences in industrial environment are discussed briefly. Finally, networked control systems are addressed in Section 5, and a list of references given in Section 6.

2. WSN in Industrial Applications

From industrial point of view, ISA SP100 workgroup introduces six classes (Class 5 – Class 0) for wireless communications based on analysis of industrial, inter-device wireless communication applications (ISA SP100.11, 2006).
Class 5 defines items related to monitoring without immediate operational consequences. This class covers applications without strong timeliness requirements. The reliability requirements may vary. Class 4 defines monitoring with short-term operational consequences. This includes high-limit and low-limit alarms and other information that may require further checking or involvement of a maintenance technician. Timeliness of information in this class is typically low (slow). Class 3 covers open loop control applications, in which an operator, rather than a controller, "closes the loop" between input and output. For example, an operator could take a unit offline, if required. The time horizon for this class is in a human scale, measured in seconds and minutes. Class 2 consists of closed loop supervisory control, and applications usually have long time constants, with the time scale measured in seconds to minutes. Class 1, closed loop regulatory control, includes motor and axis control as well as primary flow and pressure control. The timeliness of information in this class is often critical. Class 0 defines emergency actions related to safety, which are always critical to both personnel and the plant. Most safety functions are, and will be, carried out by dedicated wired networks in order to limit both failure modes and vulnerability to external events or attacks. Examples in this category are safety interlock, emergency shutdown, and fire control. (ISA SP100.11, 2006)
According to survey results (Hoske, 2006), the leading application for industrial networks (both wired and wireless) is supervisory control and data acquisition (SCADA). Next are diagnostics, testing, maintenance; both continuous and batch processing; motion control, robotic equipment; and machine control. Furthermore, the applications include pump, fan, and blower applications; continuous processing; packaging machines; materials handling equipment (elevators, cranes, hoists); and discrete product manufacturing. The most used means of communication are Ethernet TCP/IP, RS232 and 4-20 mA. Ten most used networks, communications and protocols did not include wireless alternatives.

However, applications of wireless technologies will grow especially in following areas (Low et al., 2005):

- Rare event detection
- Periodic data collection
- Real-time data acquisition
- Control
- Industrial mobile robots
- Real-time inventory management

WSN applies for example to bearings of motors, oil pumps, engines, vibration sensors on packing crates, or to many inaccessible or hazardous environments. For these environments, the wired solution may be impractical due to e.g. isolation required for cables running near to high humidity, magnetic field or high vibration environment. Wireless solutions are feasible for mobile applications. (Low et al., 2005)

Compared to wired solutions in industrial applications, the wireless systems and WSN have several advantages. These include, for example (Aakvaag et al., 2005); (Low et al., 2005); Shen et al., 2004); (US Department of Energy, 2007) :

- Flexibility in installing/upgrading network
- Reduced deployment and maintenance costs
- Decentralization of automation functions
- Better coping with regulatory and safety obstacles in running cables in constricted or dangerous areas
- Applicable for moving and rotating equipment
- Improved fault localization and isolation: for example, critical tasks are often ensured with redundant wires, which may pose difficulties for fault location and isolation
- Incorporating short-range technologies to automation system (which has possible interfaces to wide area networks) forms a heterogeneous network, which may improve automation system efficiency
- Exploitation of micro-electromechanical systems (MEMS): integrated wireless sensors with built-in communication capabilities offer a more robust design than attaching wires to small-sized devices.

A small demo system was developed in Control Engineering Laboratory, University of Oulu for testing the performance of WSN in industrial environment. A steam boiler produces steam for a laboratory-scale chemical pulping process. The process uses fuel oil and includes the water storage tank, boiler, and pipelines. The feed water temperature is approximately 20 °C and after the boiler, the steam temperature is approximately 200 °C. There are four measurements implemented: three for the temperature and one for the steam pressure. Fig. 1 shows the measurement locations.

The temperature is measured from the flame in the combustion chamber (the required measuring range from the room temperature to approximately 1500 °C), from the combustion gas pipe (from the room temperature to over 300 °C) and from the surface of the steam pipe (from the room temperature to approximately 300 °C). The pressure, which normally is approximately 13 bar, is measured from the bypass manifold. During shutdowns and maintenance operations, however, the pressure may vary between 0-40 bars. The lower temperatures from the combustion gas pipe and from the surface of the steam pipe are measured by Pt100-sensors. Since the measurements are located closely to each

other, the sensors are attached into one two-channel wireless transceiver node. The higher temperature from the flame in the combustion chamber is measured with the S-type thermocouple and it has its own wireless transceiver node. For the thermocouple, the mains power is required for the transmitter head. Additionally, the pressure sensor has its own wireless transceiver node and requires the mains power. Altogether, the equipment has three nodes and one gateway (Fig. 2). The gateway uses the OPC interface, which passes the measurement information to the LabVIEW™ development system. The test environment has potential sources of disturbances such as thick cement walls, metal pipes, humidity, and varying temperatures.

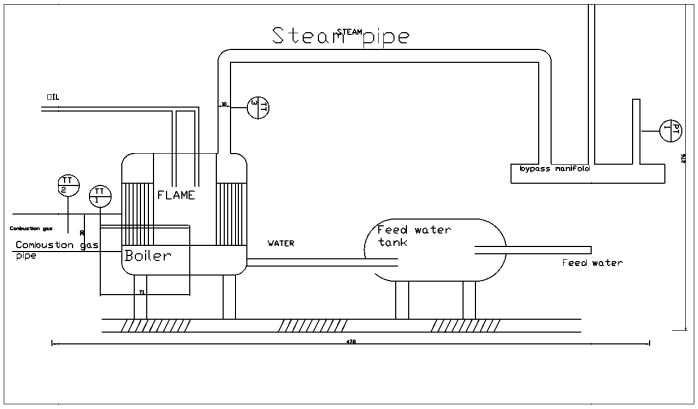


Fig. 1. The steam boiler process with measurements.

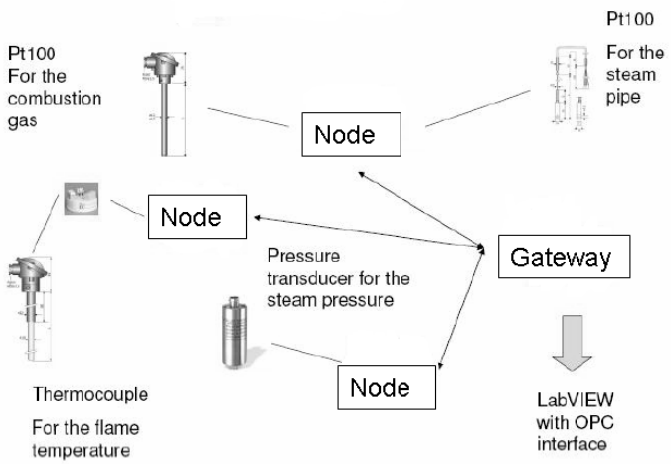


Fig. 2. The WSN and sensors applied for monitoring the temperatures and the pressure of the steam boiler.

Concerning the ISA classes, the steam boiler monitoring application belongs to class 4. For example, it could be used to inform operator about abnormal changes in pressure or temperature. Due to the slow process, the requirements for message timeliness are low, however. Typically, the measurement is expected to arrive within tens of seconds. The performance of the WSN in the presence of interferences is discussed briefly in Section 4 and more detailed in (Paavola & Ruusunen, 2008). More information about the application, for example requirements definition and lessons learned, can be found from (Paavola, 2007).

3. Protocols and standards in the industrial WSN

3.1 Protocols

As mentioned above, the application requirements for the wireless communications in the industrial environments may vary significantly. Taking the demo system presented in this Chapter as an example, the amount of data is little and the acceptable latency within tens of seconds. On the other hand, at the lowest level of the factory automation systems, also a limited amount of data is exchanged, but within very strict real-time constraints, typically 10 ms (Vitturi et al., 2007). These cases provide very different requirements for the WSN protocol stack (see Fig. 3).

In order to introduce radio-based technologies to the industrial automation systems, the automation domain specific requirements have to be fulfilled. These requirements include guarantees for the real-time (RT) behaviour, functional safety, and security (Neumann, 2007). However, the primary objective of the wireless sensor network design has been to maximise the lifetime of the network and nodes, leaving the other performance metrics as secondary objectives (Demirkol et al., 2006). Indeed, many schemes presented in the literature do not concentrate on the joint energy conservation and real-time (RT) performance (Pantazis et al., 2009). It should also be noted, that in some industrial applications, especially in the factory automation domain, the energy consumption may not be critical requirement since mains power is generally available (Flammini et al., 2009).

In this section, the protocols applied in the industrial WSN are discussed, excluding the proprietary protocols (a short introduction to several industrial communication systems as well as to some proprietary protocols can be found in (Neumann, 2007)).

Regarding to industrial WSN protocol development the following requirements can be found from the literature:

- RT, reliable communication, also in heterogeneous networks (Heo et al., 2009)
- Coping with transient interferences: guarantee deterministic and timely data delivery in case of temporary link failures (Song et al., 2006)
- Design, that takes the resource-constrains of the WSN (low processing power, limited energy and small memory) into account (Al-Karaki & Kamal, 2004); (Akyildiz et al., 2002)
- Energy-efficiency: operate at low duty cycles, maximising shutdown intervals between packet exchanges (Rowe et al., 2008))
- Deterministic node lifetime (Rowe et al., 2008)
- Scalability (Rowe et al., 2008)
- Capability for localisation, synchronization and energy management (Flammini et al., 2009)
- Safety and security (not discussed in this paper) (Neumann, 2007)

All these requirements have significant impact on the WSN protocols stack (Fig. 3). In the horizontal planes, the layers of the Open Systems Interconnection (OSI) Reference Model protocol stack, developed by International Organization for Standardization (ISO), are presented. The vertical planes illustrate the modifications required specifically by WSN. In some applications, knowledge of positions, provided by the localisation capability, is required (Flammini et al., 2009). The power manager handles on-board power sources or energy scavenging units (Yeatman, 2007). Finally, to support RT communication, synchronisation capability is needed (Rowe et al., 2008).

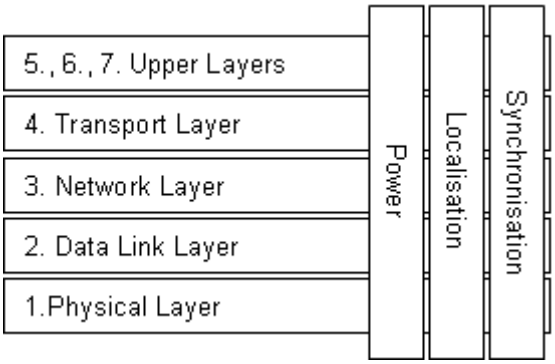


Fig. 3. They layers of the WSN protocol stack. (Flammini et al., 2009)

Concerning the horizontal planes, the tasks of the physical layer include frequency selection, modulation, and data encryption. Since the short-range transceivers are more efficient in terms of the energy consumption and the implementation complexity, their use is preferred. Most widespread commercial solutions available implement spread spectrum modulation techniques and are capable for data rates ranging from 0.1-1 Mbps (Flammini et al., 2009). The common IEEE 802.15.4-standard based radio uses industrial, scientific and medical (ISM) bands, whose selection is dependent on country-specific legislation (see Section 3.2).
The most typical spread spectrum modulation techniques include direct sequence spread spectrum (DSSS) and frequency spread spectrum (FHSSS) (Hu et al., 2008). These have different physical characteristics, and therefore they react differently in industrial settings. In general, FHSS is more suitable for harsh environments due to frequency hopping (Low et al., 2005). Moreover, user can choose not to use certain frequencies, if there is known narrowband interference present (Low et al., 2005). On the other hand, the DSSS can remove the interference completely, if the interfering signal power is within the jamming margin (Low et al., 2005). For more discussion about the modulation regarding the industrial environment, refer to (Low et al. 2005); (Hu et al., 2008).
The data link layer incorporates multiplexing of data streams, data frame detection, medium access control (MAC), and error control. The MAC controls the radio and therefore, it has remarkable impact on the energy consumption and node lifetime (Flammini et al., 2009). The MAC also decides when the nodes access the shared medium and tries to ensure that the competing nodes do not interfere with each other's transmissions. The two main approaches to the sharing of the radio channel are the contention and schedule-based ones. In the former, nodes contend over the resource, and collisions are possible. In the latter, the

transmissions are based on a schedule. The contention-based MAC protocols, such as commonly used CSMA, suffer from overhearing, hidden terminal problem and performance degradation with high contention levels (Wang et al., 2006). In schedule-based protocols, such as commonly applied TDMA, the hidden terminal problem can be handled by scheduling. However the synhronization required, as well as the collisions presents a fundamental challenge (Rowe, et al. 2008). Moreover, the scalability of the network may be worse (Wang et al., 2008.). Additionally, hybrid approaches can be found in the literature (for an industrial example, applied in real-time temperature monitoring, refer to (Flammini et al., 2007).

Several different MAC protocols have been proposed for the WSN in the literature. Discussion and comparisons can found from e.g. (Demirkol et al., 2006); (Rowe et al., 2008); (Pantazis et al., 2009); (Martinez et al., 2007). The focus in WSN MAC protocol development has been in the energy efficiency (Demirkol et al. 2006); (Pantazis et al., 2009). However, some studies (Phua et al., 2009); (Rowe et al., 2008); (Zhou et al., 2008) have addressed the reliability and RT performance of the MAC protocol regarding to the industrial automation domain. (Phua et al., 2006) presented a TDMA-based protocol that uses link state dependent scheduling. In the approach, the node gathers samples of the channel quality and generates prediction slots. The nodes wake up to transmit/receive only during the slots that are predicted to be clear. The proposed approach could improve the reliability of the transmission. (Zhou et al., 2008) proposed an approach of dividing 802.15.4 MAC layer into three services, each of which had additional sub-classes. The division was carried out to meet the RT communication needs of the industrial applications, as presented in ISA classification (see Section 2). The proposed approach could improve the real-time performance of the network. (Rowe et al., 2008) presents an interesting TDMA-based protocol which uses pluggable time-synchronization modules. The hardware-based globally synchronized link protocol could achieve sub-100 μs network synhronization, being still cost-effective and energy efficient. Moreover, the end-to-end latency remained constant in the multi-hop networks.

The network layer is responsible for routing the data from the upper layers of the source nodes to the corresponding layers of the sink node. In case of a single-hop architecture, the source and sink nodes are directly connected. In the multi-hop network, the nodes can forward information not intended for them.

The possible topologies include star (single-hop), mesh (multi-hop) and hybrid (cluster-tree), presented in Fig. 4. The advantage of the star topology is energy efficiency and long lifetime, even if a node collapses. Namely, energy is not consumed on listening to network changes and relaying messages between the nodes, as in case of multi-hop architecture. As a disadvantage of the star topology, smaller number of nodes compared to the multi-hop network is allowed. However, this may not be a problem, if the coordinators use wired links. On the other hand, the multi-hop networks have a longer range and since all the nodes are identical, separate sink nodes are not necessarily needed. However, in addition to the aforementioned energy consumption, the network may suffer from increased latency. The hybrid architecture attempts to combine the low power and simplicity of the star topology as well as the longer range and self-healing of the mesh network. Also in this approach, nonetheless, the latency may still be a problem. (Flammini et al., 2009)

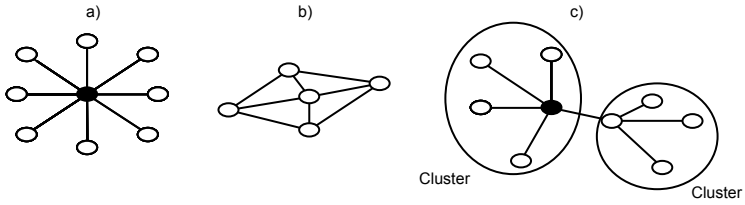


Fig. 4. WSN topologies. Star (a), mesh (b), and hybrid (c). (Flammini et al., 2009)

A comparison of several different network layer protocols from network performance point of view has been presented in (Martinez, et al., 2007). (Heo et al., 2009) discusses several RT routing protocols, concerning especially the industrial applications. They also propose an approach, EARQ that takes into account the RT, reliability and energy efficiency of the communications. EARQ can set the reliability of a packet to manage the trade-off between energy and reliability. Concerning energy awareness, lost packets or packets missing deadlines, the EARQ was reported outperforms other RT protocols discussed in the study. Moreover, it was concluded, that in the practical environments networks are often heterogeneous, compromising of several technologies. Therefore, a protocol ensuring RT also in these operating environments was considered necessary.

The transport layer is usually implemented to provide the end users with an access to WSN through the internet (Flammini et al., 2009). The upper layer is usually combined to a generic application layer, intended to hide the implementation details from the end-user (Flammini et al., 2009). (Vitturi et al., 2009) addresses the importance of the application layer from both the standardisation and performance point of view (Vitturi et al., 2007). In the study, an excellent analysis of the application layer implementation and performance issues using a prototype layer derived from wired fieldbus systems is carried out. It is concluded, that the performance of the implemented approach is worse than expected on the basis of the protocol analysis. According to the authors, the performance degradation is related to several factors: structure of the developed application layer, implementation of the communication standards and software execution times of the components. Moreover, (Vitturi et al., 2007) give a brief introduction to application layer in the industrial communication systems, as well as to the related literature. For more detailed description of the WSN protocol stack in general, refer to (Jiang et al., 2006); (Flammini et al., 2009).

In the classic layered architecture, each protocol layer acts as an independent module with dedicated functions, and handles data packets coming from layer above or below it. The layered architecture is proven to function well in the wired world, but has faced challenges in wireless networks, mainly due to typical characteristics of WSN, such as shared transmission medium, limited resources and lossy communication channels (Zhuang et al., 2007). To overcome these issues, cross-layer design approach (Goldsmith & Wicker, 2002) has been proposed. Cross-layer design allows communications between different protocol layers and the actual functions can be designed jointly. The benefits of the approach include improved efficiency, throughput, and better allocation of resources, lower delay, and more effective energy consumption (Goldsmith & Wicker, 2002). Practical examples of cross-layer design in a real industrial monitoring case can be found in (Franceschinis et al., 2008); (Lu et al., 2008).

In the next section, the industrial automation related standards are shortly discussed. Since some of the standards are already discussed extensively in the literature, only brief introductions with references are given. However, recently published WirelessHART-standard is discussed more closely.

3.2 Standards

The IEEE 802.15.4 (IEEE, 2006) standard defines the protocol and interconnection of devices via radio communication in a low data rate, low power consumption, and low cost personal area network (PAN). The media access is contention-based, applying carrier sensing multiple access with collision avoidance (CSMA/CA) in non-beacon enabled -mode. However, using the optional superframe structure, guaranteed time slots (GTS) can be allocated by the PAN coordinator to devices with time critical data in beacon enabled -mode. Connectivity to higher performance networks is provided through a PAN coordinator. The PHY is defined to for operation in three different ISM frequency bands: 868-868.6 MHz (Europe), 902-928 MHz (North America) and 2400-2483.5 (worldwide). The supported network topologies include star and peer-to-peer. For more detailed description of IEEE 802.15.4, refer to (IEEE, 2006); (Hameed et al., 2008); (Zhuang et al., 2007). (Hameed et al., 2008) also present a performance evaluation and optimisation of IEEE 802.15.4 beacon enabled -mode. Based on the simulation results, the GTS mechanism outperformed the CSMA/CA being able maintain constant MAC delay. Applying the proposed optimisation algorithm, the number of nodes with GTS could be improved.

ZigBee (ZigBee Alliance, Inc., 2008) consists of standard IEEE 802.15.4 (lower layers of the protocol stack) and specifications and profiles defined by the ZigBee specification. A recent study (Pinedo-Frausto & Garcia-Macias, 2008) provides a detailed introduction to ZigBee, as well as extensive performance analysis carried out with a real implementation. Based on the analysis, the authors conclude, that the technology is suitable for applications in ISA usage classes 3 to 5 (see Section 2), but it is not adequate for applications in classes 0 to 2 (emergency actions to closed loop control).

Two recent studies, (Körber & al., 2007); (Lill & Sikora, 2008) lend support to (Pinedo-Frausto & Garcia-Macias, 2008). Namely, they report the inapplicability of the current standard wireless solutions, such as IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4 and ZigBee for hard real-time (5 ms trigger limit) applications. In (Lill & Sikora, 2008) a custom hardware and firmware for communication, synchronisation, and frequency hopping functionalities were designed. Based on the initial results presented, the proposed approach could meet the 5 ms limit. In (Körber et al., 2007), development of a hard real-time sensor actuator is presented extensively, starting from user requirements and ending to a prototype implementation. The proposed approach applies the star network topology, combines frequency division multiple access with TDMA (F/TDMA), and a low-power commercial radio transceiver. The initial results report the trigger limit performance between 6 ms and 11 ms (worst case). (Lill & Sikora, 2008), also mention a commercial alternative capable of meeting strict RT requirements, namely ABB WISA (Scheible et al., 2007). However, as a disadvantage in their case, (Lill & Sikora, 2008) report inapplicability to the battery-powered devices. Moreover, although WISA capable of reaching 10 ms trigger limit and thus suitable for several applications, it was considered unable to reach the extremely tight 5 ms limit.

WirelessHART, developed by HART Communication Foundation, is a wireless interface to widely-adopted HART standard. It aims to address the need for an open standard, which

fulfils the industrial requirements for wireless technology as well as ensures that the customers are not locked to a single supplier. In addition to supporting WirelessHART compatible products from different vendors, the specification is also intended for diverse array of applications, including process monitoring and control, asset management, health safety and environmental monitoring.
A Wireless HART network is formed by a group of network devices. The devices can be either field devices, connected directly to the process plant, or handheld devices. The network supports both star and mesh topology, and therefore, each network device must be able to work as a source, sink or router. A WirelessHART gateway connects the network to the plant. Moreover, a network manager is applied to maintain network status information. Together with the network manager, a security manager is utilised to prevent possible attacks and intrusions. (Kim et al., 2008)
The WirelessHART standard specifies the communication protocol stack using the OSI model, and supports also cross-layer design. The PHY layer of wireless HART is based on IEEE 802.15.4-2006, operating on 2.4 GHz unlicensed band, with maximum data rate of 250 kbps. The modulation applies combination of DSSS and FHSS to provide robust communications against both the broadband and the narrowband interferences. At MAC layer, TDMA is utilised to ensure contention free transmission. Data link layer takes care of sharing of the wireless medium, formatting the data packets as well as correcting bit errors. The responsibilities of the network layer include routing, topology control, end-to-end security and session management. The transport layer ensures end-to-end reliability and flow control. Additionally, block transfer of large data sets is supported. Moreover, a four-level priority classification is supported. (Kim et al, 2008)
The development of the WirelessHART specification is still in progress. For example, the current specification does not consider mobility, interference from time-varying wireless channels, localisation, and effective handover when operator moves from one network /device to another or constant change in topology. For more details about WirelessHART, refer to (Kim et al, 2008); (Lennvall & Svensson, 2008).

4. Interferences in industrial environment

As mentioned earlier, the reliable and real-time communications are required for industrial automation applications. The harsh industrial environment, however, may decrease the network performance due to e.g.:

- multipath propagation: the signal strength may be severely affected by the reflections from the walls (Werb & Sexton, 2007)
- interferences from other devices using ISM bands (Werb & Sexton, 2007)
- noise generated from the equipments or heavy machinery (Low et al., 2005)
- wide operating temperatures, strong vibrations, and airborne contaminants (Low et al., 2005)

It is important to understand the radio channel characteristics in order to predict the communications performance in industrial operating conditions (Low et al., 2005). In this section, the empirical studies on the effect of different interferences on wireless communication performance are surveyed. Especially, the focus is on the research performed in actual IEEE 802.15.4 environment, commonly applied in the WSN.

Based on the literature, the area that has gained the most attention (Bertocco et al., 2007); (Vanheel et al., 2008); (Bertocco et al., 2008a); (Bertocco et al., 2008b); (Toscano et al., 2008) is the co-existence of several wireless communication systems in the same ISM band. (Bertocco et al., 2007); (Bertocco et al., 2008a) and (Bertocco et al., 2008b) concentrate on the performance of the CSMA/CA without interferences and in the presence of interferences. In all these papers, the network under study is based on IEEE 802.15.4 and the performance is evaluated both in cyclic polling and in acyclic alarm task.
In (Bertocco, et al., 2007), a signal generator is applied to produce the interference specified for radiated immunity tests of electromagnetic compatibility (EMC). The interference varying between 300 MHz to 1000 MHz (out of the ISM band) did not cause any significant changes in the WSN behaviour. Moreover, the same apparatus is applied to emulate interference from IEEE 802.15.1 and IEEE 802.11 networks, both continuous and in bursts. In both cases, the performance of the polling task was degraded (in this study, the performance in alarm task was studied only without interference). Especially, when the continuous interference signal exceeds the clear channel assessment (CCA) threshold of the CSMA/CA, the polling task is hindered completely. (Bertocco et al., 2008a) extends the aforementioned study, proposing methods for industrial WSN performance evaluation, and assessing the effect of burst interference on alarm tasks. The interference increased the alarm latencies notably. (Bertocco et al., 2008b) investigates clear channel assessment (CCA) procedures in the presence of IEEE 802.15.1 (Bluetooth), IEEE 802.11g and IEEE 802.15.4 (ZigBee) interference. The network under study is based on IEEE 802.15.4 and the performance is evaluated both in cyclic polling and in acyclic alarm task. The applied metric is PER. The results show that the interference IEEE 802.11g and IEEE 802.15.4 are significant. The interference from Bluetooth, however, is equivalent to "no interference" –situation. Interestingly, best performing CCA procedure seems to be the "no-CCA" in which no channel assessment is carried out at all. Disabling CCA completely resulted to the best resistance against the interference as well as best performance in the cyclic and acyclic tasks.
(Vanheel et al., 2008) study the distance of interference sources (IEEE 802.11b and IEEE 802.11g) on IEEE 802.15.4 (ZigBee) connection. For both the sources, minimum distance causing worst case packet error ratio (PER) of 0.1 on IEEE 802.15.4 link is examined. The measurements agree with the simulation results presented for IEEE 802.11b interferer in the standard IEEE 802.15.4, Annex E: large frequency offsets allow close-proximity co-existence.
(Toscano et al., 2008) study the cross-channel interference in IEEE 802.15.4 –network. The results imply that if the powers of the interfering signal and source node are comparable at the receiver, the interference has only small effect (in this case 4.5% worst case PER). However, when either the power of the interfering signal is significantly stronger or the actual source node is considerably farther from the receiver, the packet loss ratio can increase significantly, especially if unacknowledged communication is used together with high duty-cycle transmissions. Moreover, (Toscano et al., 2008) also assess the sensitivity of the received signal strength indicator (RSSI) in detecting the interferences. Based on results, the RSSI seems to be incapable of detecting cross-channel interference. The performance measurements presented in (Paavola & Ruusunen, 2008)) are in good agreement with the aforementioned studies, concerning interference from the IEEE 802.15.1 and the IEEE 802.11 networks.
In addition to aforementioned studies about the influence of IEEE 802.15.1 and IEEE 802.15.4, (Paavola & Ruusunen, 2008) evaluated some network design parameters, and the

effect of frequency converter -originated interference in the monitoring environment described in Section 2. Especially, the focus was on clarifying, if the communications remained isochronous. For this, variability in latency, jitter, was investigated. Several statistical quantities were compared in analysing the jitter distributions, and a novel approach, entropy, was proposed. The entropy performed well in the jitter analysis, being able to point out statistically factors decreasing isochronous performance. The effect of frequency converter was equal to IEEE 802.11. Based on the study, both of these, however, have a smaller influence than the network design parameters, such as distance from nodes to gateway, number of nodes and sampling interval. Additional information and some references to interference studies can be found from (Paavola & Ruusunen, 2008).

5. Networked control

Networked Control Systems (NCS) are spatially distributed systems consisting of the process to be controlled together with its actuators, sensors, and controllers (Antsaklis & Baillieul, 2007); (Hespanha et al., 2007). The communication between system components takes place via a shared band-limited digital communication network. The systems may also operate in an asynchronous manner, but aim at desired overall objectives. The design of NCS must relay on both control and communication theories. The control theory assumes data transfer through "ideal channels", whereas the communications theory takes the data transfer through "imperfect channels" into account (Hespanha et al., 2007). On the other hand, the communication theory is concerned with reliable transfer of data independent from its usage, where as the control theory is interested in using data in feedback control for some purpose requiring a certain performance (Nair et al., 2007).
Networked control, be it wired or wireless, set some new challenges for the design and performance of the control systems:

- real-time requirement
- band-limited channels
- network delay
- data consistency (packet dropout)
- network architectures
- multipath fading

Real-time behaviour:
Real-time operation is necessary for control functions. In this connection, "real-time" means that the system must be able to response to control requests timely, so that corrections still have their desired effect on process operation. This presumes both the real-time operation system and data transfer; deterministic operation is the most important requirement.
Real-time requirements depend on the application and they can be divided in four categories (Neumann, 2007):

- non-real-time applications in diagnosis, maintenance, commissioning, slow mobile applications
- soft real-time applications: in process and factory automation, mainly in data acquisition and monitoring
- hard real-time applications in process and factory control, fast mobile applications, machine tools
- isochronous hard real-time applications especially motion control.

According to (Neumann, 2007) industrial control problems belong to hard real-time applications. They depend on scheduling of data traffic on top of MAC-layer with the cycle time of 1–10 ms. Most of these wireless radio networks can be used in non-real-time applications, some of them in soft real-time applications, but industrial applications are often outside their capability because of challenging environments and ISM band limit.
Real-time operating systems require the following properties:

- multitasking
- interrupt handling
- task scheduling using priority-based event scheduling and/or time sharing using clock interrupts
- dynamic memory allocation

Band-limited channels:
Any communication channel can carry a finite amount of information per unit of time. In systems with large communication bandwidth, communication and control can be designed independently (Nair et al., 2007). However, recent industrial control networks (both wired and wireless) share a common digital communication network between multiple sensors and actuators. The total data transfer capacity of the network may be large, but each component can utilise only a small portion of it. This may lead to large quantization errors due to the low resolution of the transmitted data that deteriorate the control performance, making even a stable system impossible (Hespanha et al., 2007). The situation is especially difficult in energy-limited systems and in cases, where the number of components is high.
Network delay:
The network delay can be constant, with the time varying or even random, and it occurs when sensors, actuators, controllers and humans exchange data over the network. It depends on the network structure, media and protocol and it is divided into communication and computational delays. In slow process control, delays have only a small effect on the control performance. In fast control loops, the delays impair the performance and can even destabilize the system.
To transmit a continuous-time signal over a network, the signal must be sampled, encoded in a digital format, transmitted over the network, and finally the data decoded at the receiver side (Hespanha et al., 2007). The overall delay includes the network access delays and the transmission delays. They both depend on network conditions such as congestion and channel quality (Hespanha et al., 2007).
Another way of dividing network delays is given in (Mattina & Yliniemi, 2005):

- Waiting time delay is the delay, of which a source has to wait for queuing and network availability before actually sending a frame or a packet
- Frame time delay is the delay during which the source is placing a frame or a packet on the network
- Propagation delay is the delay for a frame or a packet travelling through a physical media

In the networked control loop of Fig. 5, the network delay consists of the following transmission delays

- communication delay between the sensor and the controller, τ_k^{sc}
- communication delay between the controller and the actuator, τ_k^{ca}

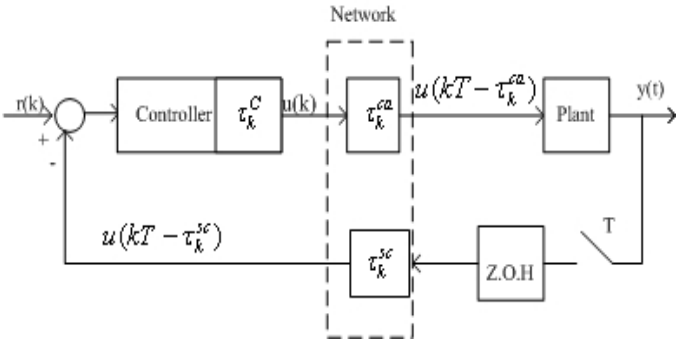


Fig. 5. Block diagram of the network-based control system with different delays (Yliniemi & Leiviskä, 2006).

The total network delay includes also computational delays, e.g. in the controller, but they can be embedded in the above-mentioned delays. In addition to transmission and computational delays, delays may be due to the network load and failures. Errors can occur in data transmission causing retransmission and increasing delay. Collision risks exist, if two nodes send messages at the same time. In cyclic networks based on token passing and TDMA (Time Division Multiple Access) the delay is caused primarily from the waiting time. In Ethernet, CAN (Controller Area Network) and Internet the delay behaves randomly due to the CSMA technology (Yliniemi & Leiviskä, 2006).

Fig. 6 compares experimental process responses with a sample frequency 1 Hz across Ethernet, Internet, FUNET, and without network (Yliniemi & Leiviskä, 2006). As the Figure shows, the control across Ethernet is similar to the control without the network. The control performances across Internet and FUNET are slower than across Ethernet as the responses show. The experiments with the sample frequencies 2 and 5 Hz give the similar results.

In (Yliniemi & Leiviskä, 2006), the ability of different wired networks to the process control was also examined by measuring the time which goes when a measurement signal is sampled to when it is used in the actuator. The sampling frequency in all experiments was 1 Hz. In Ethernet, this time is very small i.e. in average 1 ms. The corresponding time in Internet and FUNET is about 30 ms.

Several methods for compensating the network delay have been proposed. To take the randomness of the network into account, either as a constant probability function or as a Markov chain together with time stamping (Nilsson, 1998). Another application uses clocked buffers on the input of the controller node and in the actuator node to get rid of the time variations (Luck & Ray, 1990). Time-stamped model predictive control (TSMPC) makes possible to get better stability by decreasing the modelling error (Srinivasagipta, 2004). Also using the gain scheduling method for networked PI controller over IP network has been proposed (Tipsuwan & Chow, 2003). A fuzzy compensation method has been developed for a PI controller over the network with randomly varying delay (Almutairi et al., 2001).

Reference (Vatanski et al., 2009) proposes two methods for control over a network. The Smith Predictor-based approach was proposed for the control in the case when accurate delay measurements are accessible, and the robust control-based approach was used when only the estimate of the upper-bound end-to-end delays are available. A switched Ethernet

communication protocol was used to evaluate the performance of the methods for networked control.

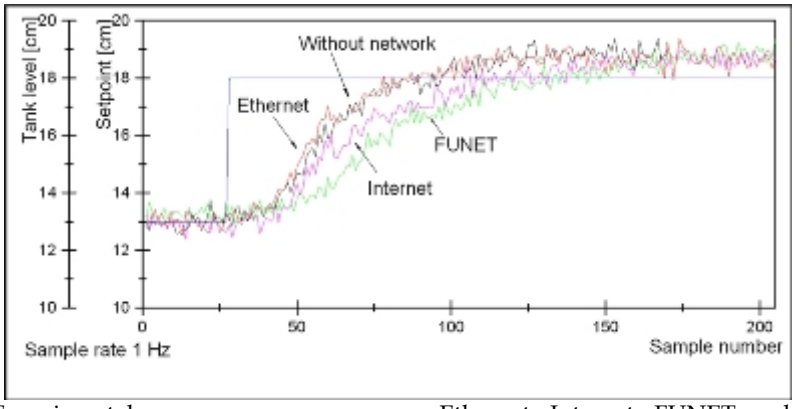


Fig. 6. Experimental process responses across Ethernet, Internet, FUNET and without network (Yliniemi & Leiviskä, 2006).

Data consistency (packet drop-out):
Opposite to standard digital control, in networked control data may be lost while in transferred through the network (Hespanha et al., 2007). Packet drop-outs result from transmission errors in physical network links or from buffer over-flows due to congestion. As mentioned earlier, this may lead to long transmission delays sometimes because of packet re-ordering. There are reliable transmission protocols, such as TCP, that could guarantee the delivery of packets, but they are not so much used in networked control.
Systems architecture:
Fig. 7 (Hespanha et al., 2007) shows the general architecture of an NCS. Encoders map measurements into streams of "symbols" that can be transmitted across the network. Encoders decide when to sample a continuous-time signal and what to send through the network. Decoders convert the streams of symbols into continuous actuation signals used finally for control. Controllers, on the other hand, take care of control calculations.
There are two points of view to architectural questions: network architecture and control architecture. According to (Fammini et al., 2009), the best solution for network architecture is to adopt small, reliable star networks that exploit time division for the network access policy. There are two alternatives for the control hierarchy (Mattina & Yliniemi, 2005): The simplest one is that the remote controller computes the setpoint for the local controller that takes care of the actual control with sensors and actuators. This approach also guarantees the continuity of control in case of bad data communication (Fig. 8 (Mattina & Yliniemi, 2005)). Another approach, also shown in the Fig. 8, is to make all control calculations in the remote controller. This approach is vulnerable for possible disturbances in the control system or data transfer.

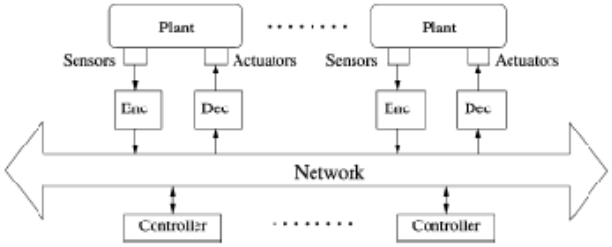


Fig. 7. General system architecture for networked control system. (Hespanha et al., 2007)

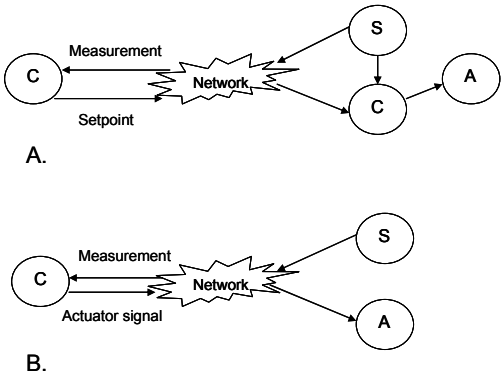


Fig. 8. Two alternatives for the control hierarchy (Mattina & Yliniemi, 2005). C denotes the controller, S the measurement and A the actuator.

6. References

Aakvaag, N.; Mathiesen, M. & Thonet, G. (2005). Timing and power issues in wireless sensor networks – an industrial test case, *Proceedings of the 2005 International Conference on Parallel Processing Workshops*, pp. 419-426, ISBN 0-7695-2381-1, Oslo, Norway, June 2005, IEEE.

Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, C. (2002). Wireless sensor networks: a survey. *Computer Networks*, Vol. 38, No. 4, March 2002, pp. 293-442, ISSN 1389-1286.

Al-Karaki, J. & Kamal, A. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, Vol. 11, No. 6, December 2004, pp. 1536-1284, ISSN 1536-1284.

Almutairi, N.B.; Chow, M.-Y. & Tipsuwan, Y. (2001). Network-based Controlled DC motor with Fuzzy Compensation, *Proceedings of IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1844-1849, ISBN 0-7803-7108-9, Denver, USA, November-December 2001, IEEE.

Antsaklis, P. & Baillieul, J. (2007). Special Issue on Technology of Networked Control Systems. *Proceedings of the IEEE*, Vol. 9, No. 1, January 2007, pp. 5-8, ISSN 0018-9219.

Bertocco, M.; Gamba, G.; Sona, A. & Vitturi, S. (2007). Performance measurements of CSMA/CA-based wireless sensor networks for industrial applications, *Proceedings of the IEEE Instrumentation and Measurement Technology Conference,2007, IMTC 2007*, pp. 1-6, ISBN 1-4244-0588-2, Warsaw, Poland, May 2007, IEEE.

Bertocco, M.; Gamba, G.; Sona, A. & Vitturi, S. (2008a). Experimental characterization of wireless sensor networks for industrial applications. *IEEE Transactions on Instrumentation and Measurement*, Vol. 57, No. 8, August 2008, pp. 1537-1546, ISSN 0018-9456.

Bertocco, M.; Gamba, G. & Sona, A. (2008b). Is CSMA/CA really efficient against interference in a wireless control system? An experimental answer, *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, 2008, ETFA 2008*, pp. 885-892, ISBN 978-1-4244-1505-2, Hamburg, Germany, September 2008, IEEE.

Culler, D.; Estrin, D. & Srivastava, M. (2004). Overview of sensor networks. *Computer*, Vol. 37, No. 8, August 2004, pp. 41-49, ISSN 0018-9162.

Demirkol, I.; Ersoy, C. & Alagöz, F. (2006). MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, Vol. 44, No. 4, April 2006, pp. 115-121, ISSN 0163-6804.

Flammini, A.; Ferrari, P.; Marioli, D.; Sisinni, E. & Taroni, A. (2009). Wired and wireless sensor networks for industrial applications. *Microelectronics Journal*, 2009, *In Press*, ISSN 0026-2692.

Flammini, A.; Marioli, D.; Sisinni E. & Taroni, A.(2007). A real-time wireless sensor network for temperature monitoring, *Proceedings of the IEEE International Symposium on Industrial Electronics, 2007, ISIE 2007,* pp. 1916-1920, ISBN 978-1-4244-0755-2, Vigo, Spain, June 2007, IEEE.

Franceschinis, M.; Spirito, M.A.; Tomasi, R.; Ossini, G. & Pidalà, M. (2008). Using WSN technology for industrial monitoring: a real case, *Proceedings of the Second International Conference on Sensor Technologies and Applications, 2008, SENSORCOMM '08*, pp. 282-287, ISBN 978-0-7695-3330-8, Cap Esterel, France, August 2008, IEEE.

Goldsmith, A. J. & Wicker, S. B. (2002). Design challenges for energy-constrained wireless ad-hoc networks. *IEEE Wireless Communications*, Vol. 9, No. 4, August 2002, pp. 8-27, ISSN 1536-1284, IEEE.

Hameed, M.; Trsek, H.; Graeser, O. & Jasperneite, J. (2008). Performance investigation and optimization of IEEE802.15.4 for industrial wireless sensor networks, *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, 2008, ETFA 2008*, pp. 1016-1022, ISBN 978-1-4244-1505-2, Hamburg, Germany, September 2008, IEEE.

Heo, J.; Hong, J. & Cho, Y. (2009). EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics*, Vol. 5, No. 1, February 2009, pp. 3-11, ISSN 1551-3203.

Hespanha, J.P.; Naghshtabrizi P. & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*. Vol. 95, No. 1, January 2007, pp. 138 – 162, ISSN 0018-9219.

Hoske, M.T. (2006). Industrial networks. *Control Engineering*. January 2006, ISSN 0010-8049. Available: http://www.controleng.com/article/CA6347555.html.

Hu H.; Min, Y.; Xie, X.; Wang, F. & Yuan, J. (2008). Distributed cooperative dynamic spectrum management schemes for industrial wireless sensor networks, *Proceeding of the 2008 Second International Conference on Future Generation Communication and Networking,* pp. 381-386, ISBN 978-0-7695-3431-2, Sanya, China, December 2008, IEEE.

IEEE. (2006). IEEE Standard for information technology, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks. ISBN 0-7381-4996-9, IEEE.

ISA-SP100.11. (2006). Call for Proposal: Wireless for Industrial Process Measurement and Control. Available: http://www.isa.org/filestore/ISASP100_11_CFP_14Jul06_Final.pdf.

Jiang, P.; Ren, H.; Zhang, L.; & Wang, Z. (2006). Reliable application of wireless sensor networks in industrial process control, *Proceedings of the Sixth World Congress on Intelligent Control and Automation, 2006 WCICA, 2006*, pp. 99-103, ISBN 1-4244-0332-4, Dalian, China, June 2006, IEEE.

Kim, A. N.; Hekland, F.; Petersen, S. & Doyle, P. (2008). When HART goes wireless: understanding and implementing the WirelessHART standard, *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, 2008, ETFA 2008*, pp. 899-907, ISBN 978-1-4244-1505-2, Hamburg, Germany, September 2008, IEEE.

Körber, H.-J.; Wattar, H. & Scholl, G. (2007). Modular wireless real-time sensor/actuator network for factory automation applications. *IEEE Transactions on Industrial Informatics*, Vol. 3, No. 2, May 2007, pp. 111-119, ISSN 1551-3203.

Lennvall, T. & Svensson, S. (2008). A comparison of WirelessHART and ZigBee for industrial applications, *Proceedings of the IEEE International Workshop on Factory Communcation Systems, 2008, WFCS 2008*, pp. 85-88, ISBN 978-1-4244-2349-1, Dresden, Germany, May 2008, IEEE.

Lill, D. & Sikora, A. (2008). Wireless technologies for safe automation – insights in protocol development, *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, 2008, ETFA 2008*, pp. 1181-1184, ISBN 978-1-4244-1505-2, Hamburg, Germany, September 2008, IEEE.

Low, K.S.; Win, W.N.N & Meng, J.E. (2005). Wireless sensor networks for industrial environments, *Proceedings of the International Conference on Computational Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies, and Internet Commerce*, pp. 271-276, ISBN 0-7695-2504-0, Vienna, Austria, November 2005, IEEE.

Luck R. & Ray A. (1990). An observer-based compensator for distributed delays. *Automatica*, Vol. 26, No. 5, September 1990, pp. 903–908, ISSN 0005-1098.

Lu, Z.; Wang, Y.; Yang, L. T.; Li, L. & Wang, F. (2008). A reliable routing for industrial wireless sensor networks, *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking,* pp. 325-328, ISBN 978-0-7695-3431-2, Sanya, China, December 2008, IEEE.

Martinez, J.-F.; Corredor, I.; López, L.; Hernández, V. & Dasilva, A. (2007). QoS in wireless sensor networks: survey and approach, *Proceedings of the 2007 Euro American Conference on Telematics and Information Systems,* ISBN 978-1-59593-598-4, Faro, Portugal, May 2007, ACM.

Mattina, V. & Yliniemi, L. (2005). Process control across network. *Report A No. 28*. Control Engineering Laboratory, University of Oulu. ISBN 951-42-7875-5.

Nair, G.N.; Fagnani, F.; Zampieri, S. & Evans, R.J. (2007). Feedback Control under Data Rate Constraints: An Overview. *Proceedings of the IEEE*, Vol. 95, No. 1, January 2007, pp. 108-137. ISSN 0018-9219.

Neumann, P. (2007). Communication in industrial automation – What is going on? *Control Engineering Practice*, Vol. 15, No.11, November 2007, pp. 1332-1347, ISSN 0967-0661.

Nilsson J. (1998). Real-Time Control Systems with Delays. Doctor Thesis, Lund Institute of Technology.

Paavola M. (2007). Wireless technologies in process automation – a review and an application example. *Report A No. 33*. Control Engineering Laboratory, University of Oulu, ISBN 978-951-42-8705-3.

Paavola, M. & Ruusunen, M. (2008). Some factors affecting performance of a wireless sensor network – entropy-based analysis, *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, 2008, ETFA,* pp. 664-671, ISBN 978-1-4244-1505-2, Hamburg, Germany, September 2008, IEEE.

Pantazis, N.A.; Vergados, D. J.; Vergados, D. D. & Douligeris, C. (2009). Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling. *Ad Hoc Networks*, Vol. 7, No. 2, March 2009, pp. 322-343, ISSN 1570-8705.

Pinedo-Frausto E. D. & Garcia-Macias, A. J. (2008). An experimental analysis of ZigBee Networks, *Proceedings of the 33rd IEEE Conference on Local Computer Networks, 2008,* pp. 723-729, ISBN 978-1-4244-2412-2, Quebec, Canada, October 2008, IEEE.

Phua, V.; Datta, A. & Cardell-Oliver, R. (2006). A TDMA-based MAC protocol for industrial wireless sensor network applications using link state dependent scheduling, *Proceedings of the IEEE Global Telecommunication Conference, 2006, GlobeCom '06*, pp. 1-6, ISBN 1-4244-0356-1, San Fransisco, USA, December 2006, IEEE.

Rowe, A.; Mangharam, R. & Rajkumar, R. (2008). RT-link: a global time-synchronized link protocol for sensor networks. *Ad Hoc Networks*, Vol. 6, No. 8, November 2008, pp. 1201-1220, ISSN 1570-8705.

Scheible, G.; Dzung, D.; Endresen, J. & Frey, J.-E. (2007). Unplugged but connected – design and implementation of a truly wireless real-time sensor/actuator interface. *IEEE Industrial Electronics Magazine*, Vol. 1, No. 2, 2007, pp. 25-34, ISSN 1932-4529.

Shen, X.; Wang, Z. & Y. Sun, Y. (2004). Wireless sensor networks for industrial applications, *Proceedings Fifth World Congress on Intelligent Control and Automation*, pp. 3636-3640, ISBN 0-7803-8273-0, Hangzhou, China, June 2004, IEEE.

Song, J.; Mok A. K., Chen, D. & Nixon, M. (2006). Using real-time logic synthesis tool to achieve process control over wireless sensor networks, *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*, pp. 420-426, ISBN 0-7695-2676-4, Sydney, Australia, August 2006, IEEE.

Srinivasagipta, D.; Schättler, H. & B. Joseph. (2004). Time-stamped model predictive control: an algorithm for control of processes with random delays. *Computers and Chemical Engineering*, Vol. 28, No. 8, July 2004, pp. 1337- 1346, ISSN 0098-1354.

Tipsuwan, Y. & Chow, M.-Y. (2003). On the Gain Scheduling for Networked PI Controller Over IP Network. I*EEE/ASME Transactions on Mechatronics*, Vol. 9, No. 3, September 2004, pp. 491-498, ISSN 1083-4435.

Toscano, E. & Lo Bello, L. (2008). Cross-channel interference in IEEE 80215.4 networks, *Proceedings of the IEEE International Workshop on Factory Communication Systems*, 2008, WFCS 2008, pp. 139-148, ISBN 978-1-4244-2349-1, Dresden, Germany, May 2008, IEEE.

US Department of Energy. (2002). Industrial Wireless Technology for the 21st century. December 2002. Available: http://www.energetics.com/rep_products.asp?Product=90.

Vanheel, F.; Verhaevert, J. & Moerman, I. (2008). Study on distance of interference sources on wireless sensor network, *Proceedings of the 38th European Microwave Conference, 2008, EuMC 2008*, pp. 175-178, ISBN 978-2-87487-006-4, Amsterdam, Netherlands, October 2008, IEEE.

Vatanski, N.; Georges, J.P.; Aubrun, C.; Rondeau, E. & Jämsä-Jounela, S.L. (2009). Networked control with delay measurement and estimation. *Control Engineering Practice*, Vol. 17, No. 2, pp. 231–244, February 2009. ISSN 0967-0661.

Vitturi, S.; Carreras, I.; Miorandi, D.; Schenato, L. & Sona, A. (2007). Experimental evaluation of an industrial application layer protocol over wireless systems. *IEEE Transactions on Industrial Informatics*, Vol. 3, No. 4, November 2007, pp. 275-288, ISSN 1551-3203.

Wang, W.; Wang H.; Peng, D. & Sharif, H. (2006). An energy-efficient pre-schedule scheme for hybrid CSMA/TDMA MAC in wireless sensor networks, *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems, 2006, ICCS 2006,* pp. 1-5, ISBN 1-4244-0411-8, Singapore, November 2006, IEEE.

Werb, J. & Sexton, D. (2005). Improved Quality of Service in IEEE 802.15.4 mesh networks, *International Workshop on Wireless and Industrial Automation*, pp. 1-6, San Francisco, California, March 2005.

Yeatman, E. M. (2007). Energy scavenging for wireless sensor nodes. (2007), *Proceedings of 2nd International Workshop on Advances in Sensors and Interface, IWASI, 2007,* pp. 1-4, ISBN 978-1-4244-1245-7, Bari, Italy, June 2007, IEEE.

Yliniemi, L. & Leiviskä, K. (2006). Process control across wired network, *IASTED International Conference on Parallel and Distributed Computing and Networks*, Innsbruck, Austria, February 2006.

ZigBee Alliance, Inc. (2008). ZigBee Specification, January 17, 2008. ZigBee Standards Organization.

Zhou, Y.; Wang, Y.; Ma, J. & Wang, F. (2008). A low-latency GTS strategy in IEEE802.15.4 for industrial applications, *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking,* pp. 411-414, ISBN 978-0-7695-3431-2, Sanya, China, December 2008, IEEE.

Zhuang.; Goh, K. M. & Zhang, J. B. (2007). The wireless sensor networks for factory automation: issues and challenges, *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, 2007, ETFA,* pp. 141-148, ISBN 978-1-4244-0825-2, Patras, Greece, September 2007, IEEE.

11

Analysis of switched Ethernet for real-time transmission

Joan Vila-Carbó, Joaquim Tur-Massanet
and Enrique Hernández-Orallo
Universidad Politécnica de Valencia
Spain

1. Introduction

There has been a growing trend by industrial automation manufacturers of using Ethernet and IP networks at all levels of a factory, thereby replacing fieldbuses and other networks that were typically used in process control. All these networks have been also replaced by AFDX (Avionics Full-Duplex Switched Ethernet) in avionics.
Thus, there is growing interest in improving some features of Ethernet that have been pointed out as important drawbacks for its use in automation and real-time control, such as time-predictability and fault-tolerance. To this aim, industrial network manufacturers have released a new generation of managed switches that allow features like Quality of Service (QoS), configurable topologies such as rings with redundant paths to support loss of connectivity or hardware failures, and virtual private networks support (VPN) to enhance security and network isolation. In this paper we only focus on time-predictability issues.
A number of improvements to Ethernet temporal behaviour have been proposed in the literature. Most of them fall under three possible approaches: suppressing the collisions, reducing their number, and resolving them in a deterministic manner. An example of the first approach is using switches or changing the MAC. Reducing collisions can be done by means of bandwidth reservations. An example of deterministic collision resolution is the CSMA/DCR protocol (Le Lann, 1983). Although this classification is conceptually simple, other classifications that take into account technological issues have also been proposed in the bibliography.
The survey by (Decotignie, 2005) classifies solutions according to their degree of compatibility with standard Ethernet. According to this, solutions can be classified into incompatible, non-interoperable, interoperable homogeneus, and interoperable homogeneus. *Incompatible solutions* include solutions that modify the MAC protocol and cannot be implemented without modifying the Ethernet hardware. They are further classified in (Kurose et al., 1984). Some of these protocols have even been integrated into Ethernet chips (i.e., Intel 82596). An example could be the 100VG-AnyLAN, which is a real-time Ethernet evolution that eventually became standard as IEEE 802.12. *Non-interoperable solutions* include all the solutions that alter the protocol in such a way that a node that is compliant with the new protocol cannot operate in the presence of network nodes that do

not implement the alterations. Examples of this solution are TDMA-based protocols (Pedereiras, 2005) and EPL (Ethernet Powerlink), an open-source solution that is CANopen compatible. EPL can be implemented either on standard or modified hardware. *Interoperable homogeneus solutions* group all solutions that may coexist with standard products. Most of them offer guarantees under the assumption that all devices use the same modifications. They lose their temporal guarantees in the presence of unmodified (IEEE 802.3 compliant) nodes. Examples are REther (Venkatramani & Chiueh, 1994) and solutions that are based on traffic-smoothing. *Interoperable heterogeneus solutions* include those proposals that offer guarantees even in the presence of Ethernet nodes that do not implement the same modifications. Examples are switches with policing features (like EtheReal (Varadarajan, 2001)), or switches with prioritization features (like the ones defined in the IEEE 802.3 extension 801.3p).
The survey by (Felser, 2005) presents another classification of real-time Ethernet solutions according to the layer where modifications are introduced. This classification is illustrated in Fig. 1. The first approach is to concentrate all real-time modifications on top of TCP/IP. In a second approach, the TCP/UDP/IP protocols are bypassed, and the Ethernet functionality is accessed directly on top of Ethernet. Finally, in the third approach, the Ethernet mechanism and infrastructure itself is modified.

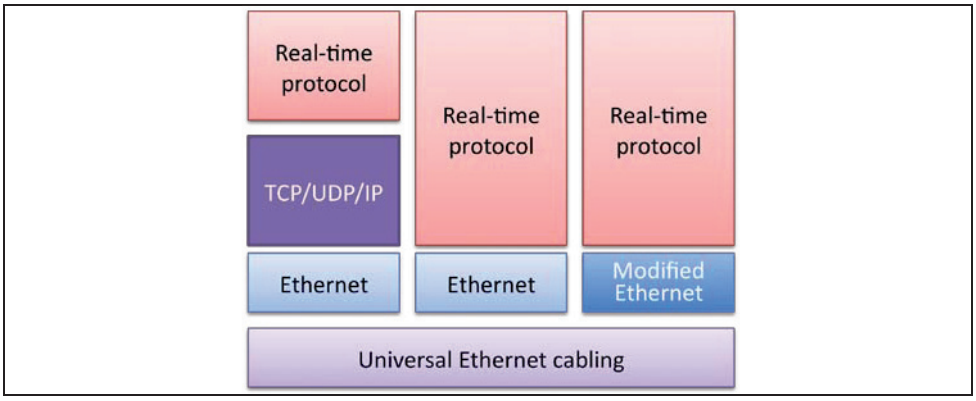


Fig. 1. Structure of real-time Ethernet solutions.

Examples of *solutions on top of TCP/IP* are: the Modbus/TCP standard with its real-time extension of the Real-Time Publisher Subscriber (RTPS) protocol; the EtherNet/IP protocol, that provides real-time communication based on priorities; or the P-NET on IP specification, which allows clients to access servers on IP networks. The *solutions on top of Ethernet* are typically those that impose some kind of time-slicing or time-slotting mechanism in the MAC layer. Examples are Ethernet Powerlink (EPL), TCnet (Time-critical Control Network), PA (Ethernet for Plant Automation) and Profinet. The *solutions that use modified Ethernet* equipment are those solutions that try to provide the typical topologies of factory automation (bus or ring topologies using Switched Ethernet, which is star oriented). This requires a switch in every connected device. The switching functionality is usually integrated inside the field device. The modifications are mandatory for all devices inside the real-time segment, but they allow non-RTE traffic to be transmitted without

modifications. Some examples are SERCOS (SErial Real-time COmmunication System Interface) (Schemm, E. 2004), EtherCAT, and Profinet IO. All the solutions referred by (Felser, 2005) became "IEC proposals for a publicly available specification for real-time Ethernet". They are briefly summarized in the referred survey and detailed in IEC documents (IEC, 2004).
This work focuses on the evaluation of what (Decotignie, 2005) calls *interoperable homogeneus solutions* and, more specifically, solutions based on hardware or software mechanisms that modify the MAC level to make it deterministic and allow the use of the TCP/IP on top of it. Examples are AFDX switches or industrial managed switches with traffic-control features. We also assume that some traffic-control features, like Linux Traffic Control (TC), are available in end-systems.
There are two main approaches for evaluating the considered solutions: *deterministic* techniques, which are based on determining analytical upper bounds; or *statistical* methods, whose goal is to obtain their probability distributions. We take both approaches into account in this work. The chapter is structured as follows: section 2 presents a taxonomy of the main solutions at the MAC level and also provides some analytical deterministic performance bounds, and sections 3 and 4 present the scenarios and the statiscal results of simulations using an industrial managed switch and Linux TC. Finally, section 5 outlines some conclusions.

2. Making Switched Ethernet predictable

The goal of this section is to present the main approaches for making Switched Ethernet deterministic by modifying the MAC level protocol. It also provides *deterministic bounds* for the performance of the proposed mechanisms, and identifies the key issues for performance.
Several methods have been proposed in the literature for obtaining deterministic bounds on network performance. The most widely used is the *Network calculus* theory (LeBoudec & Thiran, 2002). *Schedulability analysis* (Song et al., 2002) (Yiming & Eisaka, 2002) is also a well-known method, but the analysis is usually restricted to periodical flows using rate-monotonic (RM) or deadline algorithms (EDF). Finally, *Model checking* provides another method based on the use of timed automata for describing system temporal behavior. It requires a complete model of the system (which is not always available) and appropriate tools (such as UPPAAL) for evaluating it (Charara et al., 2006).
This section introduces three approaches for making Switch Ethernet predictable at the MAC level and provides their deterministic performance bounds. These are obtained analytically using the *Network Calculus* theory. The approaches considered are:

- *Traffic shaping*: based on bounding the traffic sent to an Ethernet switch.
- *Classes of Service*: based on traffic prioritization.
- *Virtual links*: based on bandwidth reservation.

Some of these methods may require the use of special switches. We refer to a *managed switch* as a switch that allows each of its individual ports to be controlled. Features vary with manufacturers and models. In general, they offer guarantees only if all network interfaces and switches allow the special features, and they may lose their temporal guarantees in the presence of unmodified nodes or switches.

2.1 Traffic shaping

Traffic shaping is a way to improve the predictability of Switched Ethernet that is based on bounding the traffic sent to the network. If all nodes limit their traffic, then it can be shown that the switch delay is bounded. A switch with *policing features* is a switch where non-conforming packets are delayed or eventually dropped. Since most Ethernet switches do not include them as built-in features, nodes need to be cooperative and avoid producing a traffic that exceeds the bounding specifications.

Traffic is usually shaped using a *token-bucket* (or leaky-bucket) specification. A token-bucket shaper with parameters *(r,b)* bounds the traffic to a long-term average rate *r* and a maximum burst size *b*. Practical implementations are usually based on defining a shaping interval T_s. If a shaper is flooded with packets, it transmits a block of size $s \leq T_s{*}r$ in n bursts of length $\leq b$ before the end of T_s. Token-bucket shaping is used for Variable Bit Rate (VBR) and non-strictly periodic traffic, like video. However, if all the n bursts are equally sized, the token-bucket shaping degenerates into strictly periodic traffic scheduling (with period T_s/n), which is really a particular case of traffic shaping.

The delay of an Ethernet switch using traffic shaping is bounded and can be calculated using Network Calculus. This method can be applied whenever the traffic is limited by an arrival curve. Using a token-bucket traffic specification, the arrival curve at the receiving port k of a switch is bounded by function $\alpha_k(t)=r_k{*}t+b_k$.

(Loeser & Haertig, 2004) showed that if the sum of the long-term average input rates for a switch transmit port does not exceed the network bandwidth C:

$$\sum_{k=1}^{N} r_k \leq C$$

then, the Ethernet Switch delay d and the buffer size B are bounded, respectively, by:

$$d = \sum_{k=1}^{N} \frac{b_k}{C} + t_{mux} \qquad B = \sum_{k=1}^{N} b_k + C \cdot t_{mux} \tag{1}$$

where t_{mux} is a fixed value provided by the vendor that includes the switching latency and the frame forwarding time.

These bounds were confirmed by several experiments performed using different commercial switches and operating systems (real-time DROPS and Linux using TC). The experiments showed that if the burst factor is low, the switch delay is reduced.

2.2 Classes of service

The concept of *Classes of Service* (CoS) is the basis for the Differentiated Services (DiffServ) architecture (Black et al., 1998). It is intended to allow network applications to share a network under different types of performance guarantees. For example, low-bandwidth sensors (like pressure sensors) and high-bandwidth sensors (like video cameras) produce highly different workloads, so they may also require different CoS with different performance guarantees. The concept of CoS allows the requirements of both traffic types to be jointly fulfilled.

A typical configuration of CoS would be the following:

- *Real-Time CoS*: suitable for low-bandwidth sensors. It would have to provide deterministic, hard real-time guarantees, like bounded deadlines and jitter.
- *Streaming CoS*: intended for video and high-bandwidth sensors and VBR streams. It would have to provide statistical guarantees, like average transmission rates.
- *Best-Effort CoS*: used by the rest of the traffic. No guarantees are provided, although some minimum bandwidth is always allocated.

CoS are accomplished through *traffic marking* and *prioritization,* so different CoS are assigned different marks and scheduling priorities.
Traffic marking can be done in several ways. For example, the managed switch used in this work (Sixnet SL-5MS) supports two mechanisms: the IEEE 802.1p tag priority field, that may be set from 0 to 7; or the ToS (Type of Service) priority field in the IP header, that may be set from 0 to 255. The switch provides four different priorities: urgent, expedited, normal, and background. All the priority tags need to be mapped to these four priorities.
Traffic prioritization can be done using two main scheduling policies: *non-preemptive priorities* or *fair scheduling*. The *non-preemptive priority* policy assures the lowest latency for high-priority data. With this policy, all packets in a higher priority queue are always sent before packets in a lower priority queue. This is done in a non-preemptive fashion, since the transmission of Ethernet frames cannot be interrupted once it has started. With *fair scheduling*, a weighted round-robin algorithm is used. This way, more high priority than low-priority packets get through, but low-priority packets are always assured some bandwidth. Unlike strict priorities, this policy avoids starvation. The Sixnet SL-5MS allows four weights: 1,2,4,8.
In Linux-based end-systems, traffic prioritization can be performed using Linux TC (Traffic Control), which is the Linux kernel mechanism that determines the way in which packets are sent. It offers a large set of functionality for multilevel traffic scheduling. Each traffic class can be assigned a different queuing discipline (Qdisc). There are two types of queuing disciplines: *classless* and *classful*. Classless Qdiscs only use one level of queuing, while Classful Qdiscs may have several. FIFO is the simplest classless Qdisc, but priority (PRIO) or token bucket-policies (TB) are also available.
Hierarchical Token-Bucket (HTB) policies are used in Linux TC to manage *spare capacity*. It is an adaptive mechanism for dynamically redistributing the unused bandwidth based on the concept of *token borrowing*: some given CoS, for example the Streaming CoS, may have children classes used for transmitting different streams. If a child class is sending at a rate below some given ceil-rate and the parent CoS has spare capacity, some other children classes requiring extra bandwidth may attempt to borrow tokens from the parent class in order to increase their transmission rates.
For switches with prioritization, a bound similar to the one presented in the previous section was obtained by (Zhang & Zhang, 2005). This paper assumes that the switch is able to discriminate frames, using the IEEE 802.1p traffic-marking feature, and the traffic is bounded by a token-bucket specification with parameters *(b,r)*.
The switch delay for traffic with priority level *k* is:

$$d_k = \frac{MTU + \sum_{j=1}^{k} b_j}{C - \sum_{j=1}^{k-1} r_j} + t_{mux} \quad (2)$$

where $j=1\ldots k$ are the priority levels that are higher than k, and MTU is the maximum transmission unit.

When a traffic flow crosses a switch with a delay d, the token-bucket parameters are modified as follows: $(r',b')=(r,b+rd)$. Thus, crossing several switches implies an increase of the burstiness b of the flow.

If a flow crosses m switches with delays $d^1,d^2,\ldots,d^m$ and if (r^{i-1},b^{i-1}) and (r^i,b^i) are the input and output flows at switch i respectively, then the end-to-end delay can be obtained by summing all the switch delays:

$$D = \sum_{i=0}^{m} d^i = \frac{b^m - b^0}{r} \quad (3)$$

A key issue when prioritizing traffic is the effect of *limited preemption*. Our experiments show that it is one of the key issues for the variability of transmission times. This effect can be more or less serious depending on the layer where packet prioritization is performed. The best situation is when packet scheduling is performed at the data-link layer, as occurs in most switches. In this case, the maximum non-preemptable data unit is the size of the MTU of an Ethernet frame, which is typically 1500 bytes. With a 100 Mbps Ethernet, this may cause a maximum delay of up to 120 μs. However, the prioritization performed by Linux TC occurs at the network level (IP level), so it schedules IP packets in a non-preemptive way. This means that the size of the IP packet has a strong impact on latencies. IP packet sizes are fixed by the upper layer protocols: UDP and TCP. When using UDP, the RFC791 recommends that hosts only send datagrams that are larger than 576 octets if they have assurance that the destination is prepared to accept them. However, in the end, IP packet sizes are fixed by the applications since UDP does not perform packet fragmentation. The maximum allowable size is 64 Kbytes. Such a packet might cause a delay of up to 5243 μs with a 100 Mbps Ethernet. Fortunately, this is not the usual case. For TCP, there is a handshake when opening a connection where two parties can agree on a MTU, using the TCP MSS option. The IP datagram size should match the minimum frame size of the underlying networks involved in a connection (for example, 1500 bytes in Ethernet and around 4470 in FDDI).

2.3 Virtual links

A *virtual link* (VL) is an abstraction for the partitioning of a network into multiple virtual instances that emulate isolated point-to-point networks. The 100 Mbps link of an end-system can support multiple virtual links. These virtual links share the 100 Mbps bandwidth of the physical link. VLs provide real-time performance guarantees in terms of bounded transmission delay and jitter. That is accomplished using the principle of *bandwidth reservation*.

One of the most powerful implementations of this concept can be found in AFDX (Avionics Full-DupleX switched Ethernet), which is Part 7 of the ARINC 664 Specification for the

exchange of data between Avionics Subsystems (Charara et al., 2006). AFDX is based on the full-duplex switched Ethernet solution. It is redundant, so each end-system is connected to two redundant networks. VLs in AFDX connect end-systems in a unidirectional way. This connection can be 1 source to N destinations. Switches use the VL identification to route the frames statically. Each VL in AFDX is defined using two parameters: the Bandwidth Allocation Gap (BAG) and the largest Ethernet frame (in bytes) that can be transmitted on the VL (*Lmax*). The BAG represents the minimum interval in milliseconds between Ethernet frames. It must be selected from a set of only 8 values *(2^n ms; n=0…7)*. For example, if a VL has a BAG of 32 ms, then Ethernet packets on that VL are never sent faster than one packet every 32 ms. If the VL has an *Lmax* of 200 bytes, then the maximum bandwidth on that VL is 50,000 bits per second (200* 8*1000/32).

The choice of BAG for a particular VL depends on the requirements of the AFDX ports that are being provided link-level transport by the VL. For example, suppose an avionics subsystem is sending messages on three AFDX communications ports that are being carried by the same VL. Let's assume the message frequencies on the ports are 10 Hz, 20 Hz, and 40 Hz, respectively. The total frequency of the combined messages (which are combined into the same VL) is 70 Hz. The average period of the message transmissions is 14.4 ms. Accordingly, to provide adequate bandwidth on the VL, a BAG that is less than 14.4 ms should be selected. The first available bag is 8 ms (125 Hz).

Fig. 2 shows the structure of the VL scheduler of AFDX with two main components: the packet *regulator* and the *multiplexor*.

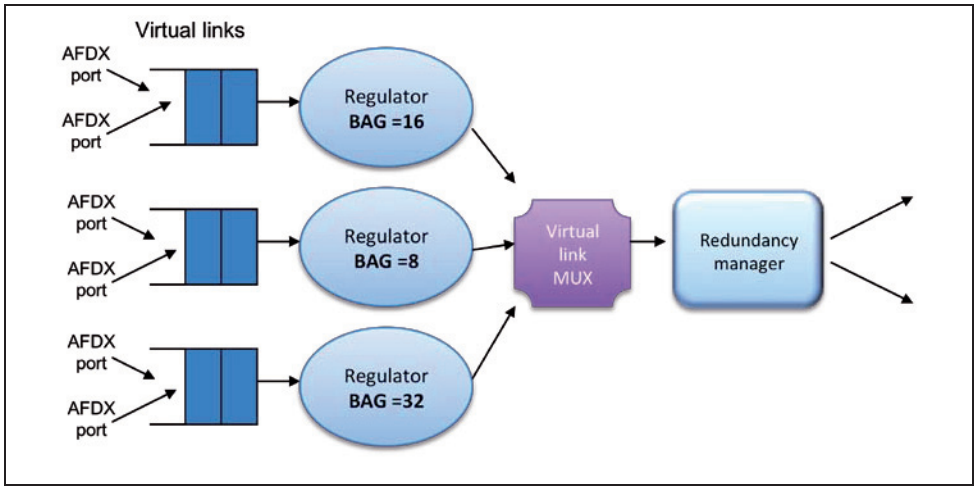


Fig. 2. Structure of the AFDX packet scheduler AFDX.

The traffic regulator performs as shown in Fig. 3. It has three main functions:

- *Traffic-shaping*: traffic is limited and frames are delayed to the next available BAG.
- *Traffic-policing*: frames that do not adjust to the specifications of the VL are dropped.
- *Traffic-filtering*: frames that do not belong to a VL are eliminated (the regulator acts as a firewall).

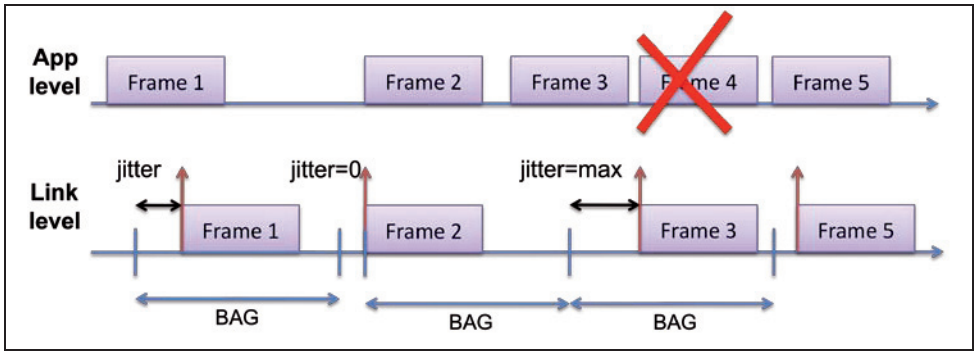


Fig. 3. Traffic regulation and jitter in AFDX.

The source end-system is required to enforce BAG restrictions for each outgoing virtual link. A number of VL scheduling algorithms can be used by the end-system.
Jitter is introduced when the regulator outputs are combined by the multiplexer (MUX); Ethernet frames arriving at input to the MUX at the same time will experience queuing delay. *Jitter* is defined as the interval from the beginning of the BAG to the first sent bit of the frame being transmitted at the virtual link maximum allocated bandwidth. The AFDX standard specifies a bound on this latency:

$$max\_jitter \leq min\left(500\mu s, \sum_{k\in\{VLs\}} \frac{(20 + Lmax_k) \times 8}{C} + 40\mu s\right) \quad (4)$$

where C is the link bandwidth, and 20 corresponds to the number of bytes for the preamble of the MAC frames. Note that the second term of the minimum represents the sender delay (d_{snd}) and it can be obtained from equation (1) by substituting b_k with $(20+Lmax_k)*8$ and the multiplexing delay t_{mux} with 40 µs.
The standard does not specify a maximum end-to-end delay bound, but it can be calculated using network calculus, as shown below. The end-to-end delay can be decomposed into the following delays:

- d_{snd}: sender delay. This is the delay between the call to the API sending function until the frame starts being transmitted to the output port. It is caused by the communication stack and VLs multiplexing. It is calculated using equation (1).
- d_{net}: network delay. It is composed mainly of two factors: the switch delay d_{sw} and the propagation delay d_{pro}, which is usually negligible. The switch delay is the sum of the delays at all the switches that a VL traverses. It mostly depends on the number and characteristics of VLs that share an output port and it can be calculated using equation (1) in a way similar to the sender delay d_{snd}. In a *store-and-forward* switch the time for buffering a complete packet would have to be added to d_{sw}. It is about 40 µs for a 500 bytes packet in a 100 Mps Ethernet, but it can be obviated in modern switches.
- d_{rcv}: receiver delay. This is usually a constant value.

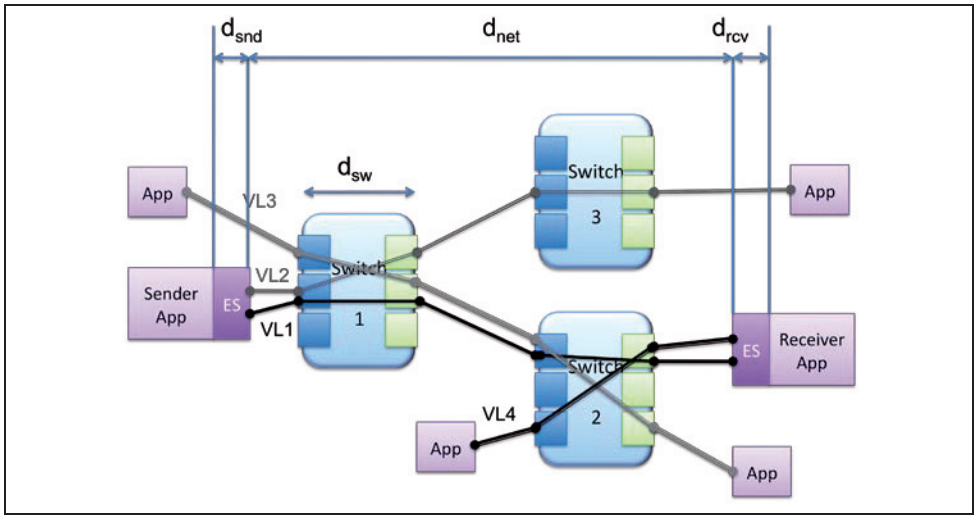


Fig. 4. AFDX sample network.

As an example, consider the network of Fig. 4 with four VLs and three switches. The VLs have the following parameters: VL_1($Lmax_1$=200, BAG_1=2ms); VL_2($Lmax_2$=400, BAG_2=16ms); VL_3($Lmax_3$=500, BAG_3=32ms); VL_4($Lmax_2$=50, BAG_2=4ms). Assume that t_{mux} = 10 μs in the switches, and d_{rcv} = 40 μs.

The end-to-end delay for VL_1 is composed by the sender delay d_{snd}, the sum of delays at switch 1 and switch 2 (d_{sw1} + d_{sw2}), the propagation delays d_{pro}, and the receiver delay d_{rcv}. To obtain the switch delays, the calculations must take into account the VLs that share some given output port with VL_1: VL_3 for d_{sw1}, and VL_4 for d_{sw2}. The sender delay d_{snd} can be calculated analogously, but now VL_1 shares the output link with VL_2. In summary, the end-to-end delay is bounded and is:

$$d = d_{snd} + d_{sw1} + d_{sw2} + d_{rcv} = 88 + 66 + 30 + 40 = 224\mu s$$

3. Evaluation of Switched Ethernet

The goal of this section is to evaluate the effectiveness of the implementations of the concepts of classes of service (CoS) and traffic-shaping on top of Switched Ethernet and Linux TC. This evaluation is based on statistics of real measurements. The experiments evaluate how a high-priority *real-time workload* is affected by a low-priority *background workload*.

3.1 Background

Before presenting the evaluation, we briefly review the main approaches for the *statistical analysis* of network performance. The most classical approach is *Queuing Theory*. However, it may not be a good choice for real-time analysis for several reasons: first, it uses Normal or Poisson inputs, which are not representative of the bursty and periodic characteristics of

real-time traffic (Song et al., 2002); second, the delay calculation is not adequate for several priority levels (Jasperneite et al., 2002). *Simulation* is another classical technique, but it is only valid when it covers a representative subset of the scenarios. Some other statistical methods proposed more recently are *Stochastic Network Calculus* (Jiang & Liu, 2008) and *Histogram Calculus* (Vila & Hernández, 2008) but they require a switch model, which is not always available.

A first evaluation of Switched Ethernet using strict periodic traffic that is based on simulation was done by (Pedreiras et al., 2003). The results showed that when the switches are heavily loaded, they may behave erratically with unpredictable delays, and they may drop high priority packets due to a lack of memory capacity. One of the most comprehensive and interesting evaluations was done by (Scharbarg & Fraboul, 2007). This evaluation was done using several switches, and it showed a comparison of several analytical and evaluation methods. The most accurate results were obtained via simulation. The major deviations from the Network Calculus were obtained with one single switch (up to 70%). A previous evaluation performed by (Jasperneite et al., 2002) also confirmed that analytical methods show noticeable deviations from simulations and the difficulty of characterizing the worst-case scenario. Another evaluation was performed by (Loeser & Haertig, 2004). It makes a comparison of analytical bounds and simulation results for the approach based on traffic-shaping.

3.2 Evaluation platform

The system platform for the evaluations of Switched Ethernet consists of several embedded Pentium III 1200 MHz boards running Linux with kernel version 2.6.23hrt (high-resolution timers). These boards are later referred to as *blade1*, *blade2*, etc. Every board is connected to two Ethernet networks: one is a conventional Ethernet, which is used for the traffic generated by the operating system (NFS and some other services); the other one is an industrial switched Ethernet network that is only used for traffic generated by the experiments. Using two networks guarantees that OS-generated traffic does not interfere with real-time traffic at all.

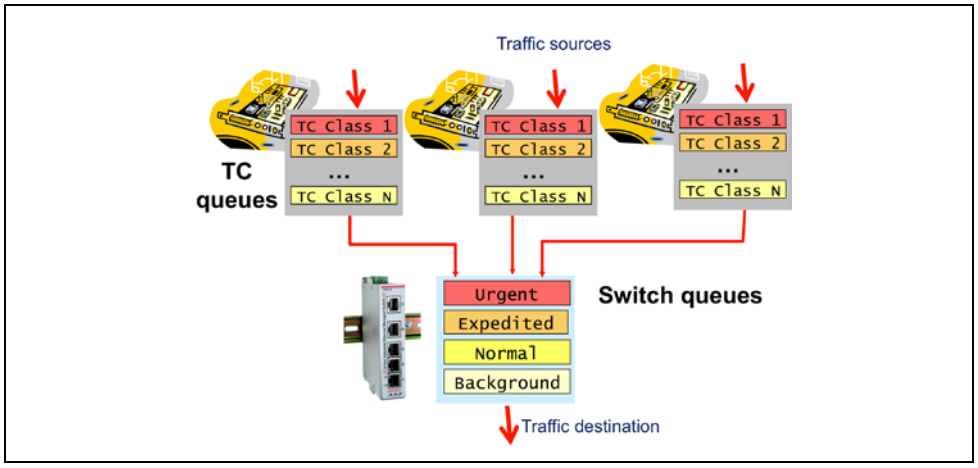


Fig. 5. System platform.

In the industrial network, traffic goes through two message queues (Fig. 5): the Linux TC queues (located at each outbound network interface) and the switch port queues.
The switch used in our benchmark is a Sixnet SL-5MS industrial 100 Mbps managed switch that provides up to four priority queues. The switch has been configured to support three CoS (Classes of Service) based on three non-preemptive priority levels: *Real-Time, Streaming,* and *Best-Effort.*
Linux TC has been configured to provide the same three CoS and, in some cases, to perform token-bucket shaping.

3.3 Metrics

The experiment metrics are the *round-trip delay* and the *delay jitter* of a real-time periodic workload. Messages are transmitted periodically from a sender process to a receiver process that, in turn, bounces the messages back to the sender.
The *round-trip delay* is defined as the time between the sending of a packet and the reception of the return message. It includes the network latency, the switch delay, the queue times, and the operating system overhead. In order to minimize this overhead, the sending and receiving tasks are scheduled at the maximum Linux user priority.
The *delay jitter* is defined as the absolute value of the instantaneous packet delay variation at the receiver process. This is the difference between successive packets. For example, consider that packets are transmitted every 1 ms. Thus, we expect to receive a packet 1 ms after the previous one. However, if the 2nd packet is received 0.8 ms after the 1st packet, the jitter is $|-0.2| = 0.2$ ms.
The evaluated metrics show the average value of the round-trip delay and the jitter and their dispersion in the form of probability distribution.

3.4 Workload definition

The workload consists of traffic flows with different classes of service, which in this evaluation were implemented using strict priorities for both the switch and Linux TC mechanisms:

- The *real-time workload* consists of UDP periodic messages. These have a fixed size of 570 bytes which is approximately the average of the Ethernet frame and a period, which is typically 0.5—1 ms.
- The *background workload* is a workload whose only goal is to consume some network bandwidth and so "disturb" the real-time workload, possibly affecting its performance metrics. Different background workloads are used. They are parameterized by two factors: *packet size* and *bandwidth utilization*.

The parameters for characterizing the background workload are those that have been reported to have a large impact on Ethernet predictability. Packet size is important because the higher it is, the higher the effect of limited pre-emption is. On the other hand, bandwidth utilization is also very important, since the main recommendation of network manufacturers for preserving the network predictability is to keep bandwidth utilization low. For the background workload, different UDP and TCP workloads were used. However, results for TCP did not differ much from UDP, so only the results for UDP are presented.

The main difference between TCP and UDP could be that in TCP, the packet size is variable and the workload is shaped to an average bandwidth utilization and a maximum burst size. However TCP breaks these packets into fixed size IP packets (fragmentation) so, in the end, the situation is not much different from UDP with fixed size packets.

3.5 Evaluation scenarios

Another purpose of the experiments is to show the effectiveness of each of the two traffic control mechanisms: the managed switch and the Linux TC mechanisms. For this reason, several scenarios have been properly configured. They are shown in Fig. 6.

To evaluate the effect of Linux TC, the real-time traffic is sent between nodes *blade2* and *blade4*, and the background workload is sent from *blade2* to *blade3*. In this scenario, the real-time messages go through the following queues[1] (see Fig. 6, Contention at TC): TC2 → SP4 → TC4 → SP2. Messages from *blade2* to *blade4* contend with the background traffic at TC2, while the messages returning from *blade4* to *blade2* meet the ACKs generated by the background traffic in SP2. According to this, only TC affects the jitter of the messages arriving at *blade4*, while both TC and the switch affect the round-trip delay. However, the effect of ACKs is negligible. Thus, this scenario mostly measures the effectiveness of Linux TC.

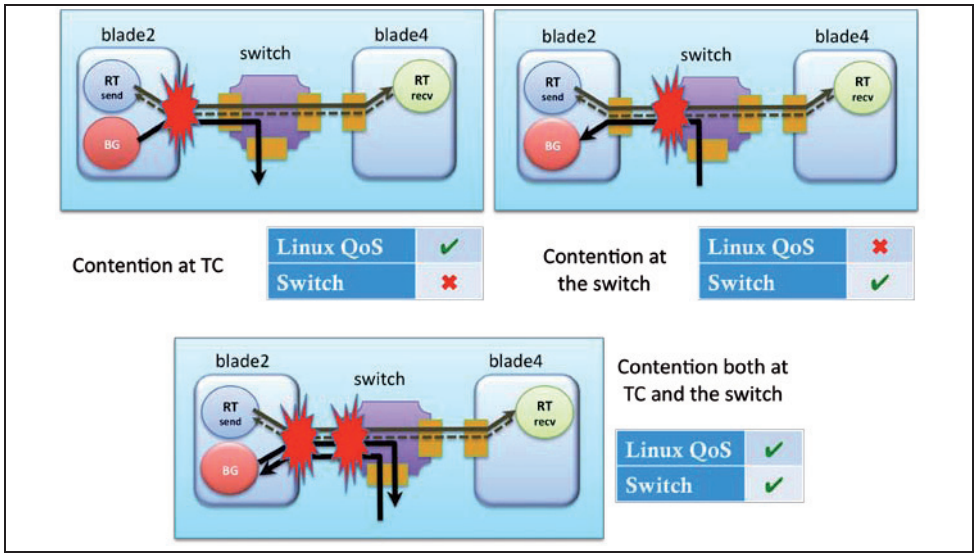


Fig. 6. Evaluation scenarios.

To evaluate the effect of the switch, the background workload is sent from *blade4* to *blade2* (see Fig. 6, Contention at the switch). Now, the real-time messages *blade2* → *blade4* contend with the background traffic in SP2, so this scenario measures the effectiveness of the switch.

[1] TCn stands for the queues of the outbound network interface of node n, while SPn stands for the queues of the switch port of node n.

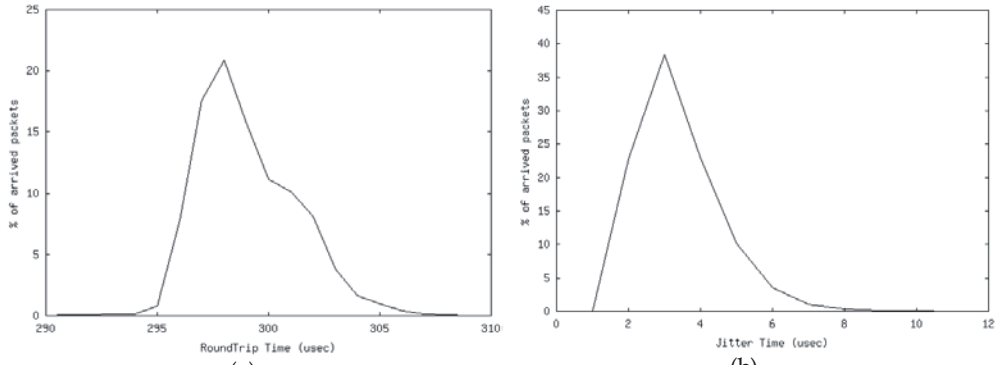
Finally, in a scenario where a background workload is sent from *blade2* to *blade3* and another background workload is sent from *blade3* to *blade2* (see Fig. 6, Contention both at TC and the switch), the real-time messages *blade2* → *blade4* contend with the background traffic in TC2 and SP2, so this scenario measures the combined effectiveness of Linux TC and the switch.

4. Evaluation results

This section presents the evaluation results of the managed Ethernet switch and Linux TC mechanisms and the influence of some parameters, like *packet size* and *bandwidth utilization* on the results. However, in order to be able to properly assess the results obtained in the evaluations, it is convenient to have some idea about the best and worst results that can be obtained. This is done through the analysis of the *idle* and *congestion* scenarios.
All the statistical results presented in this section were obtained by repeating the measures with 10^5 real-time messages.

4.1 Idle and congestion scenarios

The *idle scenario* is a scenario where the real-time workload is transmitted alone, without using any background workload. It is expected to provide the lowest achievable bounds on delay and jitter. The *congestion scenario* is a scenario where a background workload tries to saturate the network by sending packets at the maximum achievable speed. It is expected to disturb the real-time workload as much as possible and yield the worst possible results.
In the *idle scenario*, the real-time workload has a period of 1 ms and a total packet size of 570 bytes. That requires a bandwidth utilization of approx. 4.56 Mbps (4.56%). The probability distribution of the round-trip delays and the period jitter are shown in Fig. 7 (a) and (b), respectively. Round-trip times are in the range [290, 325] μs, with an average value of 300 μs. The standard deviation is very small. The average jitter value is about 3 μs and the maximum is 25 μs. The small variations in round-trips and jitter are mainly due to the unpredictability that the protocol stack (TCP, UDP, IP) and the OS drivers introduce.



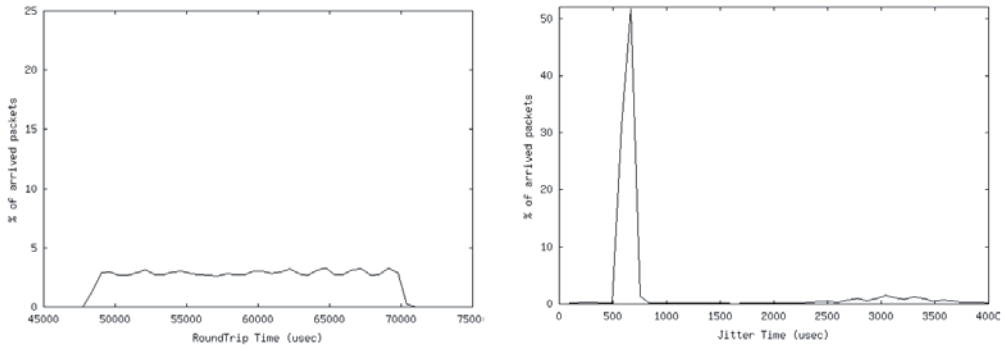
(a) (b)

Fig. 7. Idle scenario: (a) Round-trip, (b) Jitter.

The *congestion scenario* consists of the same real-time workload and a background workload generated by sending a TCP stream between the same nodes at the maximum achievable

speed. The background traffic saturated the outbound network queues of the sending node and the switch port queues of the sending and receiving stations. As a consequence, all the real-time messages suffered serious delays and even packet losses.

Fig. 8 (a) and (b) show the probability distributions of the round-trip delays and jitter, respectively. Both increase by several orders of magnitude compared to the idle scenario: the round-trip time is in the range [48000, 71000] μs, with an average of 58330 μs and a high standard deviation of 6283 μs. The jitter is in the range [0, 21429] μs with an average of 877 μs and a standard deviation of 844 μs.



(a) (b)

Fig. 8. Congestion scenario: (a) Round-trip, (b) Jitter.

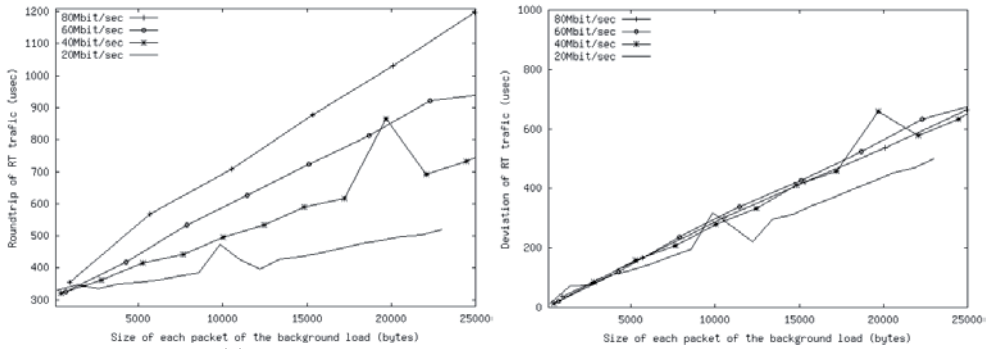
4.2 Effect of the packet size

The goal of this set of experiments is to show that the round-trip delay and jitter are affected by the packet size of the background workload. An increase in the packet size of the background workload is expected to increase the average transmission delay. This is due to the limited preemption effect: low priority Ethernet frames cannot preempt higher priority ones.

In Linux TC, this effect is much more serious since the TC scheduler treats UDP packets as indivisible; therefore, the preemption unit is the UDP packet and not the Ethernet frame. Experiments for measuring the effect of the packet size have been done using UDP because, unlike TCP, it allows setting a fixed packet-size. This is important since even though UDP packets are further fragmented into Ethernet frames, the Linux TC scheduler prioritizes UDP packets and not the Ethernet frames.

In the experiment, the real-time workload consists of UDP periodic messages of 570 bytes with a period of 1 ms. The background workload was characterized by a packet size in the range [0.1, 25] Kbytes and bandwidth utilization in the range [20, 80] Mbps (accomplished by varying the period of the background workload).

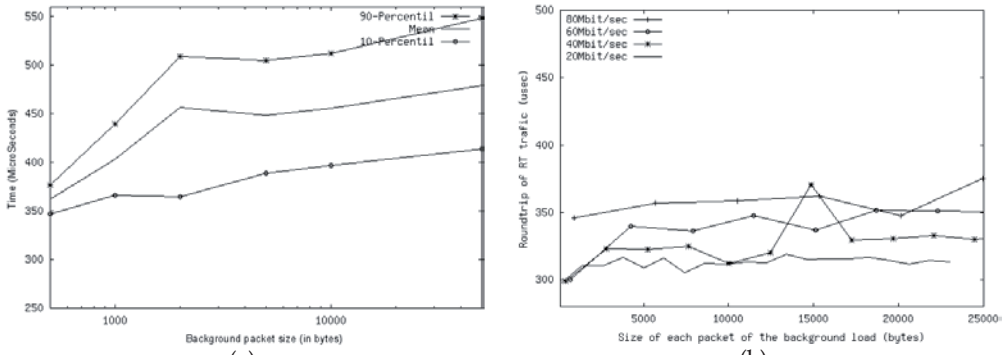
Results show that when Linux TC resolves traffic contention, round-trip delays grow linearly with the packet size, as shown in Fig. 9 (a). For example, when the bandwidth utilization increases from 1 to 80 Mb, it grows from about 300 μs to 1200 μs. The standard deviation (Fig. 9 (b)) also grows linearly up to 600 μs.



(a) (b)

Fig. 9. Effect of the packet size in Linux TC: (a) Round-trip delay, (b) Standard deviation

The switch proved to be much more effective for solving contention. This is shown through another experiment where the switch was used for prioritizing the traffic while Linux TC was used for traffic-shaping. In this experiment, we were aware that the real usefulness of Linux TC was traffic-shaping rather than traffic-prioritizing. The experiment was done using a highly bursty UDP background workload that was shaped, using a token-bucket filter at an average rate of 80 Mbps. UDP packets were fixed-size in a range of [0, 50] Kbytes. Fig. 10 (a) shows the results for the average round-trip times (on a logarithmic scale) and the percentiles 10 and 90. Delays grow linearly up to 1500 bytes (Ethernet frame size) and then remain almost constant about 460 μs, which is not much more than the idle scenario (460 μs). Fig. 10 (b) shows more clearly how the delay remains constant for larger packets and different bandwidth utilizations. The standard deviation also follows the same pattern and remains almost constant at about 45 μs for large packet sizes. Jitter shows similar results.

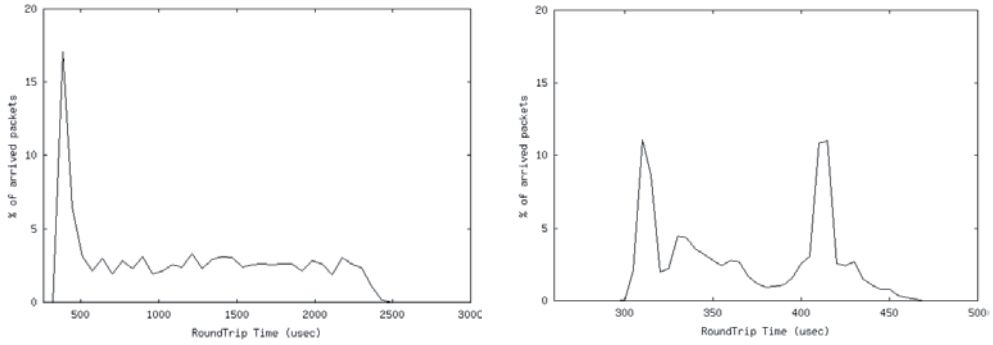


(a) (b)

Fig. 10. Effect of packet size with the managed switch: (a) Logarithmic BW=80, (b) Linear

The typical profile of the statistical distributions of the round-trip delays is shown in Fig. 11. Two cases are shown: when the contention occurs at Linux TC (Fig. 11 (a)), and when it occurs at the switch (Fig. 11 (b)). The distributions correspond to a background workload with a bandwidth utilization of 80 Mbps and an UDP packet size of 23 Kbytes, but they are

all very similar for any packet size. In general, distributions do not differ much, although, in Linux TC, the average is higher. Compared to the idle scenario, the dispersion of these distributions increases moderately.



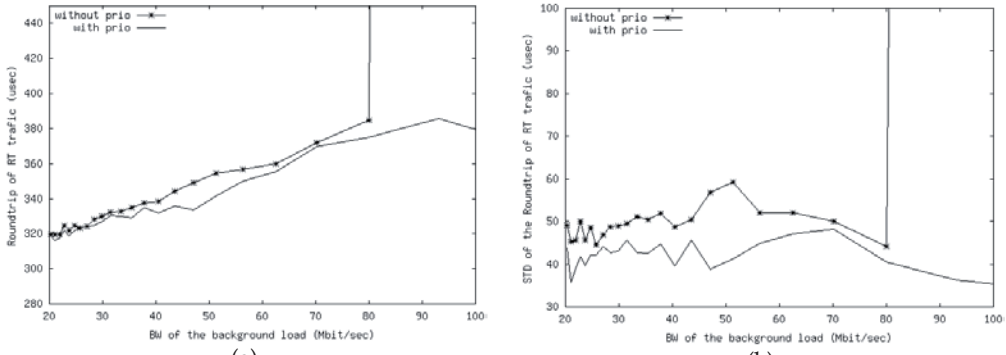
(a) (b)

Fig. 11. Round-trip distribution: (a) Contention at Linux TC, (b) Contention at the switch

4.8 Effect of the bandwidth utilization

These experiments show that increasing the bandwidth utilization of the background workload makes Switched Ethernet clearly less predictable, as reported by most manufacturers. The reason for this is that the average transmission delay increases with the bandwidth utilization because the length of packet queues grows. However, the experiment also shows that prioritizing real-time traffic suppresses this effect even for bandwidth utilizations close to saturation. This is because priorities keep high priority packets from having to wait for other queued packets.

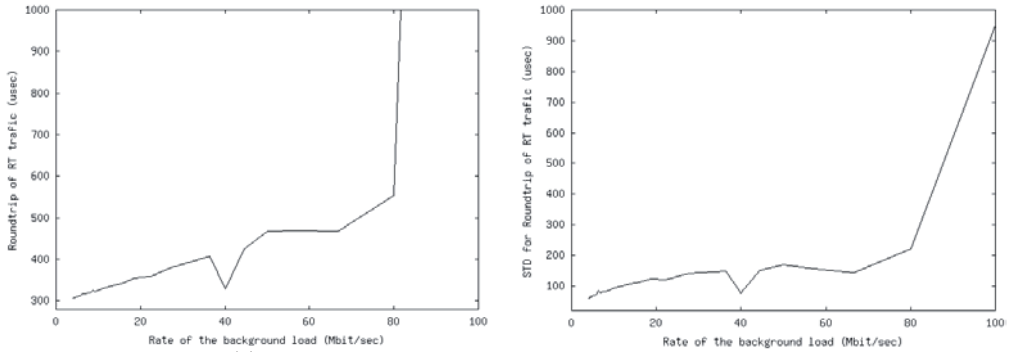
The real-time workload was the same as in the previous experiments. The background workload was a UDP flow with a packet size of 25 Kbytes and variable bandwidth utilization in the whole range (0 thru 100%).



(a) (b)

Fig. 12. Effect of bandwidth utilization with the switch: (a) Round-trip, (b) Std. deviation

The results of Fig. 12 (a) show the effect of prioritizing the real-time traffic in the switch. It shows the round-trip delays with and without prioritization. Prioritization is key for utilizations higher than 80%. Round-trip delays grow slowly from 320 μs to 380 μs for utilizations below 80%. For higher utilizations, priorities keep delays predictable, while lack of them make the system collapse. The standard deviation remains almost constant about 40 μs for the whole bandwidth range. In summary, switch prioritization is key for maintaining the performance close to an idle scenario in the whole range of bandwidth utilizations.
The effect of Linux priorities on this experiment is practically null. Fig. 13 only shows the round-trip delays with prioritization, since the results without prioritization are practically the same: round-trip delays grow from 320 μs to 550 μs when the bandwidth utilization increases from 20% to 80%. They grow dramatically for higher utilizations where results are close to the congestion scenario.



(a) (b)

Fig. 13. Effect of bandwidth utilization with Linux TC: (a) Round-trip, (b) Std. deviation

5. Conclusion

This work has presented a characterization of some approaches for making Switched Ethernet predictable. The solutions studied are based on hardware or software mechanisms that modify the MAC level and allow the use of the TCP/IP on top of it. Analytical performance bounds for these solutions have been presented. Statiscal results have also been presented for the case of an industrial managed switch using traffic prioritization features and end-systems running Linux with the TC packet scheduler.
The results show that the switch prioritization feature is key for maintaining Ethernet predictability with high bandwidth utilizations. On the other hand, the Linux TC priorities have a null effect since they schedule IP packets instead of Ethernet frames. The most useful feature of Linux TC was the traffic-shaping capability. Using traffic prioritization, average network delays and jitter have also shown to increase linearly with the packet size until reaching the Ethernet frame size, remaining constant for larger IP packet sizes. The standard deviation of the experiments also remained bounded through the whole utilization range.
In summary, Switched Ethernet has shown to be a good option for real-time transmission and maintaining a good level of predictability when using traffic-shaping and prioritization.

6. References

Black, D; Carlson, M; Davies, E; Wang, Z & Weiss W. (1998). *An Architecture for Differentiated Services,* RFC-2475. Dec. 1998.

Charara, H.; Scharbarg, J.-L.; Ermont, J. & Fraboul, C. (2006). Methods for bounding end-to-end delays on an AFDX network, *Proceeding of the 18th Euromicro Conference on Real-Time Systems*, pp. 202-212, ISBN 0-7695-2619-5, Prague Czech Republic, 30 July 5-7 2006, Published by IEEE.

Decotignie, J.D. (2005). Ethernet-based real-time and industrial communications. *Proceedings of the IEEE,* Vol. 93, No. 6, (June 2005) page numbers (1118-1128).

Felser, M. (2005). Ethernet-based real-time and industrial communications. *Proceedings of the IEEE,* Vol. 93, No. 6, (June 2005) page numbers (1102-1117).

Kurose, J.; Schwartz, M.; & Yemini, Y. (1984). Multiple access protocols and time-constrained communication. *ACM Computing Surveys*, Vol. 16, No. 1, (Mar. 1984) page numbers (43–70).

IEC International Electrotechnical Commission (2004) *Proposal for a Publicly Available Specification for Real-Time Ethernet,* documents IEC 65C/356a/NP, IEC 65C/341/NP, IEC 65C/360/NP, IEC 65C/359/NP.

Jasperneite, J; Neuman, P; Theis, M. & Watson K. (2002). Deterministic Real-Time Communication with Switched Ethernet, *Proceeding of the 4th IEEE International Workshop on Factory Communication Systems (WFCS02)*, pp. 11-18, ISBN 0-7803-7586-6, Vasteras Sweden, Published by Springer-Verlag. 2002.

Jiang, Y. & Liu, Y. (2008). *Stochastic Network Calculus.* ISBN 978-1-84800-126-8, Ed. Springer-Verlag. Berlin. 2008.

LeBoudec, J.-Y. & Thiran, P (2002).*Network Calculus: A Theory of Deterministic Queuing Systems for the Internet,* Lecture Notes in Computer Science, Vol. 2050. ISBN 3-540-42184-X, Ed. Springer-Verlag. Berlin.

Le Lann, G. (2004). *A deterministic multiple CSMA-CD protocol,* INRIA-Project Score, Internal Rep. PRO-1-002, Jan. 1983.

Loeser, J. & Haertig, H. (2004). Low latency hard real-time communication over switched Ethernet, *Proceeding of the 16th Euromicro Conference on Real-Time Systems (ECRTS04)*, pp. 13-22, ISBN 0-7695-2176-2, Catania Italy, 30 June-2 July 2004, Published by IEEE.

Pedreiras, P; Leite, R & Almeida L. (2003). Characterizing the real-time behavior of prioritized switched-Ethernet, *Proceeding of Proceedings 2nd International Workshop on Real-Time LANs in the Internet (RTLIA 2003)*, pp. 59-62, Porto, Portugal, July 2003.

Pedreiras, P.; Gai, P.; Almeida L. & Butazzo G.C.(2005). FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems. *IEEE Transactions on Industrial Informatics*, Vol. 1, No. 3, (Aug. 2005) page numbers (162–172).

Scharbarg, J-L. & Fraboul, C. (2007). Simulation for end-to-end delays distribution on a switched Ethernet, *Proceeding of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007)*, pp. 1092-1099, ISBN 78-1-4244-0825-2, Published by IEEE. 2007.

Schemm, E. (2004). SERCOS to link with Ethernet for its third generation, Computing & *Control Engineering Journal*, Vol. 15, No. 2, (April- May 2004), page numbers (30-33)

Song, Y. ; Koubaa, A. & Lorraine L. I. (2002). Switched Ethernet for real-time industrial communication: modelling and message buffering delay evaluation, *Proceeding of the 4th IEEE International Workshop on Factory Communication Systems (WFCS02)*, pp. 27-35, ISBN 0-7803-7586-6, Vasteras Sweden, Published by Springer-Verlag. 2002.

Varadarajan, S. (2001). Experiences with EtheReal: A fault-tolerant real- time Ethernet switch, *Proceedings 8th IEEE Int. Conf. Emerging Technologies and Factory Automation*, pp. 183–194, Antibes-Juan les Pins, France. IEEE Computer, Society Press, 2001.

Venkatramani, C. & Chiueh T. (1994). Supporting real-time traffic on Ethernet, *Proceedings 15 th IEEE Real-Time Systems Symposium*, pp. 282–286, San Juan, Puerto Rico. IEEE Computer, Society Press, 1994.

Vila-Carbó, J. & Hernández-Orallo, E (2008). An analysis method for variable execution time tasks based on histograms. *Real-Time Systems Journal,* Vol. 38, No. 1, (Jan. 2008) page numbers (1-37), ISSN 0922-6443.

Yiming, A. & Eisaka, T. (2002). Support industrial hard real-time traffic with switched ethernet, *Embedded Software and Systems (ICESS05).* Lecture Notes in Computer Science Vol. 3820. pp. 671–682. ISBN 978-3-540-30881-2. Springer-Verlag. 2002.

Zhang, Q & Zhang, W (2005). Priority Scheduling in Switched Industrial Ethernet, *Proceedings of American Control Conference*, Portland, OR, USA. Jun 2005.

12

Token Passing Techniques for Hard Real-Time Communication

Gianluca Franchino*, Giorgio C. Buttazzo* and Tullio Facchinetti**.
*Scuola Superiore Sant'Anna, Italy
**University of Pavia, Italy

1. Introduction

Distributed computing platforms are increasingly used to develop critical embedded systems, like control applications, sensors networks, telecommunication, and robotics systems. In such distributed applications, the correct behaviour depends on the timely execution of the tasks running on different nodes, which may frequently exchange shared data. In particular, since the nodes are typically connected through a common channel without the need of multi-hop communication, it turns out that a timely communication mainly depends on how the nodes access the channel. Even when a multi-hop communication is needed, a timely message delivery is not feasible without the support of a predictable channel access mechanism, which is implemented in the Medium Access Control (MAC) sub-layer of the communication stack.
There exist several MAC protocols designed for providing a timely communication among distributed nodes, mainly in the factory communication domain. One of the most effective solutions is given by token passing protocols, which have some nice characteristics that make them suitable for real applications. For instance, because of the token passing mechanism, the nodes do not need to be synchronized and they have an implicit bandwidth reclaiming mechanism that allows other nodes to exploit the unused bandwidth. Moreover, such protocols can serve both real-time and best-effort (non real-time) traffic.
Among token passing protocols, the timed token policy is a channel scheduling approach first proposed by Grow in (Grow, 1982). Since then, it has received a substantial attention and several relevant results have been derived (Zhang et al., 2004), which make timed token protocols suitable for the real-time communication in industrial applications. Timed token policies are used as channel access mechanism in several standard protocols such as, for instance, PROFIBUS (Profibus, 1996) and FDDI (FDDI, 1987). However, the application domain of the timed token policy is not restricted to the cited communication standards, and some examples on their use can be found in (Lenzini et al., 2004) and (Cicconelli et al., 2007).
To improve the ability of timed token protocols of managing real-time traffic, Shin and Zheng (Shin & Zheng, 1995) proposed a modification of the timed token protocol, which can guarantee a greater bandwidth for the real-time traffic with respect to the classic timed token protocol; however, under certain conditions, it cannot manage the best-effort traffic (Franchino et al., 2008). The Budget Sharing Token (*BuST*) protocol has been proposed as an improvement with respect to existing timed token protocols (Franchino et al., 2008). In

particular, *BuST* improves the bandwidth guaranteed for the real-time traffic with respect to the timed token protocol, and it can manage best-effort traffic in those situations where the modified timed token protocol fails on serving this kind of traffic.
This chapter introduces a few token passing protocols, presenting their characteristics and illustrating the main results available in the literature. The *BuST* protocol is discussed in detail, illustrating its main advantages by means of analytic and simulation comparisons. New and well known properties of the protocols for managing both hard real-time and best-effort traffic will be analyzed, describing drawbacks and strengths of each protocol. The analysis is carried out considering the main budget allocation schemes available in the literature, thus contributing to the comparative characterization of the several schemes nowadays available.

2. Network Model

The communication network is composed by n nodes sharing a common medium, e.g. a bus, where each node can transmit both real-time and best-effort traffic. The former kind of traffic is modelled by assigning each node i a synchronous message stream S_i, which is described by three parameters (C_i, D_i, T_i), where:

- C_i is maximum amount of time necessary to transmit a message generated in the stream S_i. This includes the time to transmit both the message payload and the message headers/footers;
- D_i is the relative deadline of a message generated by stream S_i, that is, the maximum amount of time that a message can wait in transmission queue before its transmission is completed. Hence, the transmission of the *j-th* message, which is queued at time $t_{i,j}$, must be completed no later than its absolute deadline $d_i = t_{i,j} + D_i$.
- T_i is the generation period of messages in stream S_i. If the *j-th* message in the stream S_i is put in the transmission queue at time $t_{i,j}$, then the *(j+1)-th* will arrive in the transmission queue at time $t_{i,j+1} = t_{i,j} + T_i$.

Without loss of generality, only a stream per node it is considered, since the case of a network with more streams per node can be represented with a logical equivalent one with a stream per logical node, as showed in (Agrawal et al., 1994). We consider that each node i is assigned a bandwidth H_i, also called time budget, used to bound the transmission of the node.
The channel utilization of each message in stream S_i is:

$$U_i^S = \frac{C_i}{\min(T_i, D_i)} \tag{1}$$

The total channel utilization of a periodic stream set is then:

$$U^S = \sum_{i=1}^{n} U_i^S \,. \tag{2}$$

which measures the channel bandwidth utilized by the real-time (periodic) traffic.
Before describing the protocols, the following definitions are needed:

Definition 1. τ *is the time needed to transmit the token between nodes, including the overhead introduced by the protocol.*

Definition 2. The *Target Token Rotation Time (TTRT) represents the expected time needed by the token to complete an entire round-trip of the network.*

Any assignment of the time budgets H_i must satisfy the following two constraints:

Definition 3. (*Protocol Constraint*) *The total bandwidth allocated to the nodes, during one complete token rotation, must be less than the available network bandwidth, that is,*

$$\frac{\sum_{i=1}^{n} H_i}{TTRT} \leq 1 - \frac{\tau}{TTRT} \tag{3}$$

The Protocol Constraint is necessary to ensure a stable operation of the protocols.
Definition 4. (*Deadline Constraint*) *If $s_{i,j}$ denotes the time at which the transmission of the j-th message in stream S_i is completed, the deadline constraint requires that for i=1, …, n, and j=1, 2, …,*

$$s_{i,j} \leq t_{i,j} + D_i \tag{4}$$

where $t_{i,j}$ is the message arrival time and D_i is its relative deadline.
Meeting the Deadline Constraint ensures that every periodic message is transmitted before its absolute deadline. Note that in Inequality (4), while $t_{i,j}$ and D_i are defined by the application, $s_{i,j}$ depends on the bandwidth (budget) allocation and on the *TTRT* value.
Definition 5. *A stream set $\Gamma=\{S_1, S_2, \ldots, S_n\}$ is said to be feasible or schedulable when both the Deadline Constraint and the Protocol Constraint are met.*
To test the Protocol Constraint it is sufficient to check whether the sum of the budgets are not greater than the *TTRT* minus the overhead τ. However, testing the Deadline Constraint may be much more complicated. The following method is often used for testing whether the Deadline Constraint is met:
Let X_i be the minimum amount of time available for node *i* to transmit its *j-th* message during the time interval $(t_{i,j}, t_{i,j} + D_i]$, then for a message stream set with message deadlines not greater than periods, the Deadline Constraint can be satisfied if and only if for all *i*, $i=1,2,\ldots,n$, $X_i \geq C_i$.

3. Timed Token Protocols

The timed token protocol (*TTP*) is a basic channel access technique, namely a MAC protocol, which can be used to manage real-time traffic while guaranteeing a fair sharing of the unused bandwidth to the best-effort traffic. In timed token approaches, a token travels between nodes in a circular fashion and each node can transmit only when it possesses the token. An important parameter is the Target Token Rotation Time (*TTRT*), which represents the expected time needed by the token to complete an entire round-trip of the network. Each node *i* has an associated time budget H_i; whenever a node receives the token, it can transmit

its real-time messages for a time no greater than H_i. It can then transmit its best-effort messages if the time elapsed since the previous token arrival to the same node is less than the value of *TTRT*, that is, only if the token arrives earlier than expected. Figure 1, shows a typical timed token based network where 4 nodes sharing a common channel are arranged in a logical ring, as far as the token passing mechanism is concerned.

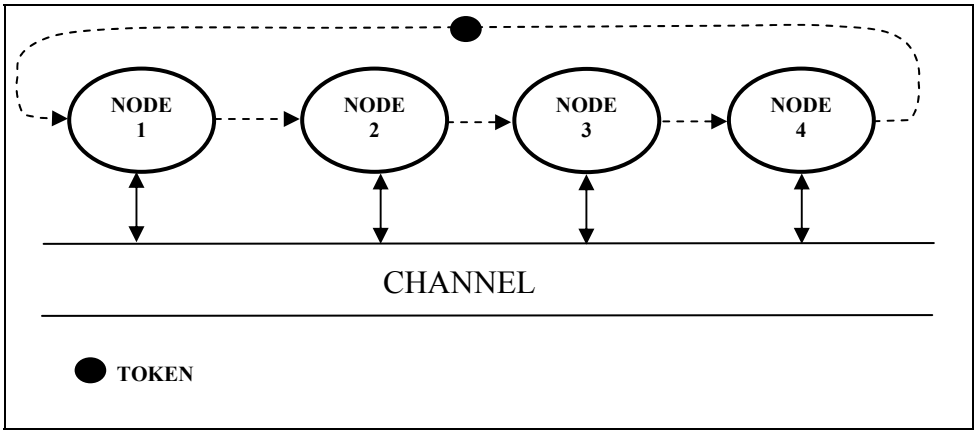


Fig. 1. Timed token network

3.1 *TTP* operation rules

To better understand the *TTP* operation, the protocol channel access rules are detailed below:

- During the network start up, each node *i* declares a *TTRT* value equal to one half of the deadline D_i related to its message stream S_i. The minimum declared value is chosen as *TTRT*, and each node *i* is then assigned a time budget H_i that depends on *TTRT*.
- Each node uses two timers, the token holding timer (*THT*) and the token-rotation-timer (*TRT*). The *TRT* counter always increases, whereas the *THT* only increases when the node is delivering best-effort traffic. When *TRT* reaches *TTRT*, it is reset to 0 and the token is signed as "late" by incrementing the node's late counter L_c by one. To initialize the timers and L_c, no messages are sent during the first token rotation after the ring initialization.
- Only the node holding the token can transmit messages. Transmission is controlled by the timers, but an in-progress transmission of a single packet is not interrupted until its completion. When node *i* gets the token, it performs the following operations:
  - If $L_c > 0$, it sets $L_c = L_c - 1$ and *THT* = *TTRT*. Otherwise, *THT* = *TRT* and *TRT* = 0.
  - If node *i* has synchronous packets, it transmits them for a time no greater than H_i.
- If node *i* has best-effort packets, it transmits them until *THT* counts up to *TTRT*, or until all the asynchronous traffic is sent, which ever comes first.

- Node i passes the token to station (i +1) mod (n).

The main drawback of *TTP* is the inability of guaranteeing the total available bandwidth for the real-time traffic. As Johnson and Sevciks (Johnson & Sevciks, 1987) showed, the average interval between two consecutive visits of the token at the same node, namely the average token rotation time, does not exceed *TTRT* and the maximum rotation time does not exceed 2*TTRT*. Thus, if D is the minimum deadline among stream deadlines, i.e. $D = min(D_i)$, it turns out that $TTRT = D/2$. From the Protocol Constraint it follows that:

$$\sum_{i=1}^{n} H_i \leq TTRT - \tau = \frac{D}{2} - \tau \tag{5}$$

In the time interval defined by D, the bandwidth available for the real-time traffic can be obtained dividing the above equation by D:

$$\frac{\sum_{i=1}^{n} H_i}{D} \leq \frac{1}{2} - \frac{\tau}{D} \tag{6}$$

Since the total available bandwidth is 1 – τ/TTRT, the *TTP* can guarantee at most half of the total available bandwidth for the real-time traffic.

3.2 Modified *TTP* operation rules

To improve the bandwidth guaranteed for the real-time traffic, Shin (Shin & Zheng, 1995) proposed to modify the timed token rules, limiting the maximum time available for best-effort traffic to T_B^{MAX}, where:

$$T_B^{MAX} = TTRT - \sum_{i=1}^{n} H_i - \tau \tag{7}$$

Notice that, under *TTP*, the maximum bandwidth allocated for best-effort messages is $T_B^{MAX} = TTRT - \tau$.

In *(Chan et al., 1997)*, the authors showed that the maximum token rotation time for the Modified Timed Token Protocol (*MTTP*) is bounded by *TTRT*. This means that, under MTTP it is possible to select a value for *TTRT* no greater than the minimum deadline D. Hence, being $TTRT \leq D$, from the Protocol Constraint it follows that:

$$\sum_{i=1}^{n} H_i \leq D - \tau \tag{8}$$

and dividing by D, it gives:

$$\frac{\sum_{i=1}^{n} H_i}{D} \leq 1 - \frac{\tau}{D} \tag{9}$$

That is, *MTTP* can guarantee all the available bandwidth for the real-time traffic.
By defining $T_S = \sum H_i$, to guarantee that $T_B^{MAX} = TTRT - T_S - \tau$, *MTTP* can be defined as follows:

1. A new token rotation time is defined as $TTRT_n = TTRT - T_S$, and used instead of *TTRT*;
2. The counting of a node's token rotation timer (*TRT*) is stopped when a real-time message is being delivered by the node.

The full details of the protocol rules are given below:

- During the network start up, each node i declares a *TTRT* value equal to the deadline D_i related to its message stream S_i. The minimum declared value is chosen as *TTRT*. Each node i is assigned a time budget H_i that depends on *TTRT*, and it sets $TTRT_n = TTRT - T_S$.
- Each node uses two timers, the token holding timer (*THT*) and the token-rotation-timer (*TRT*). The *TRT* counter increases only when the node is not transmitting real-time traffic, whereas the *THT* only increases when the node is delivering best-effort traffic.
- Only the node holding the token can transmit messages. Transmission is controlled by the timers, but an in-progress transmission of a single packet is not interrupted until its completion. When node i gets the token, it performs the following operations:
  - If $L_c > 0$, it sets $L_c = L_c - 1$ and $THT = TTRT$. Otherwise, $THT = TRT$ and $TRT = 0$.
  - If node i has synchronous packets, it transmits them for a time no greater than H_i.
- If node i has best-effort packets, it transmits them until *THT* counts up to *TTRT*, or until all the asynchronous traffic is sent, which ever comes first.
- Node i passes the token to station $(i+1) \bmod (n)$.

4. The *BuST* protocol

The *BuST* protocol has been devised to improve the ability of *TTP* in managing real-time traffic, and to overcome the problems *MTTP* can have in managing best-effort traffic (see Section 7).
Like timed token protocols, the *BuST* protocol assigns each node a time budget H_i for transmitting its real-time traffic. When a node receives the token, it can transmit the associated real-time traffic for a time no greater than the corresponding budget. The main difference with respect to *TTP* and *MTTP* concerns the best-effort traffic service. Under *TTP*, when the token arrives early, the node can transmit best-effort traffic for a time no greater than $T_A = TTRT - \tau - T_{LRT}$, where T_{LRT} is the time spent in the last round-trip of the token. Using *MTTP*, a node does the same, but with $T_A = TTRT - \tau - \sum H_i$. With *BuST*, a node can deliver non real-time traffic each time it gets the token, early or not, using the spare budget left by real-time messages. If H_i^{cons} is the budget consumed by node i to deliver periodic traffic, then it can send best-effort traffic for a time no greater than $T_{A_i} = H_i - H_i^{cons}$, even if

the token is not early. Observe that, *TTP* and *MTTP* can deliver best-effort traffic only when the token is early, that is, when $T_{LRT} < TTRT - \tau$.
The following rules specify the the *BuST* protocol in detail:

- During the network initialization phase, each node *i* declares a *TTRT* value equal to the deadline D_i of its periodic message stream. The minimum declared value is selected as *TTRT*. Each node *i* is assigned a time budget H_i that depends on *TTRT*.
- Each node has one timer, the token holding-rotation timer *THRT*. The *THRT* counter always increases. To initialize the timers, no messages are sent during the first token rotation.
- Only the node having the token can transmit messages. The transmission is controlled by *THRT*, but an in-progress transmission of a single packet is not interrupted until its completion. When node *i* gets the token, it performs the following operations:
    - It sets *THRT* = 0.
    - If node *i* has real-time packets, it transmits them until *THRT* counts up to H_i, or until all the real-time traffic is sent, whichever comes first.
    - If node *i* has best-effort packets, it transmits them until *THRT* counts up to H_i, or until all the best-effort traffic is sent, which ever comes first.
    - If a real-time message becomes ready during the transmission of best-effort packets, and $THRT < H_i$, the transmission is stopped and the node starts delivering the real-time traffic until *THRT* counts up to H_i, or until all the periodic traffic is sent, whichever comes first.
    - If node *i* completes the transmission of the periodic traffic without entirely consuming its budget, i.e. $THRT < H_i$, it starts transmitting its non real-time traffic, if any, until *THRT* counts up to H_i, or until all the best-effort traffic is sent, which ever comes first. Note that, in this case, the transmission is not stopped even if a real-time message becomes ready.
    - Node *i* passes the token to station $(i+1) \bmod (n)$.

The overhead generated when a best-effort transmission is interrupted can be easily accounted in τ. The same observation can be done for the overhead generated when an in-progress packet transmission is not interrupted until its completion. This also holds for *TTP* and *MTTP*.
As we can note from the protocol rules, under *BuST*, any node *i* exploits its time budget H_i to deliver both real-time and non real-time messages. When compared to *TTP*, *BuST* improves (as *MTTP*) the bandwidth available for real-time messages and halves the bandwidth lost due to the protocol overhead. In addition, *BuST* is able to deliver best-effort traffic also in those cases in which *MTTP* fails, as it will be shown in Section 7.
As a final remark, the implementation of *BuST* only requires one timer, instead of the two timers needed by *TTP* and *MTTP*. This can be useful when *BuST* is adopted in small embedded systems, where resources (e.g., hardware timers) are scarce.

5. Time properties

In this section, we introduce the main time properties of the protocols under analysis. These properties are the basis to understand the protocols timing behaviour, to verify if a given periodic stream set $\boldsymbol{M}=\{S_1, S_2, \ldots, S_n\}$ is feasible, i.e. both the Protocol and the Deadline

Constraints are met. Moreover, the results showed in the following can be used to analyse the protocols performance under the budget allocation schemes introduced in the next sections.
Lemmas 5.1, 5.2 and 5.3 provide an upper bound on the maximum transmission time for a real-time message under the *BuST*, *MTTP* and *MTTP* protocols.

Lemma 5.1 *Under the BuST protocol, for all budget allocation schemes, if* $T_i \geq TTRT$, $i=1,\ldots,n$, *it holds:*

$$\forall i,j : s_{i,j} \leq t_{i,j} + \left\lceil \frac{C_i}{H_i} \right\rceil \left(\sum_{k=1}^{n} H_k + \tau \right) \tag{10}$$

Proof. See (Franchino et al., 2007). □

Lemma 5.2 *Under the TTP protocol, for all budget allocation schemes, if* $T_i \geq 2TTRT$, $i=1,\ldots,n$, *it holds:*

$$\forall i,j : s_{i,j} \leq t_{i,j} + \left(\left\lceil \frac{C_i}{H_i} \right\rceil + 1 \right) TTRT + C_i - \left\lceil \frac{C_i}{H_i} \right\rceil H_i \tag{11}$$

Proof. See (Franchino et al., 2007).

□

Lemma 5.3 *Under the MTTP protocol, for all budget allocation schemes, if* $T_i \geq TTRT$, $i=1,\ldots,n$, *it holds:*

$$\forall i,j : s_{i,j} \leq t_{i,j} + \left\lceil \frac{C_i}{H_i} \right\rceil TTRT + C_i - \left\lceil \frac{C_i}{H_i} \right\rceil H_i \tag{12}$$

Proof. See (Franchino et al., 2007).

□

The results and proofs in (Chen & Zhao, 1992) and (Chan et al., 1997) have been used as basis to derive the properties of *BuST* shown in the following. For comparison, the same type of results for *TPP* and *MTTP*, available in literature, are reported as well.
Theorem 5.1 gives the upper bound between any two consecutive visits of the token at the same node.

Theorem 5.1 *Under the BuST protocol, for any* $i=1,\ldots,n$, *and for any integer* $l > 0$ *it turns out that:*

$$t_i(l+1) - t_i(l) \leq \sum_{j=1}^{n} H_j + \tau \tag{13}$$

where $t_i(l)$ *is the time the token makes the* l^{-th} *visit at node i.*

Proof. Since a node *i* can use only its time budget H_i to transmit both real-time and best-effort traffic, the length of interval $[t_i(l), t_i(l+1)]$ is equal to the sum of:

1. time for real-time traffic transmission: $t_{rt} \le \sum_{j=1}^{n} H_j$;
2. time for non real-time traffic transmission: $t_{nrt} \le \sum_{j=1}^{n} H_j - t_{rt}$;
3. time due to protocol overhead and token passing: τ.

Thus,

$$t_i(l+1) - t_i(l) = t_{rt} + t_{nrt} + \tau \le t_{rt} + \sum_{j=1}^{n} H_j - t_{rt} + \tau = \sum_{j=1}^{n} H_j + \tau \cdot$$

□

The following corollary, generalizes the last theorem providing the upper bound on the time elapsed between *v* consecutive token arrivals at the same node.

Corollary 5.1 *Under the BuST protocol, for any, i=1 ,…,n, and for any integer l > 0 and v > 0 it turns out that:*

$$t_i(l+v) - t_i(l) \le v \cdot \left(\sum_{j=1}^{n} H_j + \tau \right) \tag{14}$$

where $t_i(l)$ is the time the token makes the l-th visit at node i.

Proof. By Theorem 5.1, it follows that:

$$\begin{aligned} t_i(l+v) - t_i(l) &= [t_i(l+1) - t_i(l)] + [t_i(l+2) - t_i(l+1)] + \ldots + [t_i(l+v) - t_i(l+v-1)] \\ &\le \sum_{j=1}^{n} H_j + \tau + \sum_{j=1}^{n} H_j + \tau + \ldots + \sum_{j=1}^{n} H_j + \tau = v \cdot \left(\sum_{j=1}^{n} H_j + \tau \right) \cdot \end{aligned}$$

□

The following result, gives the upper bound between two consecutive token visits at the same node with the timed token protocol.

Theorem 5.2 *Under the TTP protocol, for any, i=1 ,…,n, and for any integer l > 0 it turns out that:*

$$t_i(l+1) - t_i(l) \le TTRT + \sum_{j=1}^{n} H_j + \tau \le 2 \cdot TTRT \tag{15}$$

Proof. See (Sevcik & Johnson, 1987). □

The next corollary, provides the upper bound on the time elapsed between v consecutive token visits at same node.

Corollary 5.2 *Under the TTP protocol, for any i=1 ,…,n, and for any integer l > 0 and v > 0 it turns out that:*

$$t_i(l+v) - t_i(l) \le v \cdot TTRT + \sum_{j=1, j\neq i}^{n} H_j + \tau \tag{16}$$

where $t_i(l)$ is the time the token makes the l-th visit at node i.

Proof. See (Chen & Zhao, 1992). □

The bound between two consecutive token arrivals at the same node, under the modified timed token protocol, is provided by the following theorem.

Theorem 5.3 *Under the MTTP protocol, for any i=1 ,…,n, and for any integer l > 0 it turns out that:*

$$t_i(l+1) - t_i(l) \le TTRT \tag{17}$$

Proof. See (Chan et al., 1997). □

The corollary below, generalizes the last theorem giving the upper bound between v consecutive token vistis at the same node.

Corollary 5.3 *Under the MTTP protocol, for any i=1 ,…,n, and for any integer l > 0 and v > 0 it turns out that:*

$$t_i(l+v) - t_i(l) \le v \cdot TTRT \tag{18}$$

where $t_i(l)$ is the time the token makes the l^{-th} visit at node i.

Proof. See (Chan et al., 1997). □

The following lemmas, give the minimum amount of time available for node i to transmit its real-time traffic before the deadline of each message. Notice that, for simplicity's sake we consider a deadline $D_i = T_i$. When the deadline D_i is less than the period T_i, it is sufficient to replace T_i with D_i in the following results.

Lemma 5.3 *Under the BuST protocol, for any i=1,...,n, if at time t a periodic message with period T_i arrives at node i, then in time interval $(t,t+T_i]$ the minimum amount of time X_i available for node i to transmit the message is:*

$$X_i \le \left\lfloor \frac{T_i}{\sum_{j=1}^{n} H_j + \tau} \right\rfloor H_i + \max\left(0, H_i - \delta_i\right) \tag{19}$$

$$where\ \delta_i = \left\lfloor \frac{T_i}{\sum_{j=1}^{n} H_j + \tau} \right\rfloor \left(\sum_{j=1}^{n} H_j + \tau \right) - T_i .$$

Proof. Suppose t the time at which a new message is ready at node i. In the worst case, the token has just left node i when the new message is ready. As a consequence of Theorem 5.1, in time interval $I_i = [t,\ t + \sum_{j=1}^{n} H_j + \tau]$ node i receives the token at least once. Let $t + t_1$ be the time at which node i receives the token since t, it turns out that $0 < t_1 \le \sum_{j=1}^{n} H_j - H_i + \tau$. Hence, $H_i \le \sum_{j=1}^{n} H_j + \tau - t_1$, that is, the time available for node i to transmit its real-time traffic in I_i is H_i. Thus, by Corollary 5.1, it turns out that in a time interval $I_i = [t,\ m \cdot (t + \sum_{j=1}^{n} H_j + \tau)]$ node i has a minimum amount of time, to deliver its real-time traffic, equal to $m \cdot H_i$.
If $\Delta_i = T_i / (\sum_{j=1}^{n} H_j + \tau)$ and $m = \lfloor \Delta_i \rfloor$, two cases are possible:

1. Δ_i is an integer, such that $\delta_i = 0$ and $I_m = \left[t, t + m \cdot \left(\sum_{j=1}^{n} H_j + \tau \right) \right] = [t, t + T_i]$, then it follows that $X_i = \left\lfloor \frac{T_i}{\sum_{j=1}^{n} H_j + \tau} \right\rfloor H_i$.

2. Δ_i is not an integer, such that $\delta_i > 0$. In the worst case, the (m+1)-th token's arrival at node i may be as late as

$$t + m \cdot \left(\sum_{j=1}^{n} H_j + \tau \right) + \sum_{j=1}^{n} H_j + \tau + H_i ,$$

then, this token arrival will be not more than $t + T_i$ if:

$$\sum_{j=1}^{n} H_j + \tau - H_i \le T_i + m \cdot \left(\sum_{j=1}^{n} H_j + \tau \right) = \sum_{j=1}^{n} H_j + \tau - \delta_i .$$

In this case, the residual transmission time available for the real-time traffic is the left time until $t+T_i$, which is equal to $H_i - \delta_i$, that is:

$$X_i \le \left\lfloor \frac{T_i}{\sum_{j=1}^{n} H_j + \tau} \right\rfloor H_i + \max\left(0, H_i - \delta_i\right)$$

□

Lemma 5.4 *Under the TTP protocol, for any i=1,...,n, if at time t a periodic message with period T_i arrives at node i, then in time interval $(t,t+T_i]$ the minimum amount of time X_i available for node i to transmit the message is:*

$$X_i \le \left\lfloor \frac{T_i}{TTRT} - 1 \right\rfloor H_i + \max\left(0, \min\left(\delta_i - \tau - \sum_{j=1, j\neq i} H_j, H_i\right)\right) \tag{20}$$

where $\delta_i = T_i - \left\lfloor \frac{T_i}{TTRT} \right\rfloor TTRT$.

Proof. See (Chen & Zhao, 1992). □

Lemma 5.5 *Under the MTTP protocol, for any i=1,...,n, if at time t a periodic message with period T_i arrives at node i, then in time interval $(t,t+T_i]$ the minimum amount of time X_i available for node i to transmit the message is:*

$$X_i \le \left\lfloor \frac{T_i}{TTRT} \right\rfloor H_i + \max\left(0, H_i - \delta_i\right) \tag{21}$$

where $\delta_i = \left\lfloor \frac{T_i}{TTRT} \right\rfloor TTRT - T_i$.

Proof. See (Chan et al., 1997). □

6. Performance analysis

The real-time guarantee of a stream set highly depends on the budget allocation scheme (*BAS*) adopted for budgets assignment given the stream set parameters.

In literature exist several budget allocation schemes provided for timed-token protocols. They are traditionally classified into global or local schemes, depending on whether they need global or local information to assign the budgets. Local information is, for instance, the node stream parameters. Global information is, for instance, the number of nodes and the total channel utilization U^S required by the streams.

In this work, the budget allocation schemes are classified as proposed in (Daoxu et al. 1998). They can be divided into two categories, depending on the way they assign the budgets. The first category is the set of the *TTRT-partitioning* schemes, where a scheme belonging to this

set assigns node budgets partitioning the time expected for the token to make a rotation of the network, i.e. $TTRT - \tau$. The *TTRT-partitioning* schemes analyzed in the following are shown in Table 1.
The second category of budget allocation schemes, is the set of C_i*-partitioning* schemes, where a scheme belonging to this class assigns each budget H_i partitioning the maximum time length, C_i , to send a message from the stream S_i among a certain number of token rotation cycles. The C_i*-partitioning* schemes analyzed in the following are shown in Table 2.

TTRT-partitioning schemes	Assignment rule
Proportional Allocation (*PA*)	$H_i = U_i(TTRT - \tau)$
Normalized Proportional Allocation (*NPA*)	$H_i = \frac{U_i}{U}(TTRT - \tau)$
Equal Partition Allocation (*EPA*)	$H_i = \frac{TTRT - \tau}{n}$

Table 1. *TTRT-partitioning* budget allocation schemes.

C_i*-partitioning* schemes	Assignment rule
Local Allocation (*LA*)	$H_i = \frac{C_i}{\left\lfloor \frac{T_i}{TTRT} - 1 \right\rfloor}$
Modified Local Allocation (*MLA*)	$H_i = \frac{C_i}{\left\lfloor \frac{T_i}{TTRT} \right\rfloor}$

Table 2. C_i*-partitioning* budget allocation schemes.

The budget allocation schemes showed in the tables above have been extensively analyzed in literature. For a complete survey, an interested reader can see (Zhang et al., 2004) for *TTP*; (Chan et al., 1997) and (Daoxu et al., 1998) for *MTTP*; (Franchino et al., 2007), (Franchino et al., 2008) and (Franchino et al., 2008a) for *BuST*.
In the following, we compare the schemes performance under the token passing protocols considered in this chapter.

6.1 Performance metric

For evaluating and comparing the performance of different budget allocation schemes several metrics have been proposed. One of the most widely adopted metric is the Worst Case Achievable Utilization (*WCAU*). The *WCAU* of a budget allocation scheme represents the largest utilization (U^*) of the network such that, for any real-time message set whose total network utilization is $U^S \leq U^*$, the budget allocation scheme can guarantee the timeliness of each single real-time message.
The *WCAU* test is useful to guarantee the feasibility of a periodic stream set when only an estimation of the amount of real-time traffic is known (i.e., the maximum time required to send a message) without requiring a detailed characterization of each single real-time message.

6.2 Budget allocation schemes analysis

In the following, we assume $D_i = T_i$ for all the streams, however, the same results can be derived for the case where $D_i < T_i$ by simply substituting D_i to T_i. To make the treatment clearer, when not differently specified, let $\beta_i = T_i/TTRT$, $\beta_{min} = min(T_i)/TTRT$ and $\alpha = \tau/TTRT$. Parameter α represents the bandwidth loss due to the overhead.

Table 3 summarizes the *WCAU*s of the budget allocation schemes considered in this work.

With the *PA* scheme, the *BuST* protocol is the only one having a *WCAU* greater than *0*, whereas with *TTP* and *MTTP* no stream set can be guaranteed.

With the *NPA* scheme, *BuST* and *MTTP* present the same WCAU which depends on β_{min}, hence the smaller *TTRT* the greater is the bandwidth that the protocols can guarantee for the real-time traffic. Since $\beta_{min} \leq 1$ the minimum *WCAU* for the *NPA*, under both *BuST* and *MTTP*, is $(1-\alpha)/2$. Instead, the *TTP* protocols with the *NPA* scheme presents a *WCAU* which does not depends on β_{min} and it is lower than that presented by the other protocols. It is worth observing that, under the *NPA* scheme, the *MTTP* protocol cannot serve best-effort traffic (Franchino et al., 2008) . Conversely, the *BuST* protocol presents the same *WCAU* of *MTTP* and can serve also best-effort traffic.

Under the *EPA* scheme, *BuST* and *MTTP* have a greater *WCAU* with respect to *TTP*. However, the *WCAU* of all the tree protocols is very poor and it depends on the number of nodes. As for the *NPA* scheme, also under the *EPA* scheme the *MTTP* protocol cannot serve non real-time traffic.

BuST, *MTTP* and *TTP* present the same *WCAU* under the *LA* scheme. As for the *NPA* scheme, the *WCAU* under the *LA* scheme depends on β_{min}, i.e. on the choice of *TTRT*. With the *LA* scheme $\beta_{min} \geq 2$, thus the minimum *WCAU* of this scheme is $(1-\alpha)/3$.

With the *MLA* scheme, *TTP* presents a null *WCAU*. Instead, both *BuST* and *MTTP* present a *WCAU* which depends on β_{min} and equivalent to that presented with the *NPA* scheme. As shown in (Franchino et al., 2008a) , with the *MLA* scheme *MTTP* can not serve best-effort traffic when $\sum_{i=1}^{n} H_i \geq TTRT + \tau$.

Allocation schemes	*TTP*	*MTTP*	*BuST*
PA	0	0	$\frac{1-3\alpha}{2(1-\alpha)}$
NPA	$\frac{1-\alpha}{3}$	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$
ELA	$\frac{1-\alpha}{3n-(1-\alpha)}$	$\frac{1-\alpha}{2n-(1-\alpha)}$	$\frac{1-\alpha}{2n-(1-\alpha)}$
LA ($\beta_{min} \geq 2$)	$\left\lfloor \frac{\beta_{\min}-1}{\beta_{\min}+1} \right\rfloor (1-\alpha)$	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$
MLA ($\beta_{min} \geq 1$)	0	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$	$\left\lfloor \frac{\beta_{\min}}{\beta_{\min}+1} \right\rfloor (1-\alpha)$

Table 3. WCAU of the considered schemes.

7. Best-effort traffic service

So far, real-time streams service have been analyzed. This section briefly describes the best-effort service of the *BuST* protocol and its improvements with respect to *MTTP*.

As shown in Section 4, under *MTTP* the maximum time a node can use to deliver non real-time traffic is $T_A = TTRT - \sum_{i=1}^{n} H_i - \tau$. It can be observed that:

- For the PA scheme $T_A = TTRT - \sum_{i=1}^{n} U_i(TTRT - \tau) - \tau = (1 - U^S)(TTRT - \tau)$;
- For the *NPA* scheme $T_A = TTRT - \sum_{i=1}^{n} \frac{U_i^S}{U^S}(TTRT - \tau) - \tau = 0$;
- For the *EPA* scheme $T_A = TTRT - \sum_{i=1}^{n} \frac{TTRT - \tau}{n} = 0$;

This means that *MTTP* is not able to deliver best-effort traffic under both the *NPA* and the *EPA* schemes, while T_A depends on U^S using the *PA* scheme. Moreover, it can be observed that *MTTP* can not serve non real-time traffic when $\sum H_i \geq TTRT + \tau$, that is, when the Protocol Constraint is not met. To verify this last statement, it is sufficient to note that, as stated in Section 3.1, each time a node receives the token it can transmit non real-time traffic for a time no greater than T_A, and since $TTRT - \tau - \sum H_i < 0$, it follows that $T_A = 0$. This means that, when this last condition holds, *MTTP* can starve best effort traffic also with both the *LA* and the *MLA* schemes.
To analyze a worst-case scenario for non real-time service, we assume that each node receiving the token has always best-effort traffic to deliver. In this case, the total channel utilization of the network, including real-time and best-effort traffic, is equal to $1-\alpha$, i.e. the channel is fully utilized.
The following theorem provides the minimum bandwidth that a node i can exploit to deliver non real-time traffic under the *BuST* protocol.

Theorem 7.2 *Under the BuST protocol, a node i can guarantee for the non real-time traffic a minimum bandwidth* U_i^{nrt}*, which depends on the budget allocation scheme adopted. In particular:*

- with *PA*, $U_i^{nrt} = U_i^S \left(\frac{1}{U^S + \frac{\alpha}{1-\alpha}} - 1 \right)$;
- with *NPA*, $U_i^{nrt} = U_i^S \left(\frac{1-\alpha}{U^S} - 1 \right)$;
- with *EPA*, $U_i^{nrt} = \frac{1-\alpha}{n}$;

Proof. See (Franchino et al., 2007), (Franchino et al., 2008) and (Franchino et al., 2008a) .

□

We have shown that, under the *BuST* protocol, non real-time traffic at each node has a minimum bandwidth guaranteed. In addition, it is worth observing that, when not all nodes of the network have to send best-effort traffic, during the round trip of the token, the value of U_i^{nrt} can increase.

8. Simulation results

In this section the performance of *BuST*, *TTP* and *MTTP* is compared by simulation. The simulations have been performed through a discrete-event simulator written for this purpose in C language. The simulation scenario considers a network consisting of 10 nodes. Each node has a periodic stream with a relative deadline ranging from 10 *msec* to 100 *msec*. An infinite amount of non real-time traffic is assumed: this means that, every time a node receives the token, it has some non real-time traffic to deliver. As already stated in Section 7, in this case the total channel utilization $U^{TOT} = U^{nrt} + U^S$ is equal to the total available bandwidth 1-α. Node budgets are assigned using the *PA*, *NPA*, *LA*, and *MLA* budget allocation schemes.

Performance is evaluated through the Maximum Deadline Miss Ratio (*MDMR*), defined as the ratio between the number of messages that miss their deadline and the total number of generated messages. The *MDMR* is measured as a function of the real-time channel utilization U^S, ranging from 0.1 to 1.0 (step of 0.1). For each value of U^S, up to 1000 simulation runs are performed, and the *MDMR* is considered among the runs. A different stream set is generated each run. In particular, for each stream set the utilizations U^S have been generated randomly with a uniform distribution using the method proposed in (Bini & Buttazzo, 2004). For each value of U_i^S, a relative deadline D_i is generated randomly with a uniform distribution in the interval [10, 100] *msec*. Periods are assumed equal to deadlines, i.e for all i $T_i = D_i$. The message lengths C_i have been computed as $C_i = U_i^S D_i$. The overhead τ is assumed equal to 20 *μsec*.

In the simulations, the token is considered as never lost and no ring recovery process is implemented. In this way, a violation of the Protocol Constraint will not compromise the stability of the protocols.

For each scheme, three different scenarios have been considered in the simulations. One where $TTRT = min(D_i)$, one where $TTRT = min(D_i)/2$, and the last with $TTRT = min(D_i)$ and only real-time traffic in the network.

Note that, when there is only real-traffic *BuST*, *TTP* and *MTTP* operate in the same way, that is, they are in practice the same protocol. Therefore, as it will be shown in the following, in case of only real-time traffic all the three protocols present the same performance in terms of *MDMR*.

8.1 Results with the *PA* scheme

In this subsection the results with the *PA* scheme are analyzed. Figure 2 shows the *MDMR* when the *PA* scheme is used to assign the node budgets, and $TTRT = min(D_i)$.

As expected from the results showed in Section 6, as long as $U^S \leq 0.5$, with *BuST* all the messages meet their deadlines (*MDMR* = 0), while *TTP* and *MTTP* present a non-null

deadline miss ratio. For U^S = 0.6, *BuST* presents a very small *MDMR* (equal to 0.5%) that cannot be appreciated in the figure. For $U^S \geq 0.7$, *BuST* presents a significant *MDMR* which is about 76% when U^S = 1, that is, when the channel is over-utilized (remember that the available bandwidth is 1-α).
While *TTP* presents a significant *MDMR* for all values of U^S, *MTTP* presents a *MDMR* lower than that shown by *BuST* for $U^S \geq 0.7$.
Figure 3 shows the *MDMR* with $TTRT = min(D_i)/2$. Notice that, the performance of both *BuST* and *MTTP* improve as expected. Instead *TTP* still presents a very poor performance.
Figure 4 shows the *MDMR* when nodes have only real-time traffic to deliver and $TTRT = min(D_i)$. In this case, as said before, the three protocols operate in the same way, producing the same performance. As long as U^S is not greater than 0.5, there are not deadline misses. With U^S = 0.6 we have a very small *MDMR*, which is equal to 0.69%. For $U^S \geq 0.7$ the *MDMR* starts increasing significantly.

8.2 Results with the *NPA* scheme

Figure 5 reports the *MDMR* when the *NPA* scheme is used to assign node budgets and $TTRT = min(D_i)$.
As long as $U^S \leq 0.5$, *BuST* and *MTTP* have no deadline miss; for $U^S \geq 0.6$, they start experiencing deadline misses. It is worth noticing that, for U^S = 0.6, *BuST* presents a *MDMR* close to 0%, which is not appreciable in the figure.
TTP has deadline misses for all values of U^S; this is due to the fact that *TTP* requires that $TTRT \leq min(D_i)/2$ to work properly.
For $U^S \geq 0.9$, *MTTP* performs better than *BuST*. However, it is worth remembering that, under the *NPA* scheme, *MTTP* is not delivering non real-time traffic. This is the reason why it can provide a better service for real-time traffic.
Figure 6 shows the *MDMR* with $TTRT = min(D_i)/2$. Notice that, with a lower *TTRT,* the performance of all the three protocols improves, as expected by the theoretical analysis showed in the previous sections. In particular, *TTP* has no deadline miss as long as $U^S \leq 0.3$. For U^S = 0.4 and U^S = 0.5, *TTP* presents an *MDMR* close to 0%. For $U^S \geq 0.6$, *TTP* presents an *MDMR* significantly greater than 0%. Notice that, *BuST* and *MTTP* provide more or less the same performance, with the difference that *BuST* serves also non real-time traffic.
Figure 7 shows the *MDMR* when nodes have only real-time traffic to deliver and $TTRT = min(D_i)$.

8.3 Results with the *LA* scheme

Figure 8 shows the *MDMR* when the *LA* scheme is used to assign node budgets and $TTRT = min(D_i)/2$.
As long as $U^S \leq 0.8$, *MTTP* and *BuST* present a null *MDMR*. For U^S = 0.9, they present a *MDMR* not appreciable in the figure, which is less than 0.05% for both protocols. *TTP* presents a null *MDMR* as long as $U^S \leq 0.4$, and a *MDMR* less than 0.3% for $0.5 \leq U^S \leq 0.7$, which is not appreciable in the figure. For $U^S > 0.7$, *TTP* presents a *MDMR* significantly greater than *BuST* and *MTTP*.
Figure 9 shows the *MDMR* when the nodes have only real-time traffic to deliver. As it can be noted, the absence of non real-time traffic improves the protocols performance

considerably. In particular, as long as $U^S \leq 0.8$, the *MDMR* is null; for $U^S = 0.9$, the *MDMR* is still close to 0%, and when $U^S = 1$ the *MDMR* is about 7%.
Finally, it is important to notice that a message deadline miss is due to the Protocol Constraint violation. Furthermore, as highlighted in Section 7, when the Protocol Constraint is not met, or even if the sum of the budgets is equal to *TTRT* - τ, *MTTP* cannot serve non real-time traffic.

8.4 Results with the *MLA* scheme

Figure 10 shows the *MDMR* when the *MLA* scheme is used to assign the node budgets, and TTRT = $min(D_i)$.
As long as $U^S \leq 0.7$, *MTTP* and *BuST* present a null *MDMR*; for $U^S = 0.8$ and for $U^S = 0.9$, the MDMR is less than 1% for both protocols. Under *TTP*, the *MDMR* is non-null for all values of U^S. This is due to the fact that, as stated in Section 6, the Deadline Constraint cannot be satisfied when the *MLA* scheme is used under *TTP*.
Figure 11 shows the MDMR under the *MLA* scheme when *TTRT* = $min(D_i)/2$. For *BuST* and *MTTP*, as long as $U^S \leq 0.8$, the *MDMR* is null. When $U^S = 0.9$, the *MDMR* is not greater than 0.3%, hence is not appreciable in the figure, while when the channel is overloaded, i.e. $U^S = 1$, the *MDMR* is approximately equal to 15%.
Figure 12 depicts the *MDMR* when nodes have only real-time traffic to deliver and *TTRT*= $min(D_i)$. As said before, in this case, all the three protocols present the same performance, and the absence of non real-time traffic improves the protocols performance considerably. In particular, as long as $U^S \leq 0.8$, the *MDMR* is null or very small. For $U^S = 0.9$, the *MDMR* is less than 2%, and when $U^S = 1$ the *MDMR* is close to 5%.
As highlighted in Section 7, when the *MDMR* is not null it means that the Protocol Constraint is not met and, because of this, *MTTP* is not able to deliver non real-time traffic.

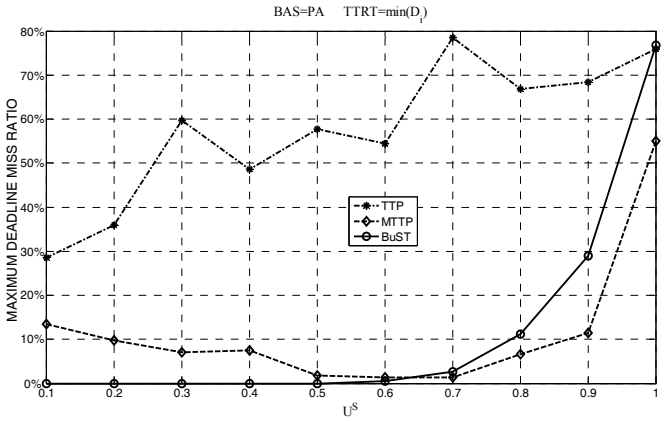


Fig. 2. Maximum Deadline Miss Ratio with the *PA* scheme.

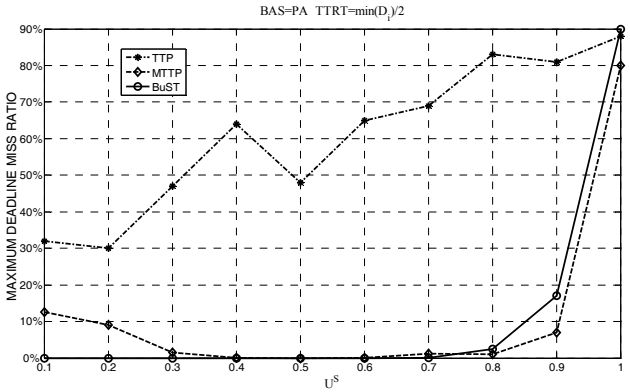


Fig. 3. Maximum Deadline Miss Ratio with the *PA* scheme when $TTRT = min(D_i)/2$.

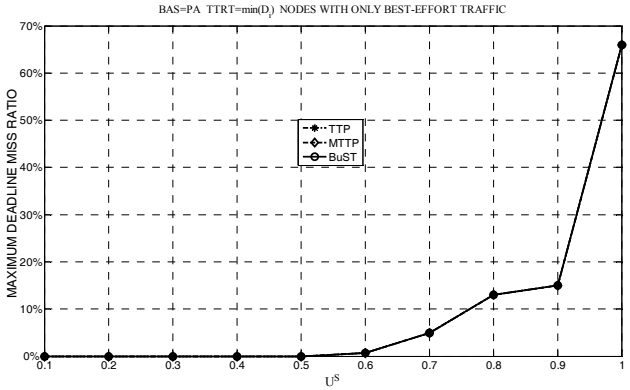


Fig. 4. Maximum Deadline Miss Ratio with the *PA* scheme with only best-effort traffic.

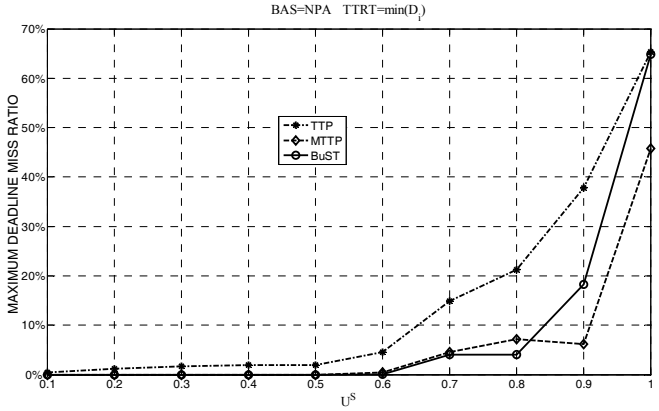


Fig. 5. Maximum Deadline Miss Ratio with the *NPA* scheme.

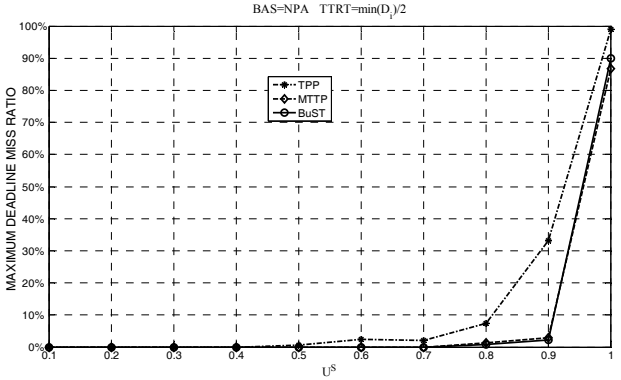


Fig. 6. Maximum Deadline Miss Ratio with the *NPA* scheme when $TTRT = min(D_i)/2$.

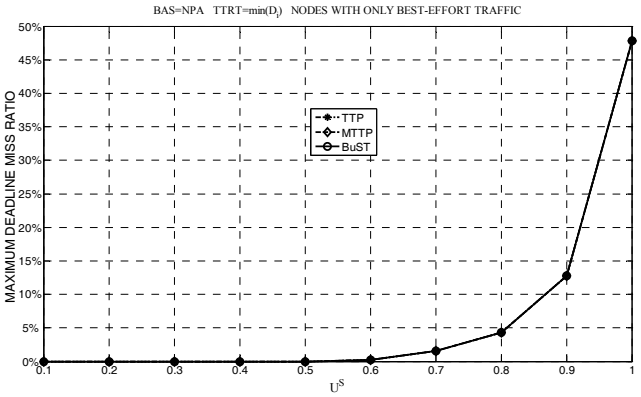


Fig. 7. Maximum Deadline Miss Ratio with the *NPA* scheme with only best-effort traffic.

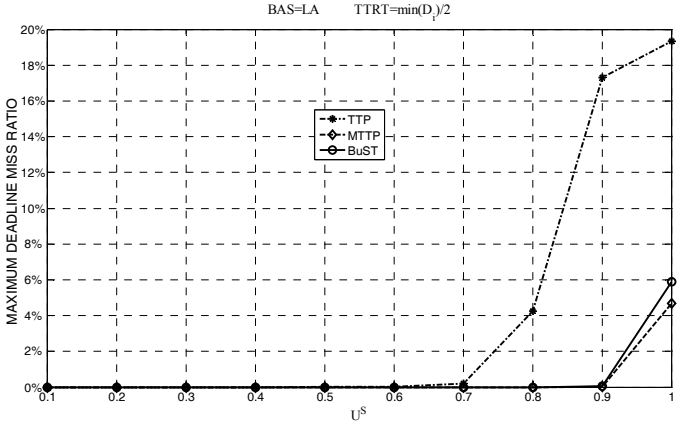


Fig. 8. Maximum Deadline Miss Ratio with the *LA* scheme.

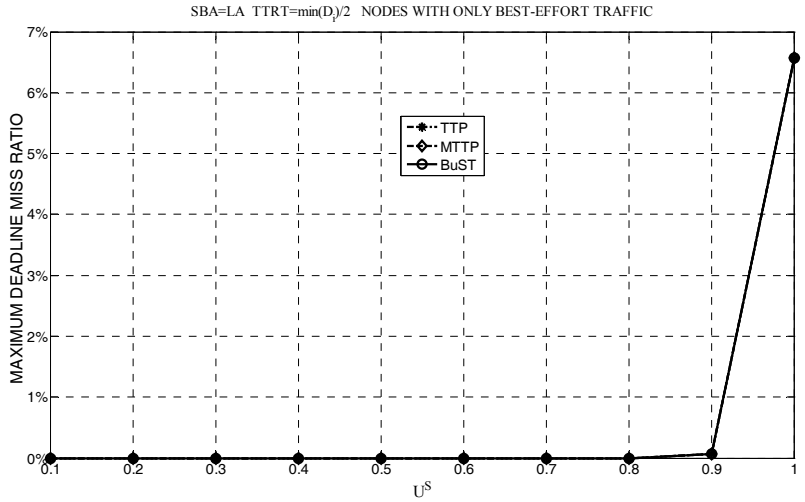


Fig. 9. Maximum Deadline Miss Ratio with the *LA* scheme with only best-effort traffic.

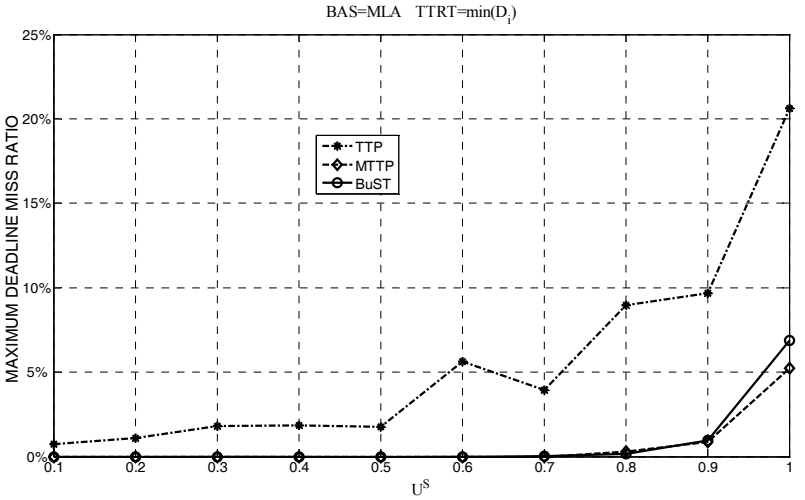


Fig. 10. Maximum Deadline Miss Ratio with the *MLA* scheme.

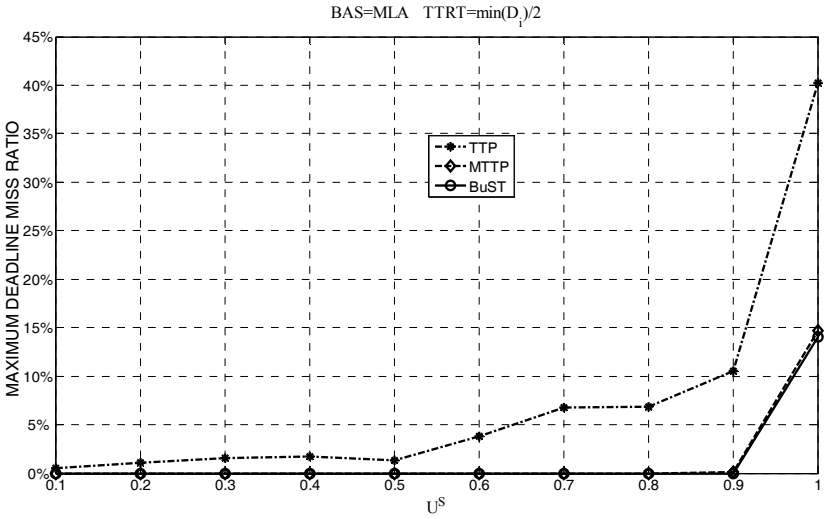


Fig. 11. Maximum Deadline Miss Ratio with the *MLA* scheme when $TTRT = min(D_i)/2$.

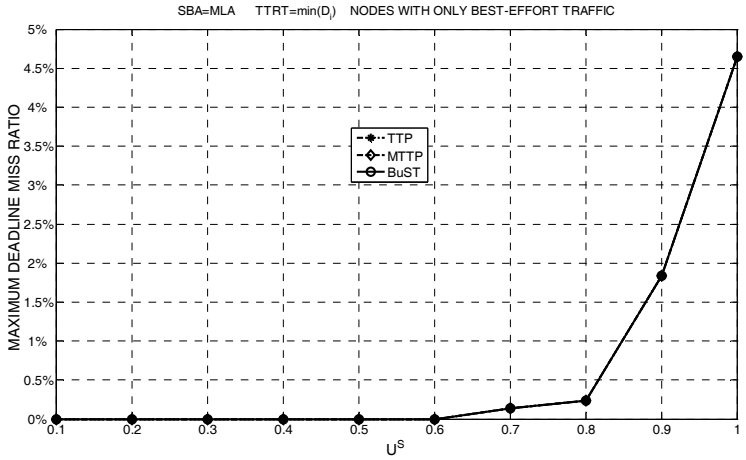


Fig. 12. Maximum Deadline Miss Ratio with the *MLA* scheme with only best-effort traffic.

9. Conclusions

In this work, we introduce and analyzed the *BuST* protocol in comparison with timed token and modified timed token protocols. All the three protocols are discussed in detail, showing their strengths and their drawbacks under different budget allocation schemes. New and well known time properties of the protocols are presented, together with their capability on managing both hard real-time and best-effort traffic. In particular, we have seen that both *BuST* and *MTTP* are superior on serving real-time traffic with respect to the traditional

timed token protocol. However, it has been shown that when both *NPA* and *EPA* schemes are used to assign the node budgets, *MTTP* can starve best-effort traffic. Conversely, the *BuST* protocol can serve also best-effort traffic under all the analyzed budget allocation schemes.
Future work is focused on extending the performance analysis of *BuST* with other allocation schemes available in literature.

10. References

Agrawal, G. ; Chen, B. ; Zhao, W. and Davari, S. (1994). Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Protocol, *IEEE Transaction on* Computers, Vol. 43, No. 3, (March 1994), pp. 327-339, ISSN 0018-9340.

Bini, E. and Buttazzo, G. C. (2005). Measuring Performance of Schedulability Tests, *Real-Time Systems*, Vol. 30, No. 1-2 , (May 2005), pp. 129-154, ISSN 0922-6443.

Chan, E. ; Daoxu, C. ; Cao, J. ; and Lee, C. (1997). Timing Properties of the FDDI-M Medium Access Protocol. *The Computer Journal,* Vol. 40, No. 1, (1997), pp. 43-49, ISSN 0010-4620.

Chen, D. and Zhao, W. (1997). Properties of the Timed Token Protocol. *Technical Report 92-038,* Computer Science Dept., Texas A&M University, October 1992.

Daoxu, C. ; Chan, E. ; and Lee, V. C. S. (1998). Timing Properties of the FDDI-M Medium Access Protocol for a Class of Synchronous Budget Allocation Schemes. *Proceedings of 7-th Int. Conferance on Computer Communications and Networks*, *(ICCCN98)*, pp 825-832, Lafayette, Lousiana, USA, Oct. 1998.

Cicconetti, C. ; Lenzini, L. ; Mingozzi, E. & Stea, G. (2007). An Efficient Cross Layer Scheduler for Multimedia Traffic in Wireless Local Area Networks with IEEE 802.11e HCCA. *ACM Mobile Computing and Communication Review,* Vol. 11, No. 3, (July 2007), pp. 31-46, ISSN 1559-1662.

Fiber Distributed Data Interface (FDDI) (1987). Token Ring Medium Access Control (MAC), May 1987. ANSI Standard X3.139.

Franchino, G.; Buttazzo, G. C., Facchinetti, T. (2007). BuST: Budget Sharing Token Protocol for Hard Real-Time Communication, *Proceedings of 12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA07)*, pp. 1278-1285, Patras (Greece), September 2007, IEEE.

Franchino, G.; Buttazzo, G. C., Facchinetti, T. (2008). Time Properties of the BuST Protocol under the NPA Budget Allocation Scheme, *Proceedings of the Conference on Design, Automation and Test in Europe (DATE 2008)*, pp. 1051-1056, Munich (Germany), March 2008, ACM.

Franchino, G.; Buttazzo, G. C., Facchinetti, T. (2008a). Properties of BuST and Timed Token Protocols in Managing Hard Real-Time Traffic, *Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA08)*, pp. 1205-1212, Hamburg (Germany), September 2008, IEEE.

Grow, R. M. (1982). A Timed Token Protocol for Local Area Networks. *Proceedings of Electro'82,* Paper 17/3, May 1982.

Lenzini, L. ; Mingozzi, E. & Stea, G. (2004). Design and Performance Analysis of the Generalized Timed Token Service Discipline. *IEEE Transaction on Computers,* Vol. 53, No. 7, (July 2004), pp. 879-891, ISSN 0018-9340.

Profibus (1996). General Purpose Field Communication System. European Standard EN50170, CENELEC, Vol. 2/3, Profibus, July 1996.

Sevcik, K. C. and Johnson, M. J. (1987). Cycle time properties of the FDDI token ring protocol. *IEEE Transaction Software Engineering,* Vol. SE-13, No. 3, (1987), pp. 376-385.

Shin, K. G. and Zheng, Q. (1995). FDDI-M: A Scheme to Double FDDI's Ability of Supporting Synchronous Traffic. *IEEE Transaction on Parallel and Distributed Systems,* Vol. 6, No. 11, (November 1995), pp. 1125-1131, ISSN 1045-9219.

Zhang, S. ; Burns, A. ; Chen, J. & Lee, E. (2004). Hard Real-Time communication with the Timed Token Protocol : Current State and challenging Problems. *Real-time Systems,* Vol. 27, No. 3 , (September 2004), pp. 271-295, ISSN 0922-6443.

13

Performance and Reliability of Fault-Tolerant Ethernet Networked Control Systems

Ramez M. Daoud[1], Hassanein H. Amer[1]
and Hany M. ElSayed[2]
[1]*American University in Cairo*
[2]*Cairo University*
Egypt

1. Introduction

In many control applications, networks are being used as a transmission medium for control data such as sensor readings, controller signals, and alarm signals. The resulting control system is termed a Networked Control system (NCS) (Clauset et al., 2008; Hespanha et al., 2007; Yang, 2006). Examples of NCS application areas include industrial automation, building automation, home automation, intelligent vehicle systems, and advanced aircraft and spacecraft. Compared to point-to-point wiring, this approach simplifies wiring in complex systems where several subsystems are interconnected and where sensors and actuators may be physically remote from the controller. System hence becomes easier to control and maintain. Networks also enable communication between several control loops and fault-tolerance through redundancy of components. This chapter summarizes work done by the authors in the area of performance and reliability of networked control systems.
Communication networks were first introduced in digital control systems in the 1970's. Since then, several types of communication networks have been developed to serve this field. Protocols for these networks can be grouped into *fieldbuses* (e.g. FIP and PROFIBUS), *automotive buses* (e.g. CAN), other *machine buses* (e.g. 1553B and the IEC train communication network), *general-purpose networks* (e.g. IEEE LAN's and ATM-LAN) and a number of *research protocols* (e.g. TTP).
In manufacturing applications, the network connecting controllers with sensors and actuators typically constitutes one level in a hierarchy of networks. Fig. 1 illustrates a general network hierarchy model (Lian et al., 2001b). This model consists of five levels, each one having different goals and also different communication capabilities, protocols and complexity. Level one is the device or sensor-actuator level that is used to interconnect controllers, sensors or actuators. Level two is the cell control level and it is designed to be used with cell controllers such as at milling, lathe and control workstations in manufacturing plants. Generally, levels one and two are called *sensor* and *fieldbus*, respectively. Level three is the supervisory level and is used to interconnect machine cells that perform different manufacturing processes. Level four is the plant management level and is used to coordinate various tasks executed inside a plant such as manufacturing

engineering, production management, and resource allocation. Level five is the corporate management level. It may interconnect workstations located in different cities or countries (Lian et al., 2001b).

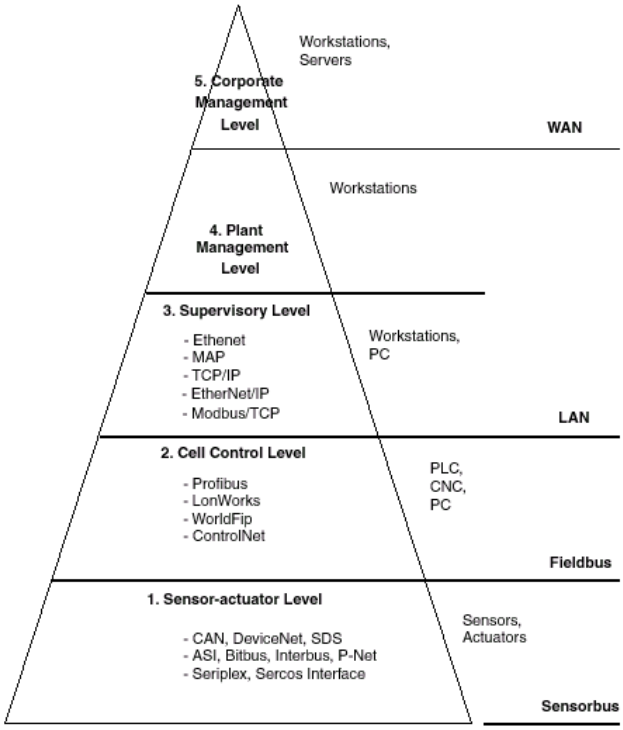


Fig. 1. A general network hierarchy model (Lian et al., 2001b)

A wide variety of network protocols can be used to build an NCS, each suitable for a particular application sector (Thomesse, 2005). However, the use of Ethernet remains a viable and interesting option (Decotignie, 2005). Ethernet is now the dominant local area networking solution in the home and office environments. It is fast, economic and easy to install. Many computerized equipments now come with built-in Ethernet Interfaces. These are some of the reasons why a number of manufacturers of industrial control systems are now migrating to the use of Ethernet on the production floor and integrate it with the management floor (Decotignie, 2005).

A highly desirable requirement is for the office Ethernet communication capability to be fully retained when applying it for control, i.e., the best solution would be if no protocol change were introduced (Daoud et al., 2003; Felser, 2005). Capabilities of Ethernet networks allow us to envision the merger of several hierarchy levels in a single network. A main focus of this research is thus to test the network operation and performance in the presence of mixed traffic.

A major problem in networked control systems is the delays introduced by the network in the control loop. Further problems may be caused by possible loss of data packets. Beyond

certain limits, delays will result in poor system response and tend to destabilize the control loop. Network delays are generally variable and depend on such factors as the used network protocol, network topology, and the amount of network traffic from other sources.
In this chapter the focus is on both the performance and reliability aspects of Ethernet based NCS. The rest of the chapter is organized as follows: Section 2 presents a small survey on previous works done in the area of NCS using Ethernet as communication protocol. Section 3 Introduces the network and its different components. Section 4 describes the model for simulation. Section 5 presents the network simulation results. Sections 6, 7 and 8 discuss the reliability and availability of fault-tolerant production lines. Section 9 concludes this research.

2. Ethernet Networked Control Systems

Abundant research results on the use of Ethernet as a communication network for NCS have been published in the recent years. Interested readers may refer to (Brahimi, 2007; Decotignie, 2005; Felser, 2005; Georges, 2005; Kumar, 2001; Lian et al., 2001a; Marsal, 2006a; Nilsson, 1998; Skeie et al., 2002).

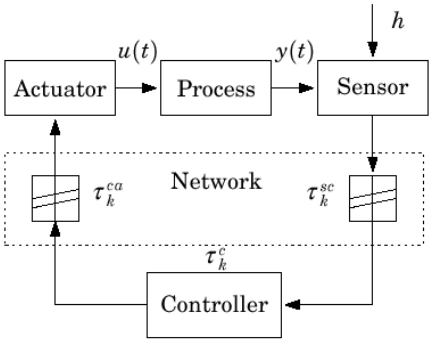


Fig. 2. NCS Block Diagram (Nilsson, 1998)

Automated workcells (Morris, 2005) always include sensors, controllers and actuators connected over the network. The control packet flow on the Ethernet channel is based on publisher/subscriber mode of communication: once the packet is generated, it can be heard by any node on the network. This facilitates the data flow and eliminates the duplication of sensors. A comparison between different methods of communication is given in (Marsal, 2006a). The model of Fig. 2 (Nilsson, 1998) shows a schematic of NCS.
In this model, the physical data is sensed on a periodical basis (clock driven) every h seconds. It is transmitted from the sensor to the controller of the network facing a delay τ_k^{sc}. It is consumed at the controller node and processed over τ_k^c delay. The controller sends the control action over the network to reach the actuator after a delay τ_k^{ca}.
The source of non-determinism in switched Ethernet is queuing delays. For example, the controller node may generate non-real-time traffic (also known as explicit messaging, in contrast with implicit messaging used to designate the real-time control) like FTP sessions or HTTP. This will perturb the queues and the processing loads at the controller node.

Accordingly, the end-to-end delays (the delay measured from the sensor node to the actuator node taking into consideration all kind of encapsulation/de-capsulation, processing and propagation delays) will not be constant. This is what is called mixed traffic environment. It is important to test the Ethernet NCS behavior in simple control environment (only control packets are communicated) and mixed traffic environment.
Early works such as (Meditch & Lea, 1983) tried to modify the medium access sub-layer of CSMA/CD to distinguish between real-time and other traffic packets. Studies were conducted to test stability of the communication channel and to optimize its performance.
Rockwell-Automation studied the use of Ethernet in its switched topology in control and they merged Ethernet with ControlNet to make what is called EtherNet/IP (Ethernet/IP; Lounsbury & Westerman, 2001; ODVA1; ODVA2; ControlNet). By using both TCP/IP and UDP/IP protocols to encapsulate networked messages, both real-time I/O and "explicit messaging" can occur. Also, by providing Ethernet users with real-time I/O, device-configuration, and diagnostic capabilities, along with interoperability and interchangeability, EtherNet/IP provides an Ethernet standard for automation (Ethernet/IP).
In (Walsh & Ye, 2001), a new dynamic scheduling technique for NCS is proposed. The network here is not only dedicated for control purposes, but it can also accommodate communication frames. This gives rise to network induced delays due to unpredictable loads. The control algorithm was made off-line ignoring network delays. This simplified the analysis tremendously. Including time delays is a new approach to validate their work. Also, the simplicity of this approach makes it attractive to be used in general studies for any NCS. The simplicity of the approach in (Walsh & Ye, 2001) comes from the fact that they are using a simple state space representation of the overall NCS.
In (Marsal, 2006a), an analysis is made to define the source of delay in an Ethernet NCS; it shows that the overall response time is the sum of three delays: processing time, waiting time for synchronization of asynchronous processes and waiting time for availability of shared resources. A comparison between two methods to evaluate this response time (simulation and colored Petri nets) can be found in (Marsal et al., 2006b). In this research, the synchronization delays are included as well as the time for availability of shared resources in the processing delays of the nodes. Also, these two major time delays are the focus of other research works (Sundararaman et al., 2005).
In this chapter, the focus is only on switched Ethernet at different speeds. In (Skeie et al., 2002), Fast Ethernet was tested to eliminate the various, usually incompatible, communication networks at the traditional substation automation. This study was conducted to test the possibility of using Fast Ethernet in the switched topology in power station control application. Results of this study were satisfactory within the time frame of the considered application. Because the application presented in (Skeie et al., 2002) had relatively large time frame limit, Fast Ethernet switch topology succeeded to run this system. Later works showed that with more tight timing requirements, especially in mixed traffic environment, the speed of Gigabit Ethernet will be necessary for successful operation.

3. Network Nodes

The network nodes of a typical NCS model are namely: sensors, actuators and controllers. These are the active nodes that generate and consume traffic. Other nodes that are present to

build the network are switches for a switched operation mode. The fabric of the network used in this research is mainly Ethernet at 100Mbps and 1Gbps speeds. This means that there are two types of networks that are tested in this study: Switched Fast Ethernet (100Mbps Ethernet) and Switched Gigabit Ethernet (1Gbps Ethernet).

3.1 Sensor/actuator Networking Level

At this level, networked devices consist essentially of smart sensors, networked controllers, and smart actuators. Smart sensors are nodes that have the capability of data acquisition, intelligence and communication. They acquire proper physical data such as temperature or speed from the industrial environment and have a network-capable application processor to interface with the network. Intelligence gives smart sensors the ability to function independently. Finally to be able to communicate over the network, the sensor must be able to properly encode the information before sending it out on the network.

Smart actuators have the features of actuation, intelligence and communication. They are able to decode the information from the network medium and apply it to the physical devices (Lian et al., 2001b).

Networked controllers have the major function of analyzing the sensor data, making decisions, and giving commands to actuators. The control algorithm should handle decentralized information analysis as well as traditional centralized analysis. Networked controller nodes may also provide a human-machine interface to operators or higher-level managers.

Candidate networks protocols at this level must meet two main criteria: bounded time delay and guaranteed transmission. Unsuccessfully transmitted or large time-delay messages may deteriorate system performance. The system can even become unstable. Several protocols have been proposed to meet these requirements for control systems (Nilsson, 1998). The performance requirements mentioned above are used to determine the capability of the network medium and to provide design specifications to control parameters such as sampling rates as well as network parameters such as communication rates.

3.2 Using Ethernet in the Sensor/Actuator Level

With the use of Ethernet at this level, many things that were not possible in past implementations of NCS will be enabled. Once the industrial floor (the machines network connection) is running on top of Ethernet, it can be interconnected with the management floor (engineering and management network connections). This will help in problem diagnostic and set-up. Now more and more functions can be added. One possibility is on-line system diagnostics and fix-up, by logging into the machine while running in normal operation and setting-up some parameters without the need to stop the operation. This means integration of communication packets (log-on, request/download file, up-load file, log-off) while performing the usual control tasks (traffic of real-time control packets). Moreover, some tasks that can be performed by the operator can be enabled like web-browsing and e-mail check. These tasks add to the communication load that the network handles as an overhead to the pure control load that it is built to support.

3.3 Performance Metric

The performance metrics of network systems that impact control system requirements include: access delay, transmission time, response time, message delay, message collisions (percentage of collision), message throughput (percentage of packets discarded), network utilization, and determinism boundaries. For control systems, candidate control networks generally must meet two main criteria: bounded time delay and guaranteed transmission; i.e., a message should be transmitted successfully within a bounded time delay (Lian et al., 2001b). Unsuccessfully transmitted or large time-delay messages from a sensor to an actuator may deteriorate system performance or make a system unstable. Several protocols have been proposed to meet these requirements for control systems (Nilsson, 1998). The performance metrics mentioned above are used to determine the capability of the network medium and to provide design specifications to control parameters such as sampling rates as well as network parameters such as communication rates (Daoud et al., 2004a).

As in (Georges, 2005), the focus of this research is to use Ethernet IEEE802.3 Std without modifications. A previous study was made to use Ethernet in control by changing the frame structure for real time packets (Tolly, 1997). Another study was made to design a real-time controller to control traffic of the communication medium in case of real-time constraint (Lee & Cho, 2001). More research can be found in (Brahimi et al., 2006; Eker & Cervin, 1999; Georges et al., 2006; Jasperneite & Elsayed, 2004a; Lian et al., 2001a; Vatanski et al., 2006; Wang & Keshav, 1999; Wittenmark et al., 1998; Zhang et al., 2001).

In this research, the system success or failure is evaluated based on measuring the delay faced by the sensor data traveling over the network to reach the controller, the processing delay at the controller node, the propagation delay from the controller to the actuator node faced by the control packet, and finally the processing delay at the actuator node before applying the control word to the physical process. This end-to-end delay takes into consideration all kind of data encapsulation, propagation, de-capsulation and processing in all nodes on the network. End-to-end delay for Ethernet NCS can also be analyzed with network calculus as in (Georges, 2005; Grieu, 2004). The network delay can be expressed as:

$$D_T = D_T\big|_{\text{controller}} + D_T\big|_{\text{actuator}} \tag{1}$$

where D_T is the total end-to-end delay.

$$D_T\big|_{\text{controler}} = T_{ps}\big|_{\text{sensor}} + T_{encap}\big|_{\text{sensor}} + T_p\big|_{\text{sensor}\rightarrow\text{controller}} + T_q\big|_{\text{controller}} + T_{decap}\big|_{\text{controller}} \tag{2}$$

$$D_T\big|_{\text{actuator}} = T_{ps}\big|_{\text{controller}} + T_{encap}\big|_{\text{controller}} + T_p\big|_{\text{controller}\rightarrow\text{actuator}} + T_q\big|_{\text{actuator}} + T_{decap}\big|_{\text{actuator}} \tag{3}$$

Where T_{ps} is the processing delay

T_{encap} is the encapsulation delay

T_p is the propagation delay

T_q is the queuing delay

T_{decap} is the de-capsulation delay

4. Models Description

Many control applications naturally run at very low speeds compared to the speeds of the new network standards. This may imply that the required delays of control packets can be met under realistic loading conditions without handling these packets in a special manner.
Individual machines can be transformed into automated workcells (Soloman, 1994). Networked Control Systems (NCS) make it possible to transform machines into small networks (machine LANs) (Daoud et al., 2003). All sensors are sources of traffic. All actuators are sinks of traffic. Data produced at sensor nodes is sent over the machine network to reach the controller node. At the controller, control information is computed and sent once again over the network to reach the corresponding actuator node. At the actuator, control action is applied on the physical system (the plant).
Sensors and actuators in this scheme are smart. Smart sensors as well as smart actuators have the capability of data encapsulation/de-capsulation. They have network capability to be able to communicate over the machine network. Smart sensors are clock driven nodes: data is sampled at the sensor nodes at constant frequency. This frequency can vary from one sensor to the other depending on the physical quantity it is sensing (temperature is sampled at a lower sampling frequency than speed for example). Once the data is ready at the sensor node, it is encapsulated in network packet format and sent over the machine network.
The controller node, which is an Industrial Personal Computer (IPC), is event driven. When it receives a packet form a sensor and after its de-capsulation and error check, it starts computing the necessary control action. Again, it encapsulates the control word and sends it in packet format to the actuator node. A smart actuator receives the control word and applies the appropriate control action to the system after de-capsulation and error check. Smart actuators are event driven as well (Daoud & Amer, 2007).
Two main Models are presented in this work. The Stand Alone Machine Model and the In-Line Production Model.

4.1 Stand Alone Machine Model Description

Stand Alone Machine Model tests the operation of a single machine (workcell) having all its connections based on Ethernet to implement the Ethernet NCS model.

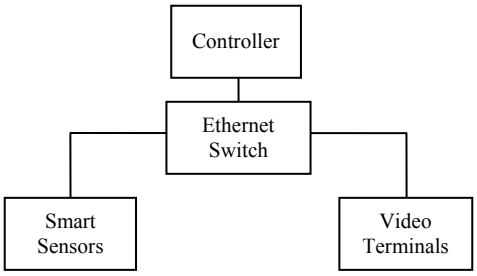


Fig. 3. Stand alone machine model

Two models were built for this study. One model is run on-top-of Fast Ethernet and the other one is run over Gigabit Ethernet for performance comparison. One model consists of 16 sensors, one controller, and 4 actuators, based on the model of (Skeie et al., 2002). In the

following, this model will be referred to as the *light traffic system*. The other model consists of 48 sensors, one controller, and 4 actuators. This model will be referred to as the *heavy traffic system*.
Sensors and actuators are smart. For traditional control using PLCs, 1 revolution per second is encoded into 1,440 electric pulses for electrical synchronization and control. This is why, the system presented in this study is operating at a sampling frequency of 1,440 Hz. Consequently, the system will have a deadline of 694 μs, i.e., a control action must be taken within a frame of 694 μs as round-trip delay originating from the sensor, passing through the controller, and transmitted once more over the network to reach the actuator.
It should be noted that the heavy traffic case should be accompanied by an increase in the processing capabilities of the controller itself. Thus while in the light traffic case the controller was able to process 28,800 packets per second, this number was increased to 74,880 in the heavy traffic case. (These numbers result from multiplying the number of sources and sinks by the sampling rate). The packet delay attributable to the controller will thus be reduced in the heavy traffic case.
OPNET (Opnet) was used as a simulation platform. Real-time generating nodes (smart sensors and smart actuators) were modeled using the "advanced workstation" built-in OPNET model. This model allows the simulation of a node with complete adjustable parameters for operation. The node parameters were properly adjusted to meet the needed task as source of traffic (smart sensor) or sink of traffic (smart actuator). The Controller node was simulated also using "advanced workstation". The Controller node is the administrator in this case: it receives all information from all smart sensors, calculate control parameters, and forward control words to dedicated smart actuators. Producer/ Customer model is finally used to send data from Controller node to smart actuators.
All packets were treated in the switch in a similar manner, i.e., without prioritization. Thus, the packet format of the IEEE 803.2z standard (IEEE, 2000) was used without modification.
Control signals in the simulations are assumed to be UDP packets. Also, the packet size was fixed to minimum frame size in Gigabit Ethernet (520 bytes).
Simulations considered the effect of mixing the control traffic with other types of traffic. These include the option of on-line system diagnostic and fix-up (log-on, request/ download file, up-load file, log-off) as well as e-mail and web-browsing. FTP of 101KB files was considered (Skeie et al., 2002). HTTP, E-mail and telnet traffic was added using OPNET built-in heavy-load models (Daoud et al, 2003).

4.2 In-Line Production Model Description

In many cases, a final product is not produced only on one machine, but, it is handled by several machines in series or in-line. For this purpose, the In-Line Production Model is introduced and investigated. The idea is simply connecting all machine controllers together. Since each individual machine is Ethernet based, interconnecting their controllers (via Ethernet) will enable them to have access to the sensor/actuator level packet flow.
The main function of the controller mounted on the machine is to take charge of machine control. An added task now is to help in synchronization. The controller has the major role of synchronizing several machines in line. This can also be done by connecting the networks of the two machines together. To perform synchronization, the controller of a machine sends its status vector to the controller another machine, and vice versa. Status vector means a complete knowledge of machine information, considering the cam position for example, the

production rate, and so on. These pieces of information are very important for synchronization, especially the production rate. This is because, depending on this statistic, the machines can speed up or slow down to match their respective productions.
A very important metric also, is the fact that the two controllers can back-up data on each other. This is a new added feature. This feature can achieve fault tolerance: in case of a controller failure, the other controller can take over and the machine is not out of service. Although this can slow down the production process, the production is not stopped (Daoud et al., 2004b). Hardware or software failure can cause the failure of one of the controllers. In that case, the information sent by the sensors to the OFF controller is consumed by another operating controller on another machine on the same network (Daoud et al., 2005). "OFF" controller is used instead of failed because the controller can be out of service for preventive maintenance for example. In other words, not only failure of a controller can be tolerated, but regular and preventive maintenance also; because in either cases, failure or maintenance, the controller is out of order.

5. OPNET Network Simulations & Results

First, network simulations have to be performed to validate the concept of Ethernet integration in its switched mode as a communication medium for NCS. OPNET is used to calculate system performance.

5.1 Stand Alone Machine Models Simulation Results

For the light traffic system, and integrating communication as well as control traffic, results for Fast Ethernet are found to be 671 µs round-trip delay in normal operating conditions, and 683 µs round-trip delay as peak value. Results for Gigabit Ethernet are found to be 501 µs round-trip delay in normal operating conditions, and 517 µs round-trip delay as peak value. As the end-to-end delay limit is set to 694 µs (one sampling period), it can be seen that 100Mbps Ethernet is just satisfying the delay requirements while 1Gbps Ethernet is excellent for such system (Daoud et al., 2003).
For the heavy traffic system that consists of 48 smart sensors, 4 smart actuators and one controller, results for Fast Ethernet are found to be 622 µs round-trip delay in normal operating conditions, and 770 µs round-trip delay as peak value. Results for Gigabit Ethernet are found to be 450 µs round-trip delay in normal operating conditions, and 472 µs round-trip delay as peak value. The round-trip delay limit is still 694 µs (one sampling period). It can be seen that 100Mbps Ethernet exceeds the time limit while 1Gbps Ethernet is runs smoothly and can accommodate even more traffic (Daoud et al., 2003).
All measured end-to-end delays include processing, propagation, queuing, encapsulation and de-capsulation delays according to equation 2 (Daoud, 2008).

5.2 In-Line Production Light Traffic Models Simulation Results

The first two simulations consist of two light-traffic machines working in-line with one machine having a failed controller. The failed controller traffic is switched to the operating controller node. One simulation uses Fast Ethernet while the other uses Gigabit Ethernet as communication medium.

Other simulations investigate Gigabit Ethernet performance with more failed controllers on more machines in-line with only one functioning machine controller. In this case, the traffic of the failed controllers is deviated to the operational controller. Other simulations are run to test machine speed increase. As explained in the previous section, the nominal machine speed tested is 1 revolution per second (1,440Hz).
Non-real-time traffic (as in (Daoud et al., 2003)) is added in the three simulations. This is to verify whether or not the system can still function and also if it can accommodate real and non-real-time traffic.
Let the sensors/actuators of the machine with the operational controller be called **near** sensors/actuators. Also, let the sensors/actuators of the machine with the failed controller be called **far** sensors/actuators (Daoud, 2004a).
Results for Fast Ethernet indicate that the delay is too high. The real-time delay a packet faces traveling from the near sensor to the controller and then to the near actuator is around 732 μsec. This is the sum of the delay the real-time packet faces traveling from sensor to controller and the delay it faces traveling from controller to actuator. For the far sensors and actuators, the delay is again too large: around 827 μsec.
Results for Gigabit Ethernet indicate that the delay is small: Only 521 μsec round-trip delay for near nodes (see Fig. 4) and 538 μsec round-trip delay for far nodes.
For three machines with only one controller node operational and running on-top-of Gigabit Ethernet, a round-trip delay of approximately 567 μsec was found for near nodes and approximately 578 μsec round-trip delay for far nodes (Daoud et al., 2004b).
When non-real-time traffic (of the same nature discussed in (Daoud et al., 2003)) is applied in order to jam the control traffic in all three scenarios, a considerable delay is measured. This delay is too large and causes a complete system failure because of the violation of the time constraint of one sampling period. Because of the 3 msec delay that appears in these circumstances with 2 OFF controllers and only 1 ON controller, explicit messaging must be prevented. Explicit messaging here refers to a mixture of non-real-time load of HTTP, FTP, e-mail check and telnet sessions. This is in contrast with "implicit messaging" of real-time control load.

Machine Speed (rps)	Maximum Permissible Delay (μs)	Number of Machines	Number of OFF Controllers	Maximum Measured Delay (μs)
1	694	1	0	501
1	694	2	1	538
1	694	3	2	578
1	694	4	3	682
1	694	5	4	0.266s
1.2	579	3	2	536
1.2	579	4	3	545
1.3	534	2	1	509
1.3	534	3	2	534
1.3	534	4	3	545
1.4	496	1	0	476
1.4	496	2	1	501
1.5	463	1	0	476

Table 1. OPNET Simulation Results for In-Line Light Traffic Machine Model (Daoud et al., 2005)

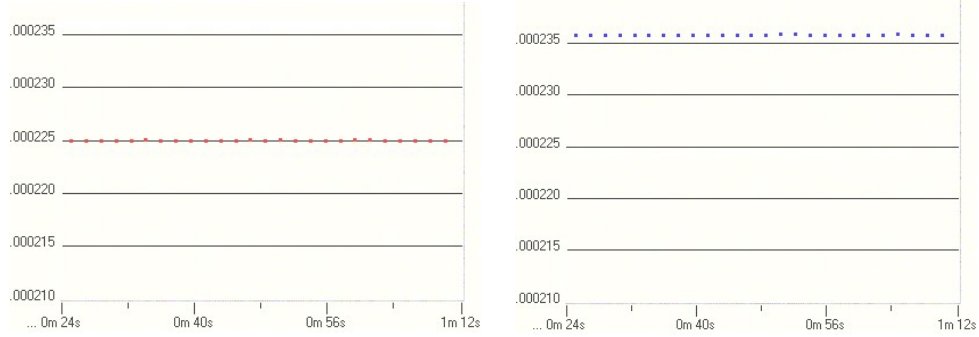
This combination of non-real-time traffic loads simulates a real overhead jamming load introduced by the operator or chief engineer (specially FTP loads). This constraint is quiet acceptable in critical operation and preventing all kinds of non-real-time traffic is a justifiable sacrifice (Daoud et al., 2005). Final results are tabulated in Table 1.

5.3 In-Line Production Heavy Traffic Models Simulation Results

In this section, a simulation study of heavy traffic machines model consisting of 48 sensors, 1 controller and 4 actuators working in-line, is conducted using OPNET. This NCS machine is simulated as switched Star Gigabit Ethernet LAN. Sensors are sources of traffic. The Controller is an intermediate intelligent node. Actuators are sinks of traffic. Having 52 real-time packet generation and consumption nodes (48 sensors and 4 actuators) produces a traffic of 74,800 packet per second on the ether channel. This is because the system is running at a speed of 1 revolution per second (rps) to produce 60 strokes per minute (Bossar). Each revolution is encrypted into 1,440 electric pulses, which means that the sampling frequency is 1,440Hz (sampling period of 694μs). The number of packets (74,800) is the multiplication of the number of nodes (52) by the sampling frequency (1,440) (Daoud et al., 2003).

The most critical scenarios are studied. In these simulations, there is only one active controller while all other controllers on the same line are out of service. Studies for 2, 3 and 4 in-line production machines are done. In all simulations, only one controller is functional and accommodates the control traffic of all 2, 3, or 4 machines on the production line. It was found that the system can tolerate the failure of a maximum of 2 failed controllers in a 3-machine production line. In the case of a 4-machine production line with only one functional controller and 3 failed controllers, the deadline of 694μs (1 sampling period) is violated (Daoud & Amer, 2007).

Accordingly, it is again recommended to disable non-real-time loads during critical mode operation. In other control schemes that do not have the capabilities mentioned in this study, the production line is switched OFF as soon as one controller fails.



Delay at actuator node Delay at controller node

Fig. 4. OPNET Results for Two-Machine Production Line (Heavy Traffic)

In all cases, end-to-end delays are measured. These delays includes all types of data encapsulation/de-capsulation on different network layers at all nodes. They also include

propagation delays on the communication network and the computational delay at the controller node. Results are tabulated in Table 2. Sample OPNET results are shown in Fig. 4.

Machine Speed (rps)	Maximum Permissible Delay (μs)	Number of Machines	Number of OFF Controllers	Maximum Measured Delay (μs)
1	694	2	1	461
1	694	3	2	522
1	694	4	3	1ms
1.1	631	2	1	497
1.1	631	3	2	551
1.2	579	2	1	464
1.2	579	3	2	473
1.3	534	2	1	483
1.3	534	3	2	520
1.4	496	2	1	476
1.4	496	3	2	553
1.5	463	2	1	464

Table 2. OPNET Simulation Results for In-Line Heavy Traffic Machine Model (Daoud & Amer, 2007)

6. Production Line Reliability

In the previous sections, fault-tolerant production lines were described and studied from a communications/control point of view. It was shown, using OPNET simulations, that a production line with several machines working in-line, can work in a degraded mode. Upon the failure of a controller on one of the machines, the tasks of the failed controller are executed by another controller on another machine. This reduces the production line's down time. This section shows how to estimate the Mean Time To Failure (MTTF) and how to use it to find the most cost-effective way of increasing production line reliability.

Consider the following production line; it consists of two machines working in-line. Each machine has a controller, smart sensors and smart actuators. The sampling frequency of each machine is 1,440 Hz.. The machine will fail if the information delay from sensor to controller to actuator exceeds 694 μsec. Also, if one of the two machines fails, the entire production line fails.

In (Daoud et al., 2004b), fault-tolerance was introduced on a system consisting of two such machines. Both machines were linked through Gigabit Ethernet. The Gigabit Ethernet network connected all sensors, actuators and both controllers. It was shown that the failure of one controller on either of the two machines could be tolerated. Special software detected the failure of the controller and transferred its tasks to the remaining functional controller. Non-real-time traffic of FTP, HTTP, telnet and e-mail was not permitted. Mathematical tools are needed to justify this extra cost and prove that production line reliability will increase. One such tool is Markov chains. This will be explained next.

6.1 Markov Model and Mean Time To Failure

Continuous-time Markov models have been widely used to predict the reliability and/or availability of fault-tolerant systems (Billinton & Allan, 1983; Blanke et al., 2006; Johnson, 1989, Siewiorek & Swarz, 1998; Trivedi, 2002). The Markov model describing the system being studied, is shown in Fig. 5. This same model is also found in (Arnold, 1973; Trivedi, 2002). State START is the starting state and represents the error-free situation. If one of the two controllers fails, the system moves from state START to state ONE-FAIL. In this state, both machines are still operating but only one controller is communicating with all sensors and actuators on both machines. If this controller fails before the first one is repaired, the system moves from state ONE-FAIL to state LINE-FAIL. This state is the failure state. The transition rates for the Markov chain in Fig. 5 are explained next.

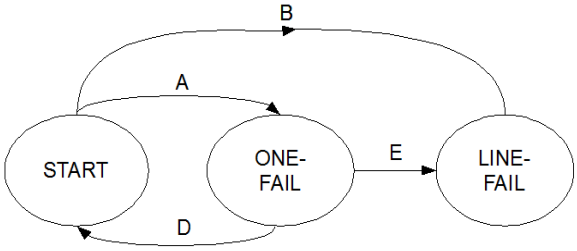


Fig. 5. Markov model

The system will move from state START to state ONE-FAIL when one of the two controllers fails, assuming that the controller failure is detected and that the recovery software successfully transfers control of both machines to the remaining operational controller. Otherwise, the system moves directly from state START to state LINE-FAIL. This explains the transition from state START to state LINE-FAIL. Let c be the probability of successful detection and recovery. In the literature, the parameter c is known as the *coverage* and has to be taken into account in the Markov model. One of the earliest papers that defined the coverage is (Arnold, 1973). It defined the coverage as the proportion of faults from which a system automatically recovers. In (Trivedi, 2002), it was shown that a small change in the value of the coverage parameter had a big effect on system Mean Time To Failure (MTTF). The importance of the coverage was further emphasized in (Amer & McCluskey, 1986, 1987a, 1987b, 1987c). Here, the controller software is responsible for detecting a controller failure and switching the control of that machine to the operational controller on the other machine. Consequently, the value of the coverage depends on the quality of the switching software on each controller.

Assuming, for simplicity, that both controllers have the same failure rate λ, the transition rate from state START to state ONE-FAIL will be equal to $A=2c\lambda$.

As mentioned above, the system will move from state START to state ONE-FAIL if a controller failure is not detected or if the recovery software does not transfer control to the operational controller. A software problem in one of the controllers, for example, can cause sensor data to be incorrectly processed and the packet sent to the actuator will have incorrect data but correct CRC. The actuator verifies the CRC, processes the data and the system fails. Another potential problem that cannot be remedied by the fault-tolerant architecture described here is as follows: Both controllers are operational but their inter-

communication fails. Each controller assumes that the other has failed and takes control of the entire production line. This conflict causes a production line failure. Consequently, the transition rate from state START to state LINE-FAIL will be equal to B=(1-c)2λ.
If the failed controller is repaired while the system is in state ONE-FAIL, a transition occurs to state START. Let the rate of this transition be D=μ. While in state ONE-FAIL, the failure of the remaining controller (before the first one is repaired) will take the system to state LINE-FAIL. Hence, the transition rate from state ONE-FAIL to state LINE-FAIL is equal to E=λ. The Markov model in Fig. 5 can be used to calculate the reliability R(t) of the 1-out-of-2 system under study.

$$R(t) = P_{START}(t) + P_{ONE-FAIL}(t) \tag{4}$$

where $P_{START}(t)$ is the probability of being in state START at time t and $P_{ONE\text{-}FAIL}(t)$ is the probability of being in state ONE-FAIL at time t. The model can also be used to obtain the Mean Time To Failure ($MTTF_{ft}$) of the system. $MTTF_{ft}$ can be calculated as follows (Billinton, 1983): First, the Stochastic Transitional Probability Matrix P for the model in Fig. 5 is obtained:

$$P = \begin{bmatrix} 1-(A+B) & A & B \\ D & 1-(D+E) & E \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where element p_{ij} is the transition rate from state i to state j. So, for example, p_{01} is equal to A=2cλ as in Fig. 5. But state LINE-FAIL is an absorbing state. Consequently, the truncated matrix Q is obtained from P by removing the rightmost column and the bottom row. So,

$$Q = \begin{bmatrix} 1-(A+B) & A \\ D & 1-(D+E) \end{bmatrix} \tag{6}$$

Let matrix $M = [I-Q]^{-1}$

$$M = \begin{bmatrix} (D+E)/L & A/L \\ D/L & (A+B)/L \end{bmatrix} \tag{7}$$

where L = {(A+B)(D+E)}- AD. M is generally defined as the fundamental matrix in which element m_{ij} is the average time spent in state j given that the system starts in state i before being absorbed. Since the system under study starts in state START and is absorbed in state LINE-FAIL,

$$MTTF_{ft} = m_{00} + m_{01} \tag{8}$$

For the system under study in this research,

$$MTTF_{ft} = \frac{A + D + E}{BE + BD + AE} \tag{9}$$

Expanding again in terms of λ, μ and c:

$$MTTF_{ft} = \frac{\lambda + \mu + 2c\lambda}{[(2\lambda)(1-c)][(\lambda + \mu] + [(2c\lambda)(\lambda)]} \tag{10}$$

6.2 Improving MTTF – First Approach

This section shows how to use the Markov model to improve system MTTF in a cost-effective manner. Let the 2-machine fault-tolerant production line described above, have the following parameters:

λ_1: controller failure rate
μ_1: controller repair rate
c_1: coverage

Increasing MTTF can be achieved by decreasing λ_1, increasing μ_1, increasing c_1 or a combination of the above. A possible answer to this question can be obtained by using operations research techniques in order to obtain a triplet ($\lambda_{optimal}$, $c_{optimal}$, $\mu_{optimal}$) that will lead to the highest MTTF. Practically, however, it may not be possible to find a controller with the exact failure rate $\lambda_{optimal}$ and/or the coverage $c_{optimal}$. Also, it may be difficult to find a maintenance plan with $\mu_{optimal}$. Upon contacting the machine's manufacturer, the factory will be offered a few choices in terms of better software versions and/or better maintenance plans. Better software will improve λ and c; the maintenance plan will affect μ. As mentioned above, let the initial value of λ, μ and c be $\{\lambda_1, c_1, \mu_1\}$. Better software will change these values to $\{\lambda_j, c_j, \mu_1\}$ for $2 \le j \le n$. Here, n is the number of more sophisticated software versions. Practically, n will be a small number. Changing the maintenance policy will change μ_1 to μ_k for $2 \le k \le m$. Again, m will be a small number. In summary, system parameters $\{\lambda_1, c_1, \mu_1\}$ can only be changed to a small number of alternate triplets $\{\lambda_j, c_j, \mu_k\}$. If n=3 and m=2, for example, the number of scenarios that need to be studied is *(mn-1)*=5. Running the Markov model 5 times will produce 5 possible values for the improved MTTF. Each scenario will obviously have a cost associated with it. Let

$$\eta = \frac{MTTF_{improved} - MTTF_{old}}{cost}$$

$MTTF_{old}$ is obtained by plugging (λ_1, c_1, μ_1) in the Markov model while $MTTF_{improved}$ is obtained using one of the other 5 triplets. η represents the improvement in system MTTF with respect to cost. The triplet that produces the highest η is chosen.

6.3 Improving MTTF – Second Approach

In this more complex approach, it is shown that λ, μ and c are not totally independent of each other. Let $Q_{software}$ be the quality of the software installed on the controller and let

$Q_{operator}$ represent the operator's expertise. A better version of the software (higher $Q_{software}$) will affect all three parameters simultaneously. Obviously, a better version of the software will have a lower software failure rate, thereby lowering λ. Furthermore, this better version is expected to have more sophisticated error detection and recovery mechanisms. This will increase the coverage c. Finally, the diagnostics capabilities of the software should be enhanced in this better version. This will reduce troubleshooting time, decrease the Repair time and increase μ.

Another important factor is the operator's expertise $Q_{operator}$. The controller is usually an industrial PC (Daoud et al., 2003). The machine manufacturer may be able to supply the hardware and software failure rates but the operator's expertise has to be factored in the calculation of the controller's failure rate on site. The operator does not just use the controller to operate the machine but also uses it for HTTP, FTP, e-mail, etc, beneficiating of its capabilities as a PC. Operator errors (due to lack of experience) will increase the controller failure rate. An experienced operator will make less mistakes while operating the machines. Hence, λ will decrease. Furthermore, an experienced operator will require less time to repair a controller, i.e., μ will increase.

In summary, an increase in $Q_{software}$ produces a decrease in λ and an increase in c and μ. Also, an increase in $Q_{operator}$ reduces λ and increases μ. Next, it is shown how to use $Q_{software}$ and $Q_{operator}$ to calculate λ, μ and c. The parameter λ can now be written as follows:

$$\lambda = \lambda_{hardware} + \lambda_{software} + \lambda_{operator} \tag{11}$$

The manufacturer determines $\lambda_{\text{hardware}}$. In general, let $\lambda_{\text{software}} = f(Q_{software})$. The function f is determined by the manufacturer. Alternatively, the manufacturer could just have a table indicating the software failure rate for each of the software versions. Similarly, let $\lambda_{\text{operator}} = g(Q_{operator})$. The function g has to be determined on site. Regarding the repair rate and the coverage, remember that, for an exponentially-distributed repair time, μ will be the inverse of the Mean Time To Repair (MTTR). There are two cases to be considered here. First, the factory does not stock controller spare parts on premises. Upon the occurrence of a controller failure, the agent of the machine manufacturer imports the appropriate spare part. A technician may also be needed to install this part. Several factors may therefore affect the MTTR including the availability of the spare part in the manufacturer's warehouse, customs, etc. Customs may seriously affect the MTTR in the case of developing countries, for example; in this case the MTTR will be in the order of two weeks. In summary, if the factory does not stock spare parts on site, the MTTR will be dominated by travel time, customs, etc. The effects of $Q_{software}$ and $Q_{operator}$ can be neglected.

Second, the factory does stock spare parts on site. If a local technician can handle the problem, the repair time should be just several hours. However, this does depend on the quality of the software and on the expertise of the technician. The better the diagnostic capabilities of the software, the quicker it will take to locate the faulty component. On the other hand, if the software cannot easily pinpoint the faulty component, the expertise of the technician will be essential to quickly fix the problem. If a foreign technician is needed, travel time has to be included in the repair time which will not be in the orders of several hours anymore. Let

$$\mu = P\{foreign - tech\}\mu_{foreign} + \left(1 - P\{foreign - tech\}\right)\mu_{local} \tag{12}$$

μ_{local} is the expected repair rate in case the failure is repaired locally. μ_{local} is obviously a function of $Q_{software}$ and $Q_{operator}$. Let $\mu_{local} = h(Q_{software}, Q_{operator})$. The function h has to be determined on site. If a foreign technician is required, travel time and the technician's availability have to be taken into account. Again, here, the travel time is expected to dominate the actual repair time on site; in other words, the effects of $Q_{software}$ and $Q_{operator}$ can be neglected. The probability of requiring a foreign technician to repair a failure can be calculated as a first approximation from the number of times a foreign technician was required in the near past. The coverage parameter c has to be determined by the machine manufacturer.
Finally, to calculate the MTTF, the options are not numerous. The production manager will only have a few options to choose from. This approach is obviously more difficult to implement than the previous one. The determination of the functions *f, g* and *h* is not an easy task. On the other hand, using these functions permits the incorporation of the effect of software quality and operator expertise on λ, c and μ. The Markov model is used again to determine the MTTF for each triplet (λ, c, μ) and η determines the most cost-effective scenario. More details can be found in (Amer & Daoud 2006b).

7. Modeling Repair and Calculating Average Speed

The Markov chain in Fig. 5 has an absorbing state, namely state LINE-FAIL. In order to calculate system availability, the Markov chain should not have any absorbing states. System instantaneous availability is defined as the probability that the system is functioning properly at a certain time t. Conventional 1-out-of-2 Markov models usually model the repair as a transition from state ONE-FAIL to state START with a rate μ and another transition from state LINE-FAIL to state ONE-FAIL with a rate of 2μ (assuming that there are two repair persons available) (Siewiorek & Swarz, 1998). If there is only one repair person available (which is the realistic assumption in the context of developing countries), the transition rate from state LINE-FAIL to state ONE-FAIL is equal to μ. Figure 6 is the same Markov model as in Fig. 5 except for the extra transition from state LINE-FAIL back to state START. This model has a better representation of the repair policies in developing countries. In this improved model, the transition from state LINE-FAIL to state ONE-FAIL is cancelled. This is more realistic, although unconventional. Since most of the repair time is really travel time (time to import spare parts or time for a specialist to travel to the site), the difference in the time to repair one controller or two controllers will be minimal. In this model, the unavailability is equal to the probability of being in state LINE-FAIL while the availability is equal to the sum of the probabilities of being in states START and ONE-FAIL. These probabilities are going to be used next to calculate the average operating speed of the production line.
In (Daoud et al., 2005), it was found that a fully operational fault-tolerant production line with two machines can operate at a speed of 1.4S where S is the normal speed (1 revolution per minute as mentioned above). If one controller fails, the other controller takes charge of its duties and communicates with all sensors and actuators on both machines. The maximum speed of operation in this case was 1.3S. Assuming λ is not affected by machine speed, the average steady state speed $\boldsymbol{Speed\_Av_{ss}}$ will be equal to:

$$Speed\_Av_{ss} = (P_{STARTss})\,(1.4S) + (P_{ONE-FAILss})\,(1.3S) \tag{13}$$

where $P_{STARTss}$ and $P_{ONE\text{-}FAILss}$ are the steady state probabilities of being in states START and ONE-FAIL respectively. If the machines had been operated at normal speed,

$$Speed\_Av_{ss} = (P_{STARTss})(S) + (P_{ONE-FAILss})(S) \tag{14}$$

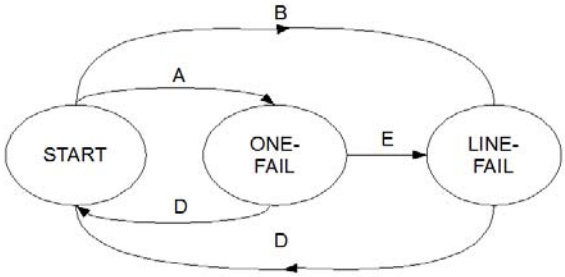


Fig. 6. Improved Markov model

Equations 13 and 14 can be used to estimate the increase in production when the machines are operated at higher-than-normal speeds. It is important to note here that machines are not usually operated at their maximum speed on a regular basis but only from time to time in order to obtain a higher turn-over. More information regarding this topic can be found in (Amer et al., 2005).

8. TMR Sensors

In the production line studied above, the sensors, switches and actuators were single points of failure. Introducing redundancy at the controller level may not be enough if the failure rate of the sensors/switches/actuators is relatively high especially since there are 32 sensors, 8 actuators, 3 switches and just two controllers. Introducing fault tolerance at the sensor level will certainly increase reliability. Triple Modular Redundancy (TMR) is a well-known fault tolerance technique (Johnson, 1989; Siewiorek & Swarz, 1998). Each sensor is triplicated. The three identical sensors send the same data to the controller. The controller compares the data; if the three messages are within the permissible tolerance range, the message is processed. If one of the three messages is different than the other two, it is concluded that the sensor responsible for sending this message has failed and its data is discarded. One of the other two identical messages is processed. This is known as masking redundancy (Johnson, 1989; Siewiorek & Swarz, 1998). The system does not fail even though one of its components is no longer operational. Triplicating each sensor in a light-traffic machine means that the machine will have 48 (=16*3) sensors, one controller and 4 actuators. The first important consequence of this extra hardware is the increased traffic on the network. The number of packets produced by sensors will be tripled. A machine with 48 sensors, one controller and 4 actuators was simulated and studied (Daoud et al. 2003); this is the *heavy-traffic* machine. The OPNET simulations in (Daoud et al., 2003) indicated that Gigabit Ethernet was able to accommodate both control and communication loads. Another important issue regarding the triplication of the sensors is cost-effectiveness. From a reliability point of view, triplicating sensors is expected to increase the system Mean Time

Between Failures (MTBF) and consequently, decrease the down time. However, the cost of adding fault tolerance has to be taken into account. This cost includes the extra sensors, the wiring, bigger switches and software modifications. The software is now required to handle the "voting" process; the messages from each three identical sensors have to be compared. If the three messages are within permissible tolerance ranges, one message is processed. If one of the messages is different from the other two, one of the two valid messages is used. The sensor that sent the corrupted message is disregarded till being repaired. If a second sensor from this group fails, the software will not be able to detect which of the sensors has failed and the production line has to be stopped. It is the software's responsibility to alert the operator using Human Machine Interface (HMI) about the location of the first malfunctioning sensor and to stop the production line upon the failure of the second sensor. System reliability is investigated next in order to find out whether or not the extra cost is justified.

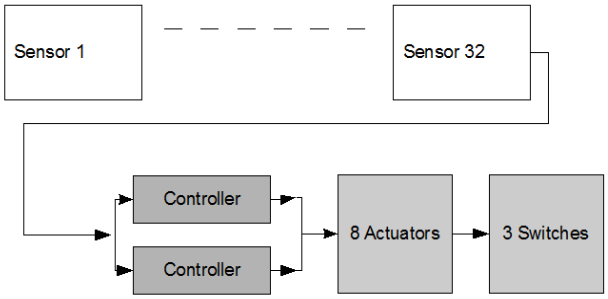


Fig. 7. RBD for Two-Cont Configuration

Reliability Block Diagrams (RBDs) can be used to calculate system reliability (Siewiorek & Swarz, 1998). Three configurations will be studied and compared. In the first configuration, there is no fault tolerance. Any sensor, controller, switch or actuator on either machine is a single point of failure. For exponentially-distributed failure times, the system failure rate is the sum of the failure rates of all its components. Let this configuration be the *Simplex* configuration. If fault tolerance is introduced at the controller level only (as in (Daoud et al., 2004b)), this configuration will be called *Two-Cont*. Figure 7 shows the RBD of the Two-Cont production line with two light-traffic machines. It is clear that fault tolerance only exists at the controller level. Figure 8 describes the RBD of the same production line but with two heavy-traffic machines. Now, every sensor is a TMR system and will fail when two of its sensors fail (2/3 system). Let this configuration be called the *TMR* configuration. Only the actuators and the switches constitute single points of failure. Instead of calculating system reliability, another approach is taken here, namely the Mission Time (**MT**). $MT(r_{min})$ is the time at which system reliability falls below r_{min} (Johnson, 1989; Siewiorek & Swarz, 1998). r_{min} is determined by production management and represents the minimum acceptable reliability for the production line. The production line will run continuously for a period of MT. Maintenance will then be performed; if one of the controllers has failed, it is repaired as well as any failed sensor. r_{min} is chosen such that the probability of having a system failure during MT is minimal.

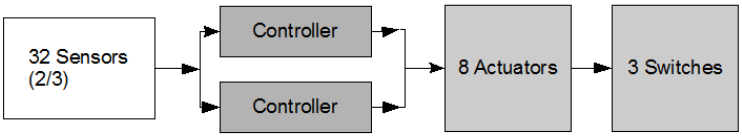


Fig. 8. RBD for TMR Configuration

It is assumed here that the production line is totally fault-free after maintenance. If r_{min} is high enough, there will be no unscheduled down time and no loss of production. Of course, if r_{min} is very high, MT will decrease and the down time will increase. Production can of course be directly related to cost. Let R_{line} be the reliability of the production line. R_{sensor}, R_{switch}, $R_{controller}$ and $R_{actuator}$ will be the reliabilities of the sensor, switch, controller and actuator, respectively. For exponentially-distributed failure times: $R = e^{-\lambda t}$. R is the component reliability (sensor, controller, ...) and λ is its failure rate (which is constant (Johnson, 1989; Siewiorek & Swarz, 1998)). Assume for simplicity that the switches are very reliable when compared to the sensors, actuators or controllers and that their probability of failure can be neglected. Furthermore, assume that all sensors on both machines have an identical reliability. The same applies for the controllers and the actuators. Next, the reliabilities of the production line will be calculated for the three configurations: Simplex, Two-Cont and TMR.

In the Simplex mode, there is no fault tolerance at all and any sensor, controller or actuator failure causes a system failure. Hence:

$$R_{line} = (R_{sensor}^{32})(R_{controller}^{2})(R_{actuator}^{8}) \tag{15}$$

Remember that each machine has 16 sensors, one controller and 4 actuators and the system (production line) consists of two machines. If fault tolerance is introduced at the controller level (as in (Daoud et al., 2004b)

$$R_{line} = \left(R_{sensor}^{32}\right)\left(1-\left(1-R_{controller}\right)^{2}\right)\left(R_{actuator}^{8}\right) \tag{16}$$

The next level of fault tolerance is the introduction of Triple Modular Redundancy at the sensor level. Each of the 32 sensors will now be a sensor assembly that consists of three identical sensors. Hence

$$R_{line} = \left(3R_{sensor}^{2} - 2R_{sensor}^{3}\right)^{32}\left(1-\left(1-R_{controller}\right)^{2}\right)\left(R_{actuator}^{8}\right) \tag{17}$$

Equations 15, 16 and 17 are then used to determine MT for a specific value of R_{line} for each of the three configurations. Hence, the cost-effectiveness of the added fault-tolerance can be quantitatively examined. More details can be found in (Amer & Daoud, 2008).

9. Conclusion

This chapter has discussed the performance and reliability of fault-tolerant Ethernet Networked Control Systems. The use of Gigabit Ethernet in networked control systems was investigated using the OPNET simulator. Real-time traffic and non-real time traffic were integrated without changing the IEEE 802.3 protocol packet format. In a mixed traffic industrial environment, it was found that standard Gigabit Ethernet switches succeeded in meeting the required time constraints. The maximum speed of operation of individual machines and fault tolerant production-lines was also studied.
The reliability and availability of fault tolerant production lines was addressed next. It was shown how to use Markov models to find the most cost-effective way of increasing the Mean Time To Failure MTTF. Improved techniques for modeling repair were also discussed. Finally, it was shown how to introduce fault tolerance at the sensor level in order to increase production line mission time.

10. References

Amer, H.H. & McCluskey, E.J. (1986). "Calculation of the Coverage Parameter for the Reliability Modeling of Fault-tolerant Computer Systems", *Proc. Intern. Symp. on Circuits and Systems ISCAS*, pp. 1050-1053, San Jose, CA, U.S.A., May 1986.

Amer, H.H. & McCluskey, E.J. (1987a). "Weighted Coverage in Fault-tolerant Systems", *Proc. Reliability and Maintainability Symp. RAMS*, pp.187-191, Philadelphia, PA, U.S.A., January 1987.

Amer, H.H. & McCluskey, E.J. (1987b). "Latent Failures and Coverage in Fault-tolerant Systems", *Proc. Phoenix Conf. on Computers and Communications*, Scottsdale, pp. 89-93, AZ, U.S.A., February 1987.

Amer, H.H. & McCluskey, E.J. (1987c). "Calculation of Coverage Parameter", *IEEE Trans. Reliability*, June 1987, pp. 194-198.

Amer, H.H.; Moustafa, M.S. & Daoud, R.M. (2005). "Optimum Machine Performance In Fault-Tolerant Networked Control Systems", *Proceedings of the IEEE EUROCON Conference*, pp. 346-349, Belgrade, Serbia & Montenegro, November 2005.

Amer, H.H.; Moustafa, M.S. & Daoud, R.M. (2006a). "Availability Of Pyramid Industrial Networks", *Proceedings of the Canadian Conference on Electrical and Computer Engineering CCECE,* pp. 1862-1865, Ottawa, Canada, May 2006.

Amer, H.H. & Daoud, R.M. (2006b). "Parameter Determination for the Markov Modeling of Two-Machine Production Lines" *Proceedings of the International IEEE Conference on Industrial Informatics INDIN,* pp. 1178-1182, Singapore, August 2006.

Amer, H.H. & Daoud, R.M. (2008). "Increasing Network Reliability by Using Fault-Tolerant Sensors", *International Journal of Factory Automation, Robotics and Soft Computing,* January 2008, pp. 71-76.

Arnold, T.F. (1973). "The concept of coverage and its effect on the reliability model of a repairable system," *IEEE Trans. On Computers,* vol. C-22, No. 3, March 1973.

Baillieul, J. & Antsaklis, P.J. (2007). "Control and Communication Challenges in Networked Real-Time Systems", *Proceedings of the IEEE,* Vol. 95, No. 1, January 2007, pp. 9-28.

Billinton, R. & Allan, R. (1983) "*Reliability Evaluation of Engineering Systems: Concepts and Techniques*", Pitman.

Blanke, M.; Kinnaert, M.; Lunze, J. & Staroswiecki, M. (2006). "*Diagnosis and Fault-Tolerant Control"*, Springer-Verlag.

Bossar Horizontal Machinery. Official Site: www.bossar.es

Brahimi, B.; Aubrun, C. & Rondeau, E. (2006). "Modelling and Simulation of Scheduling Policies Implemented in Ethernet Switch by Using Coloured Petri Nets," *Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation ETFA*, Prague, Czech Republic, September 2006.

Brahimi, B. (2007). "Proposition d'une approche intégrée basée sur les réseaux de Petri de Haut Niveau pour simuler et évaluer les systèmes contrôlés en réseau," PhD Thesis, Université Henri Poincaré, Nancy I, December 2007.

Bushnell, L. (2001). "Networks and Control", *IEEE Control Systems Magazine*, vol. 21, no. 1, 2001, pp. 22-23.

Clauset, A., Tanner, H.G., Abdallah, C.T., & Byrne, R.H. (2008). "Controlling Across Complex Networks – Emerging Links Between Networks and Control", *Annual Reviews in Control* , Vol. 32, No. 2, pp. 183–192, December 2008.

ControlNet, Official Site: http://www.controlnet.org

Daoud, R.M.; Elsayed, H.M.; Amer, H.H. & Eid, S.Z. (2003). "Performance of Fast and Gigabit Ethernet in Networked Control Systems," *Proceedings of the IEEE International Mid-West Symposium on Circuits and Systems, MWSCAS*, Cairo, Egypt, December 2003.

Daoud, R.M. (2004a). *Performance of Gigabit Ethernet in Networked Control Systems*, MSc Thesis, Electronics and Communications Department, Faculty of Engineering, Cairo University, 2004.

Daoud, R.M.; Elsayed, H.M. & Amer, H.H. (2004b). "Gigabit Ethernet for Redundant Networked Control Systems, *Proceedings of the IEEE International Conference on Industrial Technology ICIT*, December 2004, Hammamet, Tunis.

Daoud, R.M., Amer, H.H. & Elsayed, H.M. (2005). "Fault-Tolerant Networked Control Systems under Varying Load," *IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, SMCia,* Espoo, Finland, June 2005.

Daoud, R.M. & Amer, H.H. (2007). "Ethernet for Heavy Traffic Networked Control Systems", *International Journal of Factory Automation, Robotics and Soft Computing,* January 2007, pp. 34-39.

Daoud, R.M. (2008). *Wireless and Wired Ethernet for Intelligent Transportation Systems*, DSc Dissertation, LAMIH-SP, Universite de Valenciennes et du Hainaut Cambresis, France, 2008.

Decotignie, J.-D. (2005). "Ethernet-Based Real-Time and Industrial Communications," *Proceedings of the IEEE*, vol. 93, No. 6, June 2005.

Eker, J. & Cervin, A. (1999). "A Matlab Toolbox for Real-Time and Control Systems Co-Design," *6th International Conference on Real-Time Computing Systems and Applications*, Hong Kong, P.R. China, December 1999.

EtherNet/IP Performance and Application Guide, Allen-Bradley, Rockwell Automation, Application Solution.

Felser, M. (2005). "Real-Time Ethernet – Industry Prospective," *Proceedings of the IEEE*, vol. 93, No. 6, June 2005.

Georges, J.-P. (2005). "Systèmes contrôles en réseau: Evaluation de performances d'architectures Ethernet commutées," PhD thesis, Centre de Recherche en Automatique de Nancy CRAN, November 2005.

Georges, J.P.; Vatanski, N.; Rondeau, E. & Jämsä-Jounela, S.-L. (2006). "Use of Upper Bound Delay Estimate in Stability Analysis and Robust Control Compensation in Networked Control Systems," *12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM*, St-Etienne, France, May 2006.

Grieu, J. (2004). "Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques," PhD Thesis, Institut National Polytechnique de Toulouse, Ecole doctorale informatique et telecommunications, September 2004.

IEEE Std 802.3, 2000 Edition

Jasperneite, J. & Elsayed, E. (2004). "Investigations on a Distributed Time-triggered Ethernet Realtime Protocol used by PROFINET," *3rd International Workshop on Real-Time Networks (RTN 2004)*, Catania, Sicily, Italy , Jun 2004.

Johnson, B. W. (1989). "*Design and Analysis of Fault-Tolerant Digital Systems*", Addison-Wesley.

Hespanha, J.P. , Naghshtabrizi, P. & Xu, Y (2007). "A Survey of Recent Results in Networked Control Systems", *Proceedings of the IEEE,* Vol. 95, No. 1, January 2007, pp. 138-162.

Kumar, P.R. (2001). "New Technological Vistas for Systems and Control: The Example of Wireless Networks," *IEEE Control Systems Magazine*, vol. 21, no. 1, 2001, pp. 24-37.

Lee, S.-H. & Cho, K.-H. (2001). "Congestion Control of High-Speed Gigabit-Ethernet Networks for Industrial Applications," *Proc. IEEE ISIE,* Pusan, Korea, pp. 260-265, June 2001.

Lian, F.L.; Moyne, J.R. & Tilbury, D.M. (1999). "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet," Tech. Rep. UM-MEAM-99-02, February 1999. Available: http://www.eecs.umich.edu/~impact

Lian, F.L.; Moyne, J.R. & Tilbury, D.M. (2001a). "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Systems Magazine*, Vol. 21, No. 1, pp.66-83, February 2001.

Lian, F.L.; Moyne, J.R. & Tilbury, D.M. (2001b). "Networked Control Systems Toolkit: A Simulation Package for Analysis and Design of Control Systems with Network Communication," Tech. Rep., UM-ME-01-04, July 2001.
Available: http://www.eecs.umich.edu/~impact

Lounsbury, B. & Westerman, J. (2001). "Ethernet: Surviving the Manufacturing and Industrial Environment," Allen-Bradley white paper, May 2001.

Marsal, G. (2006a). "Evaluation of time performances of Ethernet-based Automation Systems by simulation of High-level Petri Nets," *PhD Thesis,* Ecole Normale Superieure De Cachan, December 2006.

Marsal, G.; Denis, B.; Faur, J.-M. & Frey, G. (2006b). "Evaluation of Response Time in Ethernet-based Automation Systems," *Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, Prague, Czech Republic, September 2006, pp. 380-387.

Meditch, J.S. & Lea, C.-T. (1983). "Stability and Optimization of the CSMA and CSMA/CD Channels," *IEEE Trans. Comm.*, Vol. 31, No. 6 , June 1983, pp. 763-774.

Morriss, S.B. (1995). "*Automated Manufacturing Systems Actuators, Controls, Sensors, and Robotics*", McGraw-Hill.

Nilsson, J., "Real-Time Control Systems with Delays," PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.

ODVA, "Volume 1: CIP Common," Available: http://www.odva.org/10_2/03_events/03_ethernet-homepage.htm

ODVA, "Volume 2: EtherNet/IP Adaptation on CIP," Available: http://www.odva.org/10_2/03_events/03_ethernet-homepage.htm

Opnet, Official Site for OPNET http://opnet.com

Siewiorek, D.P. & Swarz, R.S. (1998). "*Reliable Computer Systems – Design and Evaluation,*" A K Peters, Natick, Massachusetts.

Skeie, T.; Johannessen, S. & Brunner, C. (2002). "Ethernet in Substation Automation," *IEEE Control Syst.*, Vol. 22, no. 3, June 2002, pp. 43-51.

Soloman, S. (1994). "*Sensors and Control Systems in Manufacturing*," McGraw-Hill.

Sundararaman, B.; Buy, U. & Kshemkalyani, A.D. (2005). "Clock Synchronization for Wireless Sensor Networks: a survey," Ad Hoc Networks, vol. 3, 2005, pp. 281-323.

Thomesse, J.-P. (2005). "Fieldbus Technology in Industrial Automation", *Proceedings of the IEEE,* Vol. 93, No. 6, June 2005, pp. 1073-1101.

Tolly, K. (1997). "*The Great Networking Correction: Frames Reaffirmed,*" Industry Report, The Tolly Group, IEEE Internet Computing, 1997.

Trivedi, K.S. (2002). "*Probability and Statistics with Reliability, Queuing, and Computer Science Applications",* Wiley, New York.

Vatanski, N.; Georges, J.P.; Aubrun, C.; Rondeau, E. & Jämsä-Jounela, S.-L. (2006). "Control Compensation Based on Upper Bound Delay in Networked Control Systems," *17th International Symposium on Mathematical Theory of Networks and Systems, MTNS*, Kyoto, Japan, July 2006.

Walsh, G.C. & Ye, H. (2001). "Scheduling of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, February 2001, pp. 57-65.

Wang, J. & Keshav, S. (1999). "Efficient and Accurate Ethernet Simulation," Cornell Network Research Group (C/NRG), Department of Computer Science, Cornell University, May 1999.

Wittenmark, B.; Bastian, B. & Nilsson, J. (1998). "Analysis of Time Delays in Synchronous and Asynchronous Control Loops," Lund Institute of Technology, 37th CDC, Tampa, December 1998.

Yang, T.C. (2006). "Networked Control System: a Brief Survey", *IEE Proceedings-Control Theory and Applications.*, Vol. 153, No. 4, July 2006, pp. 403-412.

Zhang, W.; Branicky, M.S. & Phillips, S.M. (2001). "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, February 2001, pp. 84-99.

14

Study of event-based sampling techniques and their influence on greenhouse climate control with Wireless Sensors Network

Andrzej Pawlowski, José L. Guzmán, Francisco Rodríguez, Manuel Berenguel, José Sánchez and Sebastián Dormido
Department of Languages and Computation, University of Almería
Department of Computer Science and Automatic Control, UNED
Spain

1. Introduction

During last years, event-based sampling and control are receiving special attention from researchers in wireless sensor networks (WSN) and networked control systems (NCS). The reason to deserve this attention is due to event-based strategies reduce the exchange of information between sensors, controllers, and actuators. This reduction of information is equivalent to extend the lifetime of battery-powered wireless sensors, to reduce the computational load in embedded devices, or to cut down the network bandwidth (Miskowicz, 2005).

Event-based systems are becoming increasingly commonplace, particularly for distributed real-time sensing and control. A characteristic application running on an event-based operating system is that where state variables are updated asynchronously in time, e.g., when an event of interest is detected or because of delays in the computation and/or communication tasks (Sandee, 2005). Event-based control systems are currently being presented as solutions to many control problems (Arzen, 1999); (Sandee, 2005); (Miskowicz, 2005); (Astrom, 2007); (Henningsson et al., 2008). In event-based control systems, it is the proper dynamic evolution of system variables what decides when the next control action will be executed, whereas in a time-based control system, the autonomous progression of the time is what triggers the execution of control actions (Astrom & Wittenmark 1997). Current distributed control systems impose restrictions on the system architecture that makes difficult the adoption of a paradigm based on events activated per time. Especially, in the case of closed-loop control using computer networks or buses, as happens with field buses, local area networks, or even Internet. An alternative to these approaches consists of using event-based controllers that are not restricted to the synchronous occurrence of controller actions. The utilization of synchronous sampling period is one of the severest conditions that control engineers impose on the software implementation. As discussed

above, in an event-based control system the control actions are executed in an asynchronous way, that is, the sampling period is governed by system events and it is called event-based sampling. The event-based sampling indicates that the most appropriate method of sampling consists of transmitting information only when a significant change happens in the signal that justifies the acquisition of a new sample. Researchers have demonstrated special interest on these sampling techniques (Vasyuntynskyy & Kabitzsch, 2006); (Miskowicz, 2007); (Suh, 2007) (Dormido et al., 2008). Nowadays, commercial systems present more flexibility in the implementation of control algorithms and sampling techniques, especially WSN, where each node of the network can be programmed with a different sampling or local control algorithm with the main goal of optimizing the overall performance. This kind of solution allows control engineers to distribute the control process, considering centralized supervision of all variables, thanks to the application of wireless communications. Furthermore, remote monitoring and control through data-communication networks are very popular for process supervision and control (Banatre at al., 2008). The usage of networks provides many well-known benefits, but it also presents some limitations in the amount of transmitted data. This fact is especially visible in WSN, where the bandwidth of the communication channels is limited and typically all nodes are battery-powered. Event-based sampling techniques appear as possible solutions to face this problem allowing considerably saving of network resources and reducing the power consumption. On the other hand, the control system performance is highly affected due to the event-based sampling techniques, being necessary to analyze and study a compromise between control quality and reduction in the control signal commutations.

The agro-alimentary sector is incorporating new technologies due to the large production demands and the diversity, quality, and market presentation requirements. A technological renovation of the sector is being required where the control engineering plays a decisive role. Automatic control and robotics techniques are incorporated in all the agricultural production levels: planting, production, harvesting and post-harvesting processes, and transportation. Modern agriculture is subjected to regulations in terms of quality and environmental impact, and thus it is a field where the application of automatic control techniques has increased substantially during last years (King & Sigrimis, 2000); (Sigrimis, 2001); (Farks, 2005); (Straten, 2007). As is well-known, greenhouses occupy very extensive surfaces where climate conditions can vary at different points (spatial distributed nature). Despite of that feature, it is very common to install only one sensor for each climatic variable in a fixed point of the greenhouse as representative of the main dynamics of the system. One of the reasons is that typical greenhouse installations require a large amount of wire to distribute sensors and actuators. Therefore, the system becomes complex and expensive and the addition of new sensors or actuators at different points in the greenhouses is thus quite limited. In the last years, WSN are becoming a convenient solution to this problem (Gonda & Cugnasca, 2006); (Narasimhan et al., 2007). A WSN is a collection of sensors and actuators nodes linked by a wireless medium to perform distributed sensing and acting tasks (Zhu et. al., 2006). The sensor nodes collect data and communicate over a network environment with a computer system, which is called base station. Based on the information collected, the base station takes decisions and then the actuator nodes perform the appropriate actions over the environment. This process allows users to sense and control the environment from anywhere (Gonda & Cugnasca, 2006). There are many situations in which the application of the WSN is preferred, for instance, environment monitoring, product quality monitoring,

and others where supervision of big areas is necessary (Feng et al., 2007). In this work, WSN are used in combination with event-based systems to control the inside greenhouse climate.
Control problems in greenhouses are mainly focused on fertirrigation and climate systems. The fertirrigation control problem is usually solved providing the amount of water and fertilizers required by the crop. The climate control problem consists of keeping the greenhouse temperature and humidity in specific ranges despite of disturbances. Adaptive and feedforward controllers are commonly used for climate control problems. Therefore, fertirrigation and climate systems can be represented as event-based control problems where control actions will be calculated and performed when required by the system, for instance, when water is required by the crop or when ventilation must be closed due to changes in outside weather conditions. Furthermore, such as discussed above, with event-based control systems a new control signal is only generated when a change is detected in the system. That is, the control signal commutations are produced only when events occur. This fact is very important for the actuator life and from an economical point of view (reducing the use of electricity or fuel), especially in greenhouses where commonly actuators are composed by mechanical devices controlled by relays.
Therefore, this work presents the combination of WSN and event-based control systems to be applied in greenhouses. The main focus of this chapter is therefore the presentation of a complex real application using a WSN, as an emerging technology, and an event-based control, as a new paradigm in process control. The following issues have been addressed:

- the issues posed to a multivariable, interacting control system by possibly faulty communications (as in a wireless context),
- the location of sensors to correctly represent, for the purpose of control, spatially distributed quantities,
- the efficient use of actuators, the term "efficient" referring also to correct use and wear minimization,
- the effects of event-based sampling.

As a first approximation, event-based control has been applied for temperature and humidity control issues. The main advantage of the proposed control problem in comparison with previous works is that promising performance results are reached reducing the use of wire and the changes of the control signals, which are translated into reductions of costs and a longer actuator life. The ideas presented in this chapter could be easily extrapolated, for instance, to building automation.

2. The climatic control problem in greenhouses

2.1 Description of the climatic control problem

Crop growth is mainly influenced by the surrounding environmental climatic variables and by the amount of water and fertilizers supplied by irrigation. This is the main reason why a greenhouse is ideal for cultivation, since it constitutes a closed environment in which climatic and fertirrigation variables can be controlled to allow an optimal growth and development of the crop. The climate and the fertirrigation are two independent systems with different control problems. Empirically, the requirements of water and nutrients of different crop species are known and, in fact, the first automated systems were focused to control these variables. As the problem of greenhouse crop production is a complex issue,

an extended simplification consists of supposing that plants receive the amount of water and fertilizers that they require at every moment. In this way, the problem is reduced to the control of crop growth as a function of climate environmental conditions (Rodríguez, 2002); (Rodríguez, 2008).

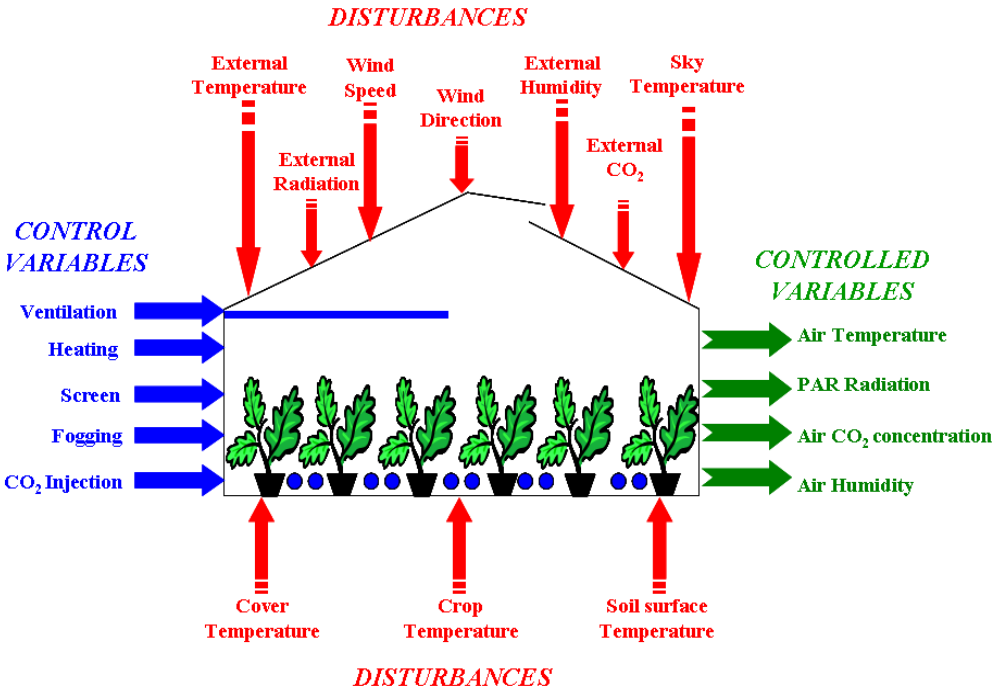


Fig. 1. Climatic control variables

The dynamic behaviour of the greenhouse microclimate is a combination of physical processes involving energy transfer (radiation and heat) and mass balance (water vapour fluxes and CO_2 concentration). These processes depend on the external environmental conditions, structure of the greenhouse, type and state of the crop, and on the effect of the control actuators (Bot, 1983). The main ways of controlling the greenhouse climate are by using ventilation and heating to modify inside temperature and humidity conditions, shading and artificial light to change internal radiation, CO_2 injection to influence photosynthesis, and fogging/misting for humidity enrichment. A deeper study about the features of the climatic control problem can be found in (Rodríguez, 2002).

The approach presented in this work is applied to the climatic conditions of the mild winter in Southern Europe (the data used for the simulations performed in this work have been collected in a greenhouse located at Southeastern Spain), where the production in greenhouses is made without CO_2 enrichment and the demand of quality products is increasing every day. Considering the greenhouse structures, the commonest actuators, the crop types, and the commercial conditions of this geographical area, the main climate variables to control are the temperature and the humidity. The PAR (Photosynthetically

Active Radiation with a spectral range from 400 to 700 Wm2) is used by the plants as energy source in the photosynthesis process.) and it is controlled with shade screens but its use is not much extended. So, this work is focused on the temperature control problems.

2.1 Air temperature control

Plants grow under the influence of the PAR radiation (diurnal conditions) performing the photosynthesis process. Furthermore, temperature influences the speed of sugar production by photosynthesis, and thus radiation and temperature have to be in balance in the way that a higher radiation level corresponds to a higher temperature. Hence, under diurnal conditions, it is necessary to maintain the temperature in a high level, being optimal for the photosynthesis process. In nocturnal conditions, plants are not active (the crop does not grow); therefore it is not necessary to maintain such a high temperature. For this reason, two temperature set-points are usually considered: diurnal and nocturnal (Kamp & Timmerman, 1996).

Due to the favorable climate conditions of Southeastern Spain, during the daytime the energy required to reach the optimal temperature is provided by the sun. In fact, the usual diurnal temperature control problem is the refrigeration of the greenhouse (with temperatures higher than the diurnal setpoint) using natural ventilation to reach the optimal diurnal temperature. On the other side, the nocturnal temperature control problem is the heating of the greenhouse (with temperatures lower than the nocturnal set-point) using heating systems to reach the nocturnal optimal temperature. In Southeastern Spain, forced-air heaters are commonly used as heating systems. In this work, the diurnal and nocturnal temperature control is analyzed to test the proposed event-based control. Therefore, typical temperature control systems with ventilation and heating are described in the following section.

The natural ventilation determines the air exchange and air flow in the greenhouse as a consequence of the differences between outside and inside temperatures. The relationship between vents aperture and inside temperature is not linear (Rodríguez, 2001), but instead of using a nonlinear control schema, it was decided to implement a gain-scheduling control algorithm based on linear models for each operating point (see Figure 2). Most commercial solutions include this kind of gain scheduling controllers to cope with both fast and slow changing dynamics due to disturbances.

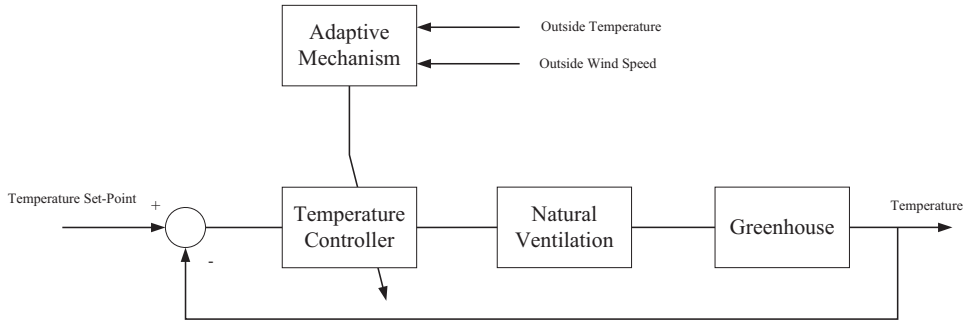
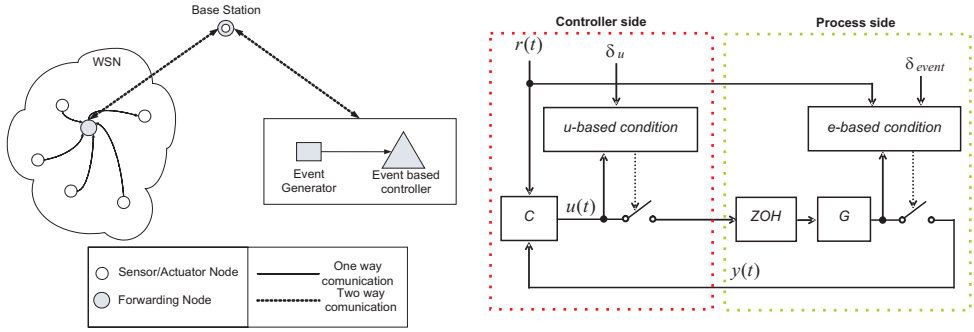


Fig. 2. Diurnal temperature controller

This controller consists of a gain-scheduling PI scheme where the controller parameters are changed based on some disturbances: outside temperature and wind speed. For the nocturnal temperature control, there exist many control strategies, but for this study an on/off control with dead zone is used in forced-air heaters, which is the controller commonly used in conventional greenhouses. A full description of these algorithms can be found in (Rodríguez, 2002).

2.2 Event-based control in greenhouses with WSN

As discussed above, in an event-based control system, the control actions are executed in an asynchronous way. The event-based sampling suggests that the most appropriate method of sampling consist of transmitting information only when a significant change in the signal occurs, justifying the acquisition of a new sample. In this work, the idea is to combine WSN with event-based control (see Figure 3) such as is proposed in (Pawlowski et al., 2008).



a) Field application b) Controller structure

Fig. 3. Event-based control with Wireless Sensor Network

In this scheme, the process (a greenhouse in this case) is provided with a WSN where each sensor transmits data according to a specific sampling approach. For instance, in (Pawlowski et al., 2008), this architecture is proposed and the level crossing method is used. Therefore, in that case, each sensor will transmit data if the absolute value of the difference between the current value of the variable, $v(t_k)$, and its value in the last transmission, $v(t_s)$, is greater than a specific limit δ. In a general way, an event-based controller consists of two parts (see Figure 3a): an event detector and a controller. The event detector indicates to the controller when a new control signal must be calculated due to the occurrence of an event. Figure 3b shows the event-based controller structure, where two type of events are generated based on "u " and "e" conditions. In our application, the actuator owns a ZOH (Zero-Order Hold), so the current control action is maintained until the arrival of a new one. Since the controller owns inputs and outputs, we have considered input- and output-side events. The input-side ones are the arrival of a new value of the controlled variable "y" (as consequence of the triggering of some sensor-side event) and the introduction of a new reference; both cases force the calculation of control actions. The *u-based* criterion of the output-side consists on just sending the new control action $u(t)$ if it is different enough of the previous control action $u(ts)$.

In next sections, the effect of different sampling techniques will be evaluated in the control system.

2.2 Control Performance

Different performance measurements have been used to compare the quality of the control system regarding to different event-based sampling techniques. These measurements are the following (Vasyuntynskyy & Miskowicz, 2007):

- IAE: The Integrated Absolute Error is defined as:

$$IAE = \int_0^\infty e(t)dt \tag{1}$$

- IAEP: It is the difference between the system response of an event-based strategy and the system response of the time-based approach:

$$IAEP = \int_0^\infty \left| y_{time-based}(t) - y_{event-based}(t) \right| dt \tag{2}$$

- NE: The Number of Events is a sampling efficiency measure to compare the quality of the system response:

$$NE = \frac{IAEP}{IAE} \tag{3}$$

- IAD: The integrated absolute difference is the difference between the IAE of the time-based strategy and the IAE of the event-based ones:

$$IAD = \int_0^\infty \left| IAE_{time-based}(t) - IAE_{event-based}(t) \right| dt \tag{4}$$

- GPI: Global Performance Index (Vasyuntynskyy & Miskowicz, 2007) shows the compromise between the control performance and the sampling efficiency in the following way:

$$GPI = W_1 \cdot Calls + W_2 \cdot Actions + W_3 \cdot Sendings + W_4 \cdot NE \tag{5}$$

where W_i are weighting factors.

A Global Performance Index is calculated taking into account the quality of the system response and the efficiency of the sampling. The influence of sampling techniques on the performance is represented by the following factors:

- *Calls*: It measures the number of communication messages sent from the sensor to the controller.
- *Actions*: Number of invocations of the controller.
- *Sendings*: Number of the control actions sent from the controller to the actuator in the event-based approaches.

3. Greenhouse climatic control problem

This section describes the different sampling techniques evaluated in the paper. According to the error based condition used in the sensor nodes, different event-based strategies are selected (Sánchez et al., 2009):

- LC - When the difference between the current value and the last acquired value is greater than δ

$$\left|x(t_k) - x(t_s)\right| > \delta \tag{6}$$

- ILC - When the value of the IAE from last acquired value is greater than δ

$$\int_{t_s}^{t_k} \left|x(t) - x(t_s)\right| dt > \delta \tag{7}$$

- LP - When the difference between a prediction of the signal value and its current value is greater than δ

$$\left|x(t_k) - \hat{x}(t_k)\right| > \delta \tag{8}$$

- ILP - The integral of the difference between the prediction and the current value is greater than δ

$$\int_{t_s}^{t_k} \left|x(t) - \hat{x}(t)\right| dt > \delta \tag{9}$$

- EN - The energy of the difference between the current value and value of last acquired value is greater than δ

$$\int_{t_s}^{t_k} \left[x(t) - x(t_s)\right]^2 dt > \delta \tag{10}$$

First and second conditions do not need a detailed explanation since both are simple well-known deadband sampling strategies. Further details on these methods can be found in (Vasyuntynskyy & Kabitzsch, 2006); (Miskowicz, 2006); (Suh, 2007). The LP method, originally described in (Suh, 2007), consists of starting the calculation of future values of the signal after an event takes place. To calculate future values, a first order predictor:

$$\hat{x}(t_s) = f(\hat{x}(t_{s-1}), \hat{x}(t_{s-2})) \tag{11}$$

is used to estimate the evolution of the signal from last time a sample was sent to the controller. When the difference between the current value and its prediction for the current time is greater than a limit δ, the condition becomes true and the current signal state is transmitted to the controller. The ILP is a new criterion based on the previous LP. In this case, the sample is taken and sent when the area between the signal and its prediction is greater than δ. The EN criterion (Miskowicz, 2005) sends a sample of the signal state when the energy of the signal from last sending exceeds a certain threshold. Depending on the error-based condition, an additional time-based expression must be included in the condition to force a sending when a time-out expires:

$$(event\_based\_condition \text{ IS } true \text{ OR } (h_{without} \geq h_{max}) \tag{12}$$

where $h_{without}$ represents the elapsed time from the last sending to the controller. The main reason to do that is the avoidance of the sticking, which happens when the signal derivative tends to zero (Vasyuntynskyy & Kabitzsch, 2007). So, the LC and LP criteria can reach situations where the sensor does not sent information to the controller in spite of having a high error. However, the sticking is avoided in criteria where integration is done (ILC, ILP, and EN) since the error-based condition can become true even though the error derivative is zero. Table 1 shows the individual limits for the commonest variables used for control purposes.

Variable	Limit (δ = 5%)	Limit (δ = 3%)
Inside Temperature	0.60	0.36
Outside Temperature	0.61	0.36
Humidity	4.9	2.9
Solar Radiation	34.30	20.58
Wind Speed	0.53	0.31
Wind Direction	17.84	10.70

Table 1. Limits for greenhouse variables

These limits of δ = 3% and δ = 5% were calculated based on the authors experience and after analyzing three years of data (Pawlowski et al., 2008).
The calculation of δ limit for each individual variable was performed studying its minimum and maximum values. The value of the change of each variable for δ = 3% and δ = 5% was determined calculating the 3% and 5% of the difference between the maximum and minimum values. Instead of choosing only one limit for each variable, these two different limits were evaluated to analyze their effects, such as presented in next section.

To compare results between event-based sampling techniques from a data transmission point of view, the following efficiency factors have been considered:

- *Samples*: The number of samples obtained and transmitted using event-based sampling.
- *Saving*: Percentage of saving that can be done in comparison with the timed-based sampling when data are transmitted every sampling time.
- $T_{average}$: Average time between two consecutive events, that is, between two consecutive sendings from the sensors.

4. Simulation results

The simulations presented in this section have been performed using the greenhouse climatic model developed by (Rodríguez, 2002) and the TrueTime MATLAB/Simulink toolbox. TrueTime is a tool developed for the Simulink environment and it is used to simulate real-time systems, networked control systems, communication models, and WSN (Anderson et al., 2005). The main feature of TrueTime is the possibility of co-simulation of the interaction between the real-world continuous dynamics and the computer architecture in the form of task execution and network communication. The TrueTime computer block (see Figure 4) executes user-defined tasks and interrupt handlers representing e.g. I/O tasks, control algorithms, or network drivers. The scheduling policy of the individual computer block is arbitrary and decided by the user. TrueTime allows simulation of context switching and task synchronization using events or monitors (Henriksson, 2003).
TrueTime simulation environment allows us to implement a code in C++ or Matlab programming language for every simulated node. Hence it is possible to reuse this written code for direct implementation in WSN motes. This solution decreases significantly the time necessary for the implementation of simulated ideas. One of the most relevant advantages of WSN nodes is the ability/capability to remotely reprogram selected motes.

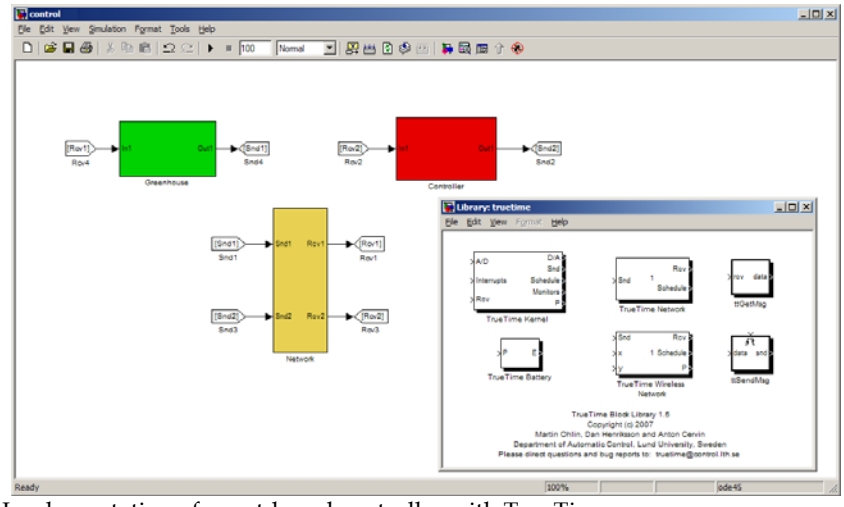


Fig. 4. Implementation of event-based controller with TrueTime

4.1 Data Transmission

Process monitoring is vitally important in companies for supervision tasks and the quality of the collected information has a great influence on the precision and accuracy of control results. Currently, the agro-alimentary market field incorporates different data acquisition techniques. Normally, the type of acquisition system is chosen to be optimal for the control algorithm to be used. In traditional climate monitoring and control systems, all sensors are distributed through the greenhouse and connected by wire to the device performing the control tasks.

These equipments use time-based data sampling techniques as a consequence of using time-based controllers. In modern control systems, it is common to use communication networks to transmit data between different control system blocks. Large amount of data are usually transmitted, and the data required by the controller in each sampling time are especially critical. The most reasonable solution from an economical point of view is to make use of existing network structure, and to share the network resources between different services, for instance, using Ethernet networks. Sometimes, this solution can produce a big network traffic burden (in a typical greenhouse control system, all data are transmitted every minute or even faster) and introduce time delays in the delivery of the data packets.

When the network load increases, the probability of data losses increases too, and this factor can be very negative for control performance. In some extreme examples, the control system needs dedicated network structure to minimize the time delay and the data losses.

On the other hand, the development of network structures in places with large distances, such as greenhouse installations, can become very expensive and with a complicated management. Wireless networks present an economic and useful solution to this problem, and more concretely, WSN for recording data and control purposes. However, most transceivers in WSN are battery-powered and the power consumption is a critical parameter.

Every transmission means power consumption and thus these systems present the problem of limitation in the amount of data to transmit. A solution to this problem is the use of the event-based sampling techniques described in previous sections. These techniques allow that only the necessary data will be transmitted and thus only the necessary power will be consumed. In this study, WSN based on IEEE 802.15.4 ZigBee protocol has been simulated and its combination with event-based sampling in the greenhouse climatic control problem. Results of simulation were evaluated for a full crop campaign of 120 days. In this work, only eight days have been selected to present the obtained results. The limits described in Table 1 were used for the event-based sampling.

Table 2 presents the results obtained after simulation of the selected eight days, where the comparison of data transmission is presented for the greenhouse variables. The table compares the number of samples obtained and transmitted using event-based sampling techniques with a time-based sampling. Figure 5 shows how the events are generated from changes in the outside temperature for the LC technique.

On the other hand, the variable dynamics highly affects the number of taken samples-events. This can be observed for variables with high-frequency changes such as the wind speed and direction. Figure 6 shows the transmission data for the wind direction. The transmission data from the sensors using level crossing sampling is shown on the top graphic, where a high transmission frequency is observed. However, in order to reduce the number of events created by this variable, the signal is filtered in the event generator before

detecting and sending events to the controller. The bottom graphics of Figure 6 shows how the number of samples is substantially reduced after filtering the signal. However, in order to cut down the number of events created by these variables, the signals should be filtered in the sensor node before sending events to the controller.

Variable	Index	TIME-BASED	LC		ILC		LP		ILP		EN	
			3%	5%	3%	5%	3%	5%	3%	5%	3%	5%
Inside Temperature	*Samples*	11808	762	359	2601	1930	1063	534	3212	2389	1042	837
	Saving	0	93,54	96,95	77,97	83,65	90,99	95,47	72,79	79,76	91,17	92,91
	T_average	1	15,5	32,89	4,54	6,12	11,11	22,11	3,68	4,94	11,33	14,11
Outside Temperature	*Sample*	11808	595	351	1715	1327	766	517	1846	1432	747	621
	Saving	0	94,96	97,02	85,47	88,76	93,51	95,62	84,36	87,87	93,6	94,74
	T_average	1	19,85	33,64	6,89	8,9	15,42	22,84	6,4	8,25	15,81	19,01
Humidity	*Sample*	11808	600	284	1794	1333	834	409	2193	1615	1246	1026
	Saving	0	94,91	97,59	84,80	88,71	92,93	96,53	81,42	86,32	89,44	91,31
	T_average	1	19,68	41,58	6,58	8,86	14,16	28,87	5,38	7,31	9,48	11,51
Solar Radiation	*Sample*	11808	826	553	1885	1464	1006	754	2244	1754	3170	2728
	Saving	0	93,00	95,31	84,03	87,60	91,48	93,61	80,99	85,14	73,15	76,89
	T_average	1	14,3	21,35	6,26	8,07	11,74	15,66	5,26	6,73	3,72	4,33
Wind Speed	*Sample*	11808	802	386	2569	1890	1024	557	3070	2318	1015	802
	Saving	0	93,20	96,73	78,24	83,99	91,32	95,28	74,00	80,36	91,40	93,20
	T_average	1	14,72	30,59	4,6	6,25	11,53	21,2	3,85	5,09	11,63	14,72
Wind Direction	*Sample*	11808	1707	993	3410	2489	2006	1162	3907	2989	5276	4607
	Saving	0	85,54	91,59	71,12	78,92	83,01	90,15	66,91	74,68	55,31	60,98
	T_average	1	6,92	11,89	3,46	4,74	5,89	10,16	3,02	3,95	2,24	2,56
	Average saving [%]	**0**	**92,53**	**95,87**	**80,27**	**85,27**	**90,54**	**94,44**	**76,75**	**82,36**	**82,36**	**85,00**

Table 2. Comparison of sampling techniques

As it can be seen, the number of events is smaller for δ = 5%. It can be observed a considerable saving in transmission is obtained for all event-based techniques for both limits, δ = 3% and δ = 5%. The average of transmission saving is over 80% for most of the variables. As an example, Figure 7 shows the original signal of the outside temperature and its sampled cases, where it can be observed how very good signal results are obtained for all event-based sampling techniques. Furthermore, it is observed that the amount of transmitted data decreases when the δ limit increases.

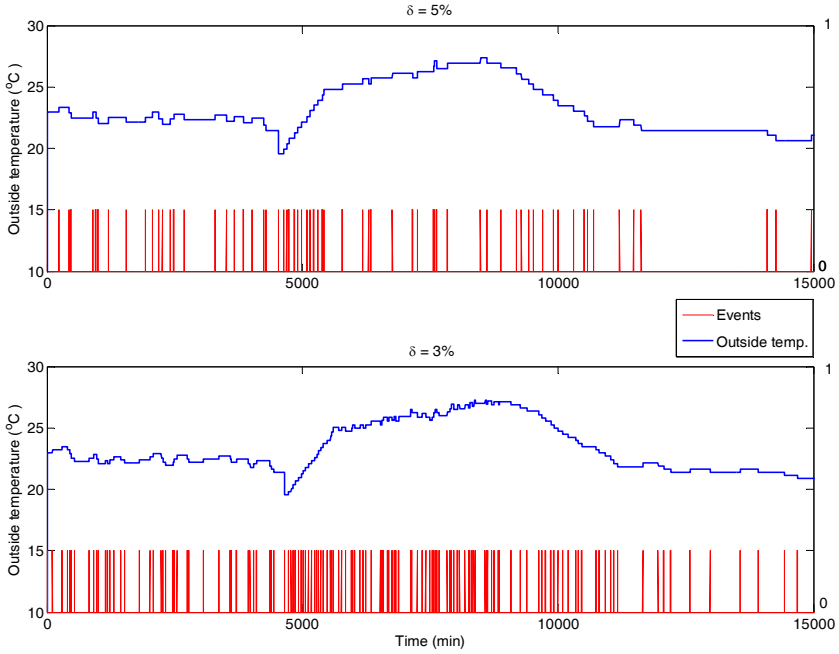


Fig. 5. Event generation for outside temperature

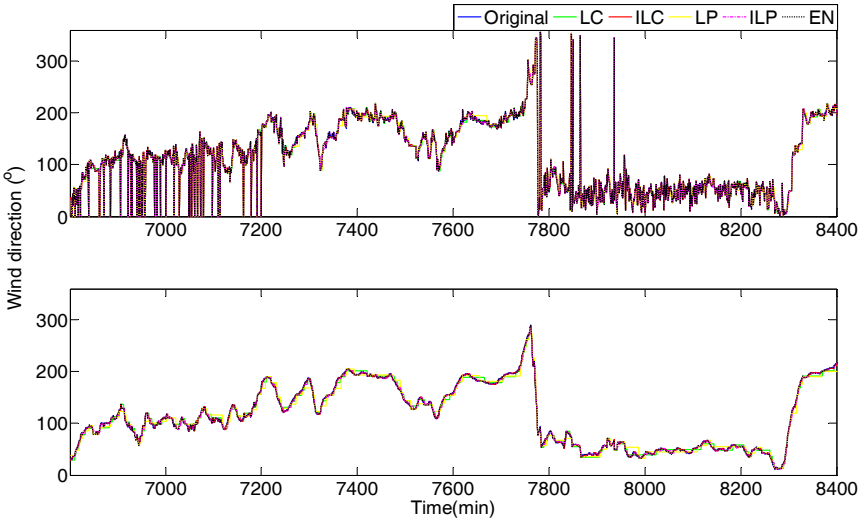


Fig. 6. Signal with high frequency dynamics

The biggest saving is obtained for the LC and LP techniques with $\delta = 5\%$ as consequence of the low sensibility to signal changes. The effects of the δ limit can be observed in Figure 8 where sampled signal of the solar radiation is shown. As it can be noticed, the transmission data is smaller for $\delta = 5\%$ but producing a bigger signal destruction (Figure 8b).
So, it is clear that the number of samples depends on two factors: the limit δ and the variable dynamics. The $T_{average}$ value was described in section 3 and is directly related to the transmission frequency. Lower values mean a high number of samples. All techniques with integrator part present better signal reconstruction property for signals in steady state or with low-frequency changes. For this reason, in these cases, it is not necessary to use the condition from equation (12). In conclusion, and from a control design point of view, by choosing $\delta = 3\%$ it is possible to obtain a high reduction of the acquired samples for all event-based sampling techniques without relevant information loss.

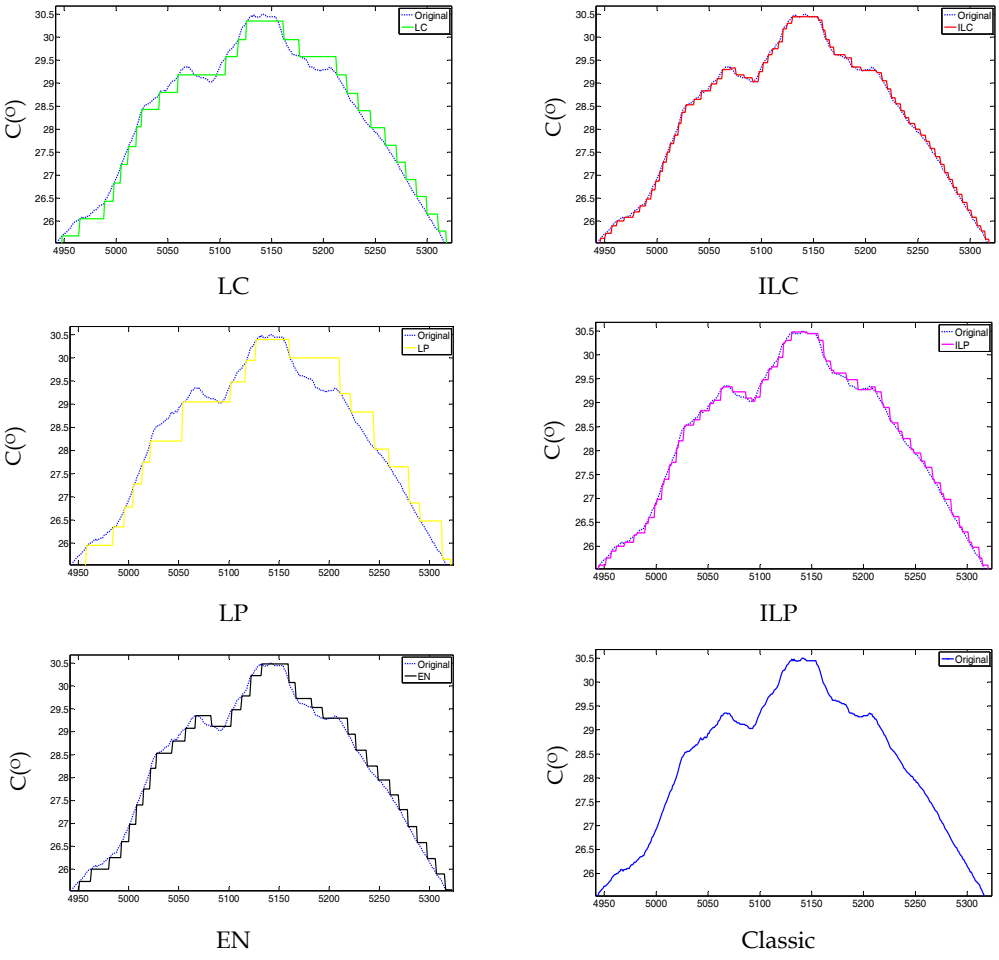
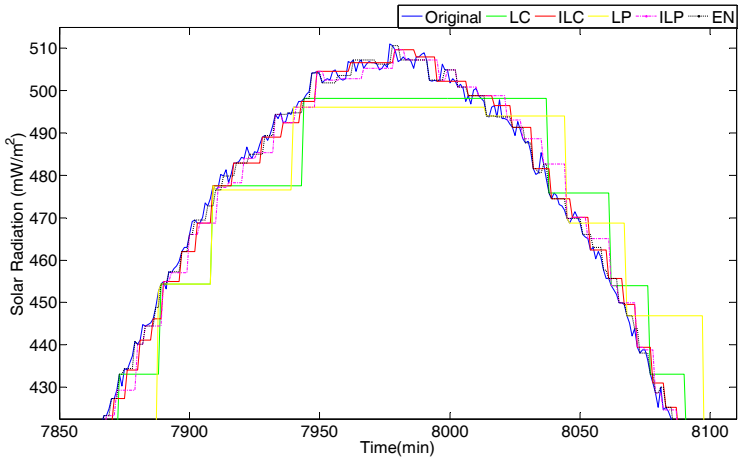
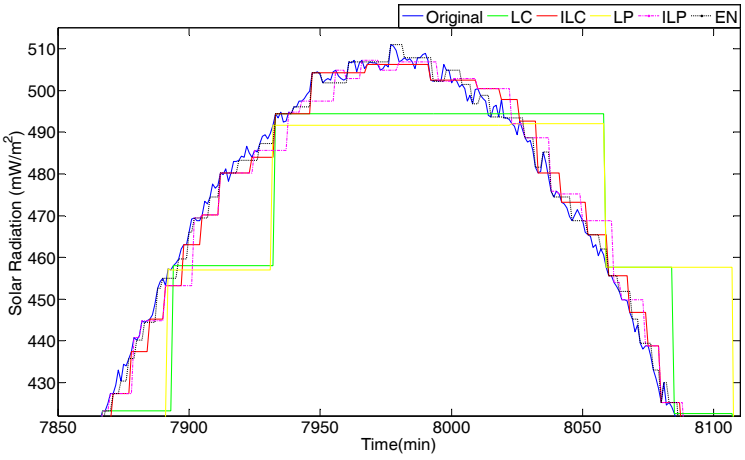


Fig. 7. Signal tracking for event-based sampling techniques



a) $\delta = 3\%$



b) $\delta = 5\%$

Fig. 8. Influence of δ limit on the signal sampling.

4.1 Comparison of control performance

This section presents the simulation results obtained for the greenhouse climatic control problem. The control system works as described in Figure 3, where the controller calculates a new control signal when an event happens. The LC and LP techniques incorporate the additional condition (12) to guarantee event generation in steady state situations. The event triggering is governed by an event generator that detects the possible events affecting the controller. For this simulation study, these events are represented by changes on: set-point, inside temperature, outside temperature, and wind speed.

The events are generated when the controller node receives a data packet, and produces a new action from the PI control task. If the new value of the control signal is different from

the value sent last time, a new transmission to actuator node is performed. As discussed above, only eight selected days have been used as representative of the simulation study. The temperature set-point (SP) was set at 26 °C and 17 °C for diurnal and nocturnal periods, respectively. Figure 9 and 10 presents the simulation results for a two-day diurnal period with the purpose of showing up the influence of event-based controllers. These Figures compare a time-based controller (TB) and an event-based controller (EB) for each different sampling method and with $\delta = 3\%$. For these specific days, event-based controllers (EB-ILC, EB-ILP, and EB- EN) present better performance that the time-based one and, at the same time, produce lesser commutations in the control signal (see Figures 9b and 10b). This effect is verified by the IAE presented in Table 3, which collects all control results for the diurnal period. Figure 11 shows all control performance indexes for diurnal period. Furthermore, the NE index confirms good results for the aforementioned techniques, especially for $\delta = 3\%$. The GPI weighting factors were set to 1 during calculation of this index, where large values of this index mean worst overall performance.

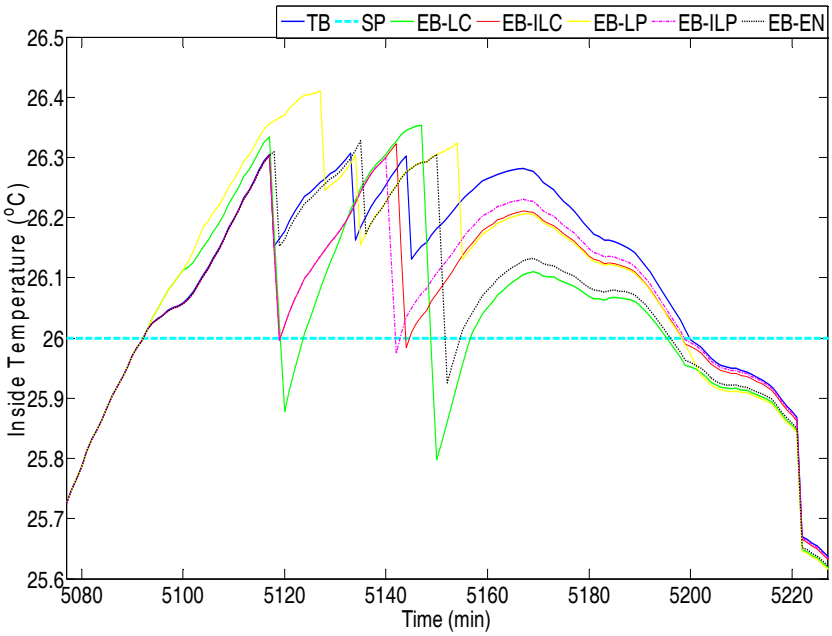
Index	Time-based	LC		ILC		LP		ILP		EN	
		3%	5%	3%	5%	3%	5%	3%	5%	3%	5%
IAE	134.91	135.88	152.96	124.03	136.31	149.72	162.87	124.46	126.64	134.18	135.45
IAEP	0	65.18	87.69	37.59	31.97	64.43	84.99	31.82	35.06	31.28	67.51
NE	0	0.479	0.561	0.303	0.234	0.430	0.521	0.255	0.276	0.233	0.498
IAD	0	0.965	21.40	10.87	1.40	14.81	27.95	10.45	8.26	0.729	0.543
GPI	24155	4379,4	2169,5	13860,3	10368,2	5775,4	3265,5	16351,2	12360,2	5662,2	4598,4
Calls	11808	2159	1060	6885	5145	2853	1608	8128	6139	2786	2260
Sending	11808	2159	1060	6885	5145	2853	1608	8128	6139	2786	2260
Actions	539	61	49	90	78	69	49	95	82	90	78

Table 3. Control performance indexes for the diurnal period

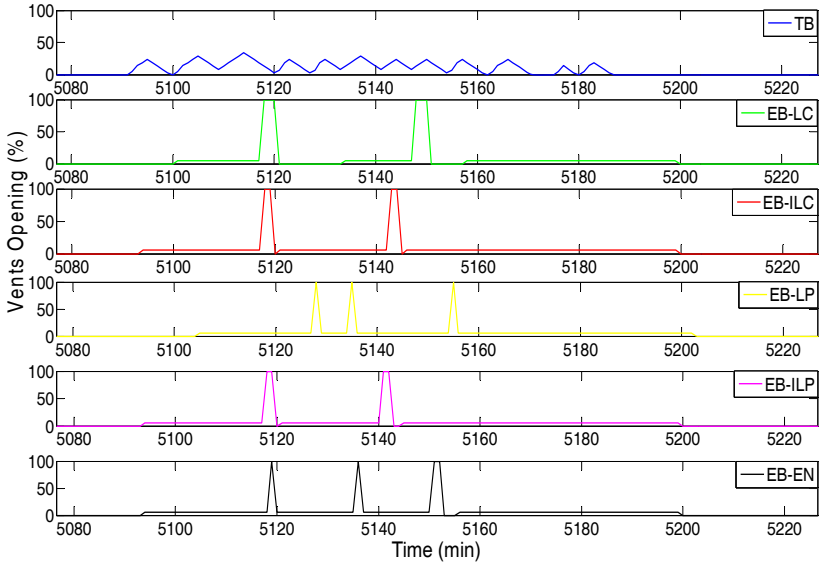
Index	Time-based	LC		ILC		LP		ILP		EN	
		3%	5%	3%	5%	3%	5%	3%	5%	3%	5%
IAE	239.72	881.19	1472.9	453.04	568.8	933.69	1421.7	388.72	493.1943	830.97	943.4
IAEP	0	840.54	1401.6	425.29	535.3	892.32	1365.0	385.52	477.5	786.79	899.5
NE	0	0.953	0.951	0.938	0.941	0.955	0.960	0.991	0.968	0.946	0.953
IAD	0	641.47	1233.2	213.32	329.1	693.97	1182.0	149.00	253.4	591.24	703.7
GPI	27045	5140,9	2622,9	15550,9	11672,9	6568,9	3772,9	18323,9	13942,9	7352,9	5902,9
Calls	11808	2159	1060	6885	5145	2853	1608	8128	6139	2786	2260
Sending	11808	2159	1060	6885	5145	2853	1608	8128	6139	2786	2260
Actions	3429	822	502	1780	1382	862	556	2067	1664	1780	1382

Table 4. Control performance indexes for the nocturnal period

The results of GPI were better for event-based controllers with $\delta = 5\%$, and it depends on the number of samples that produce each sampling technique. The EB-EN case presents a good compromise between numbers of samples and control performance for both δ limits. Figures 12 and 13 show results for the nocturnal period. The results show a worst, but acceptable, performance of the event-based controllers. The important advantage of the event-based controllers is the reduction of changes in the control signal, which it is very relevant for the actuator life and the fuel/electricity consumption. In this example, saving over 50% is obtained comparing with the time-based strategy (see Figures 12b and 13b).

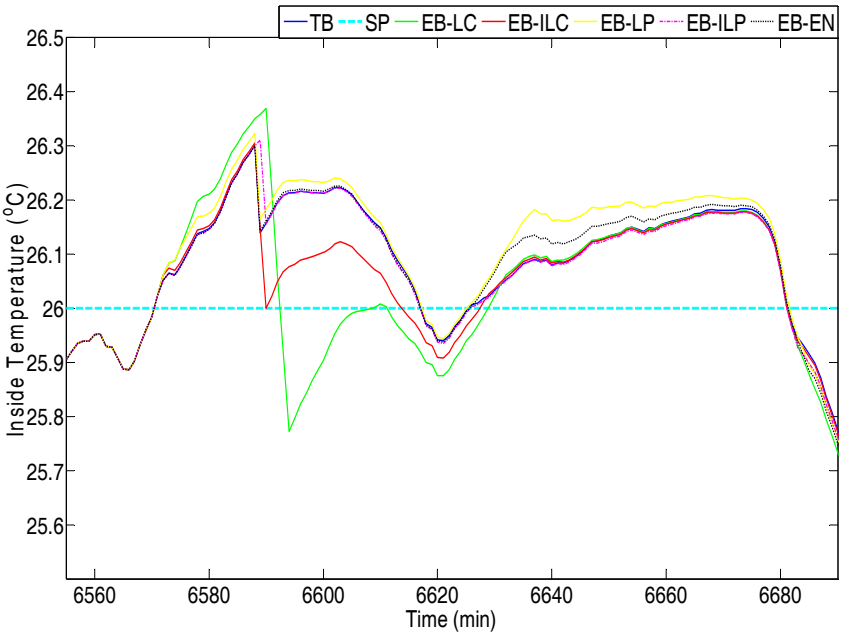


a) Control results

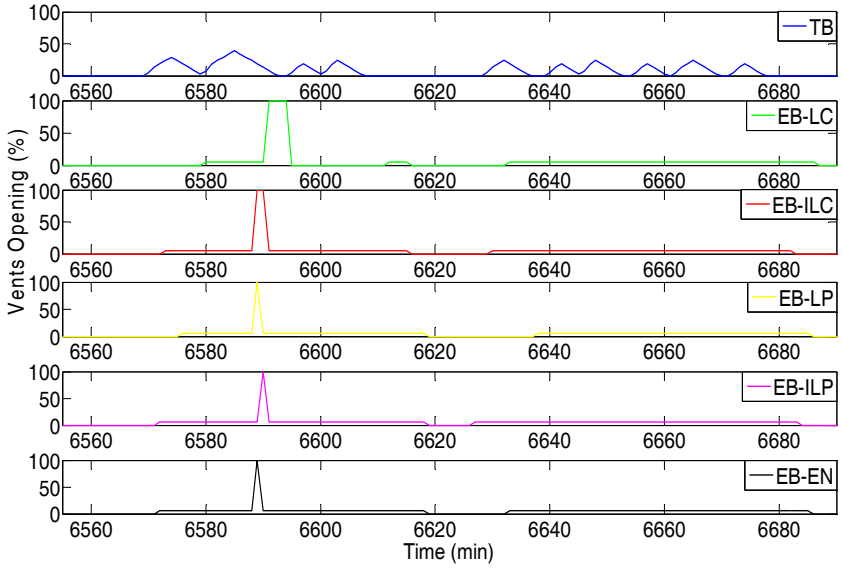


b) Control signal

Fig. 9. Control results for a two-hour diurnal period – example 1

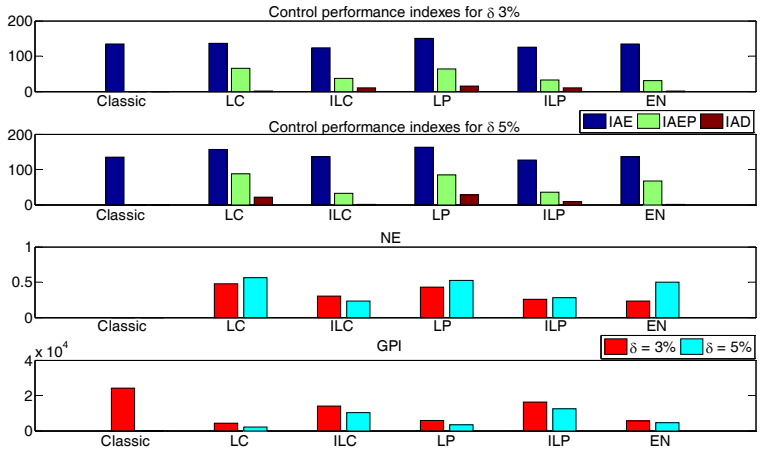


a) Control results

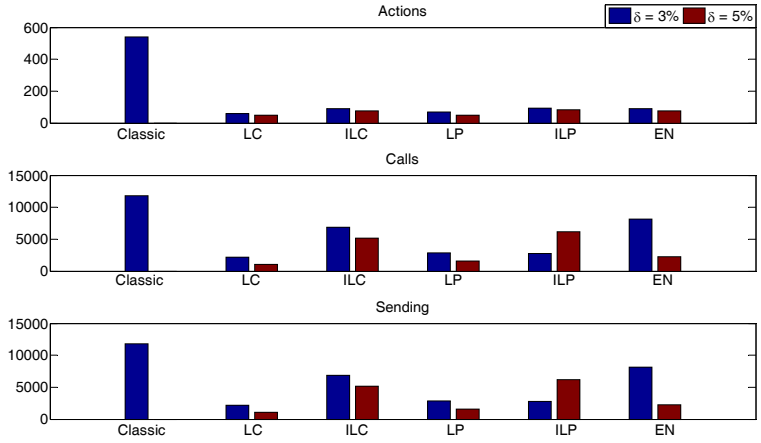


b) Control signal

Fig. 10. Control results for a two-hour diurnal period– example 2



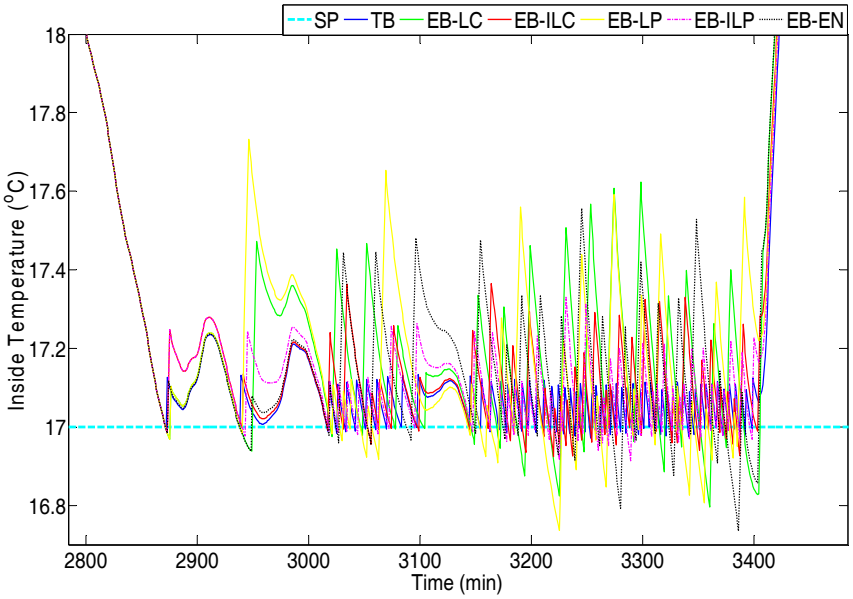
a) Control performance



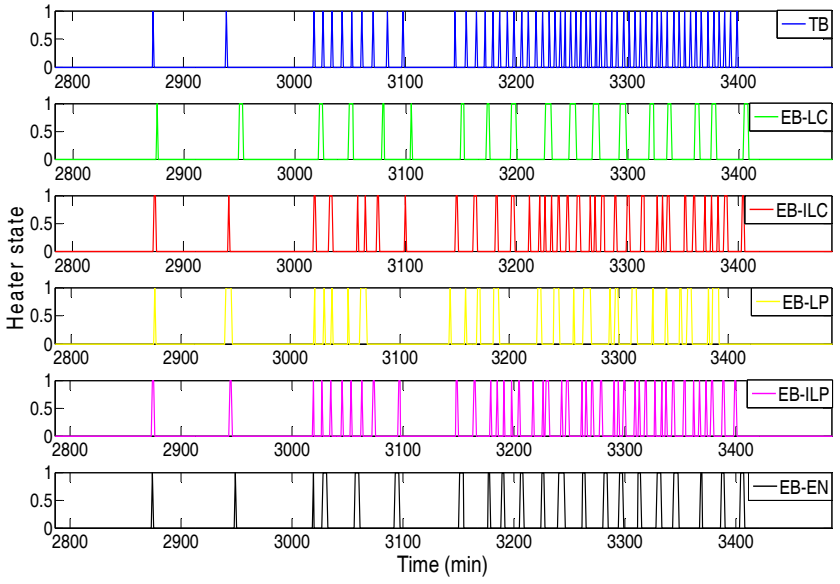
b) Communication performance

Fig. 11. Control performance indexes for diurnal period

The lowest number of commutations is produced by the EB-LC and EB-LP cases, but their errors are bigger in comparison with the other event-based controllers. Table 4 accumulates results of control performance for nocturnal period. Figure 14 shows the full list of control performance indexes for the nocturnal period. The IAE values confirm that the control quality is worst for event-based controllers and only the EB-ILP and EB-ILC obtain magnitudes approximated to the TB one.

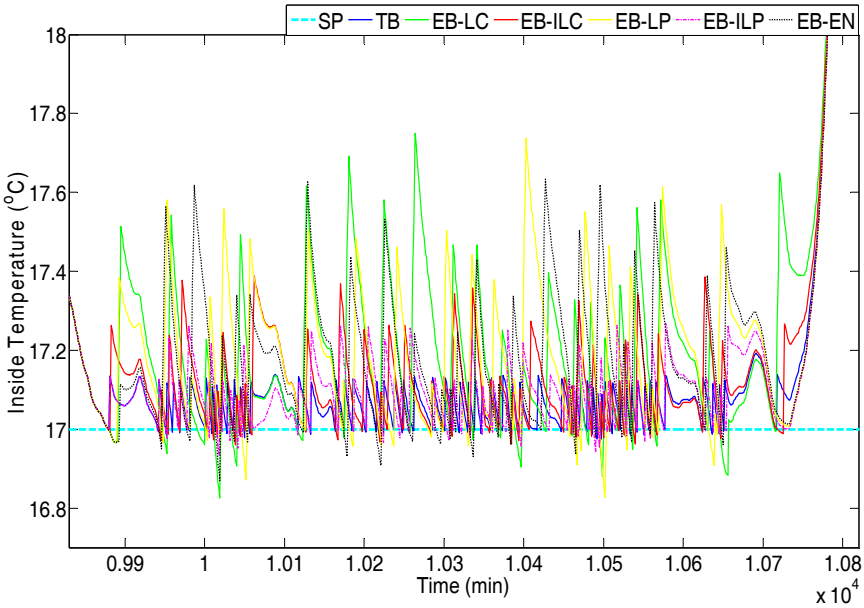


a) Control results

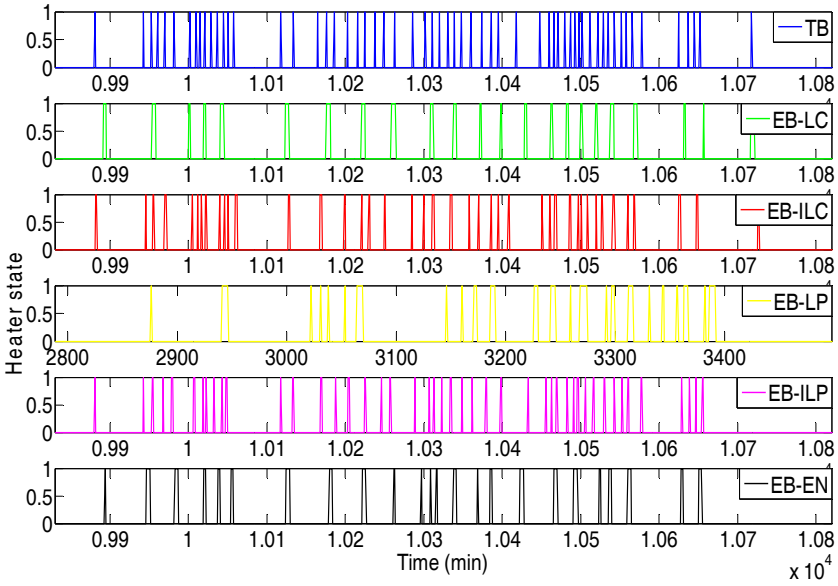


b) Control signal

Fig. 12. Control results for a twelve-hour nocturnal period – example 1

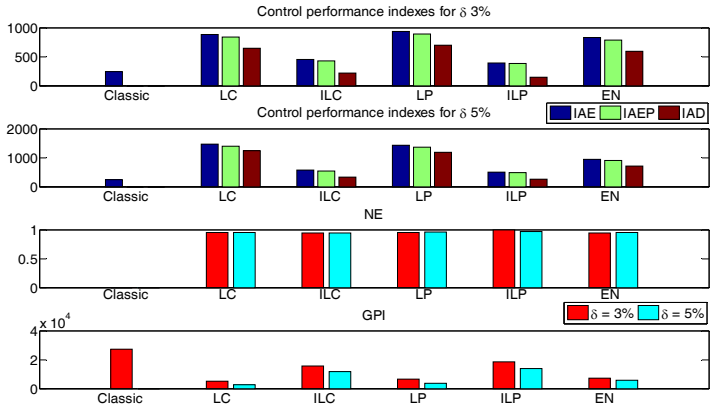


a) Control results

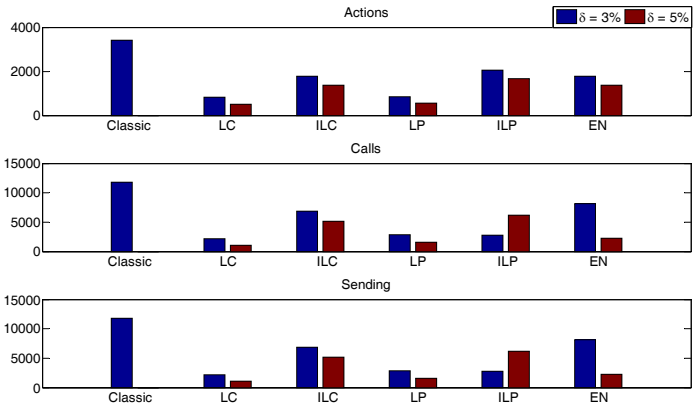


b) Control signal

Fig. 13. Control results for a twelve-hour nocturnal period – example 2



a) Control performance



b) Communication performance
Fig. 14. Control performance indexes for the nocturnal period

The consequence of this fact is the high number of transmissions between the different blocks in the control system. The analysis of the GPI index shows that EB-LC, EB-LP, and EB-EN present similar results. However, the EB-EN controller keeps reduced the transmissions without a relevant increase of the IAE in comparison to the other event-based controllers.

5. Conclusion

This paper presents a study of event-based sampling techniques and their application to the greenhouse climate control problem. It was possible to obtain important information about data transmission and control performance for all techniques. As conclusion, it was deduced

that the data rate can be set up to obtain a compromise between control performance and number of transmissions, where results for different values of δ limit where shown. The δ limit of the event-based sampling techniques has presented a great influence on the event-based control performance, where for a greenhouse climate control problem, a value of 3% has provided promising results. On the other hand, the event-based controllers reduce the number of commutations to about 80% in comparison to the traditional time-based controller. This result is very important for greenhouses since it allows reducing the electricity/fuel costs and extending the actuator life.

6. Acknowledgments

This work has been supported by the Spanish CICYT under DPI2007-66718-C04-04 and DPI2007-61068 grants.

7. References

Anderson, M.; Henriksson, D.; Cervin, A.; Arzen, K.(2005). Simulation of wireless networked control systems. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Sevilla, Spain. 476- 481.

Åström, K. J.; Wittenmark, B. (1997). *Computer controlled systems: Theory and design*. Prentice Hall, ISBN-10: 0133148998.

Åström, K.J. (2007). *Analysis and Design of Nonlinear Control Systems*. In Event based control; Astolfi,A., Marconi, L. Eds.; Springer: Berlin, Heidelberg.

Årzén, K.J. (1999). A simple event-based PID controller. *In Proceedings of 14th IFAC World Congress*,Beijing, China.

Banatre, M.; Marrón, P. J.; Ollero, A.; Wolisz, A. (2008). *Cooperating Embedded Systems and Wireless Sensor Network*, Wiley. ISBN: 9781848210004

Bot, G. P. A.(1983). *Greenhouse climate from physical processes to a dynamic model*. PhD thesis, Agricultural University of Wageningen, The Netherlands.

Dormido, S.; Sánchez, J.; Kofman, E. (2008). Sampling, event-based control and communication (in Spanish). *Revista Iberoamericana de Automática e Informática Industrial* , 5, 5–26.

Feng, X.; Yu-Chu, T. ; Yanjun, L.; Youxian S. (2007). Wireless sensor/actor network design for mobile control applications. *Sensors*, (7). 2157-2173.

Farkas, I. (2005) Modelling and control in agricultural processes. *Computers and Electronics in Agriculture*., 49, 315–316.

Gonda, L.; Cugnasca, C. E. (2006). A proposal of greenhouse control using wireless sensor networks. *In 4thWorld Congress Conference on Computers in Agriculture and Natural Resources*. Orlando, Florida, USA.

Henningsson, T.; Johannesson, E.; Cervin, A. (2008). Sporadic event-based control of first-orden linear stochastic systems. *Automatica*, 44. 2890-2895.

Henriksson, D; Cervin, A.; Årzén, K. E. (2003). Truetime: Real-time control system simulation with matlab/simulink. *In Proceedings of the Nordic MATLAB Conference*. Copenhagen, Denmark.

Kamp, P. G. H.; Timmerman, G. J. (1996). *Computerized environmental control in greenhouses. A step by step approach*. IPC Plant, The Netherlands.

King, R.; Sigrimis, N. (2000). Computational intelligence in crop production. *Computers and Electronics in Agriculture.* Special Issue on Intelligent Systems in Crop Production, 31(1).

Miskowicz, M. (2005). Sampling of signals in energy domain. *In 10th IEEE Conference on Emerging Technologies and Factory Automation*. 263-266.

Miskowicz, M. (2006). Send-on-delta concept: An event-based data reporting strategy. *Sensors*, 1(6), 29-63.

Miskowicz, M. (2007). Asymptotic effectiveness of the event-based sampling according to the integral criterion. *Sensors*. (7), 16-37.

Narasimhan, L. V.; Arvind, A.; Bever K. (2007). Greenhouse asset management using wireless sensor-actor networks. *In IEEE International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. French Polynesia, Tahiti.

Pawlowski, A.; Guzmán, J.; Rodríguez, F.; Berenguel, M.; Sánchez, J.; Dormido, S. (2008). Event-based control and wireless sensor network for greenhouse diurnal temperature control: a simulated case study. *In 13th International Conference on Emerging Technologies and Factory Automation.* Hamburg, Germany.

Rodríguez, F.; Berenguel, M.; Arahal, M. R. (2001). Feedforward controllers for greenhouse climate control based on physical models. *Proceedings of the European Control Conference ECC01*. Oporto, Portugal.

Rodríguez, F. (2002). Modeling and hierarchical control of greenhouse crop production (in Spanish). PhD thesis, University of Almería, Spain, 2002.

Rodríguez, F.; Guzmán, J. L.; Berenguel, M.; Arahal, M. R. (2008). Adaptive hierarchical control of greenhouse crop production. *International Journal of Adaptive Control Signal Processing*, 22.180–197.

Sánchez, J.; Guarnes, M.; Dormido, S.; Visioli, A. (2009). Comparative study of event-based control strategies: An experimental approach on a simple tank. *In European Control Conference*. Budapest, Hungary.

Sandee, J. H.; Heemels, W. P. M. H.; Bosch, P. P. J. (2005). Event-driven control as an opportunity in the multidisciplinary development of embedded controllers. *In Proceedings of the American Control Conference*. Portland, Oregon, USA.

Sigrimis, N.; Antsaklis, P.; Groumpos, P. (2001). Control advances in agriculture and the environment. *IEEE Control System Magazine.* Special Issue.

Straten, G. (2007). What can systems and control theory do for agriculture? *In IFAC 2nd International Conference AGRICONTROL 2007.* Osijek, Croatia.

Suh, Y. (2007). Send-on-delta sensor data transmission with a linear predictor. *Sensors*, (7). 537-547.

Vasyuntynskyy, V.; Kabitzsch, K. (2006). Implementation of pid controller with send-on-delta sampling. In International Conference Control.

Vasyuntynskyy, V.; Kabitzsch, K. (2007). Simple PID control algorithm adapted to deadband sampling. *In 12th IEEE Conference on Emerging Technologies and Factory Automation*, 932-940.

Vasyuntynskyy, V.; Miskowicz, M. (2007). Towards comparison of deadband sampling types. *In Proceedings of the IEEE International Symposium on Industrial Electronics*, 2899-2904.

Zhu, Y.W.; Zhong, X. X.; Shi, J. F. (2006). The design of wireless sensor network system based on ZigBee technology for greenhouse. *Journal of Physics*, 48.1195-1199.

15

3D RFID Simulation and Design - Factory Automation

Wei Liu and Ming Mao Wong
** Singapore Institute of Manufacturing Technology*
Singapore
71 Nanyang Drive, Singapore 638075

1. Introduction

The chapter addresses the problem by presenting an analytical model for a 3-dimensional single folded loop antenna with detection coverage in space. Based on the antenna theory, the inductance and impedance of the loop antenna is investigated. Design issues including antenna topology, read range, tag orientations, proximity of metal and other antennas are addressed to determine proper antenna for optimal performance. The proposed design is verified by field distribution measurement and implemented as a RFID reader antenna for a smart shelf application.

1.1 RFID System

Radio Frequency Identification (RFID) is an automatic identification technology that transmits the identity of an object wirelessly using radio waves. It is evolving as a major technology enabler for items tracking and inventory management [1]. The great appeal of RFID technology is that it allows information to be stored and read without requiring either contact or a line of sight between the tag and the reader [2]. Because such technology conveniently dispenses with manual counting of items, it greatly reduces man-made errors and thus improves the accuracy of information.

The RFID system consists of readers (also known as interrogators), tags (also known as transponders), and an information managing host computer [4] shown in Figure 1. In a typical communication sequence, the reader emits a continuous radio frequency (RF) carrier sine wave. When a tag enters the RF field of the reader, the tag receives energy from the field. The tag is composed of an antenna coil and a silicon chip that includes basic modulation circuitry and nonvolatile memory. When the RF field passes through an antenna coil, there is an AC voltage generated across the coil. This voltage is rectified to result in DC voltage for the device operation. The tag becomes functional when the DC voltage reaches a certain level. After the tag has received sufficient energy, it modulates the carrier signal according to the data stored on the tag. Finally, the information is relayed to a host computer.

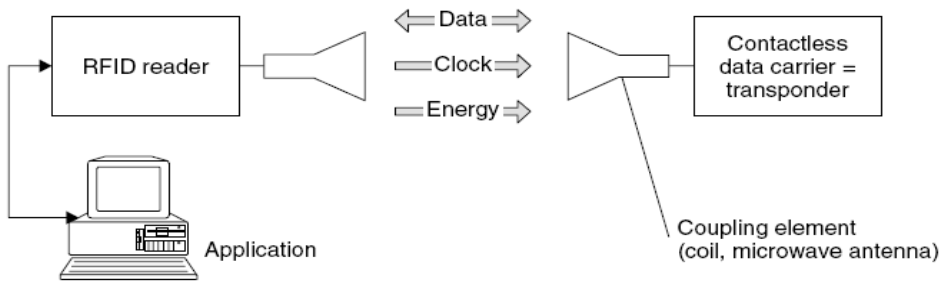


Fig. 1. RFID system

1.2 RFID Smart Shelf System

As part of an inventory management in the context of factory, the RFID smart shelf system is extremely useful for tracking of materials. By mounting a RFID reader antenna on each shelf and by placing a passive RFID tag with a unique serial number at each item, it would be possible to detect the presence of the individual item based on the unique serial number associated to it. Users can then query the database to determine the current location of the desired items.

Such a RFID smart shelf system will help to enable a fast, accurate and reliable inventory-checking and eliminates the need for manual stocktaking. Further, the information extracted from the item movements can be used to study usage patterns and detect missing items quickly.

The most common RFID smart shelf application is based on the high frequency (HF) band, which has an As one of RFID applications, the smart shelf system can be useful in maintaining the factory inventory in real time. The system is based on the inductive coupling at 13.56MHz. The read and write range at this frequency is usually not more than 1.5m, and most appropriate for the bookshelf.

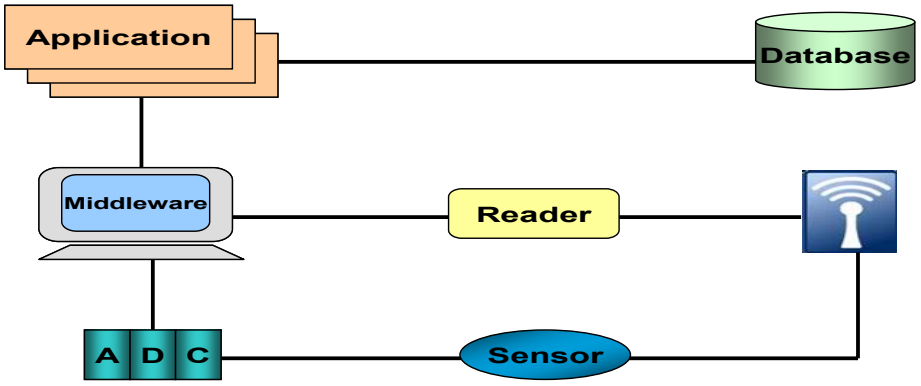


Fig. 2. RFID smart shelf system

Fig. 2. illustrates the components of a RFID smart shelf system. The middleware in the host manages the reader and issues commands. The reader and tag communicate via RF signal.

The tag receives and modifies the carrier signal generated through the reader antenna. The reader receives the modulated signal from the antenna and returns to the middleware. The application reads data from the middleware B , stores in the database and updates the user interface within the application (Figure 2).
Another feature is associated with the built-in sensor for detecting book movement and thus eliminating the need of constant manual monitoring on the application. The sensor is powered from the energy harvested from the radio wave of a reader. It is integrated with RFID system through Analog Digital Converter (ADC). Figure 7 shows the system implementation.
In order for the RFID system to operate effectively, the antenna plays a very crucial role. The design of the antenna determines the amount of coupling effect, which in turn determines the communication between the reader and the tag. If the antennas of the reader and the tag are not designed correctly, inductive coupling will not occur and the desired tag will not be activated and the whole RFID application to fail.
Among various RFID frequency bands for item level tracking, the most widely used frequency is 13.56 MHz in the high frequency (HF) band. Considering the fact that the wavelength is proportional to the antenna dimensions, designing electrically small antennas in this frequency with a wavelength of 22.12 meters is a very challenging task. To overcome this limitation, a practical and solid RFID reader antenna is proposed with appropriate dimensions as a practical and cheaper alternative [3]. A copper wire single loop antenna was chosen for its ductility, solidity and performance.
Most of the commercially available RFID smart shelf systems can only detect objects (containing RFID tag) effectively in the two-dimensional (2D) field. This requires the reader antennas and tag antennas to be in parallel planes. If they were in other orientations, the reader antenna may not be able to detect the presence of the tag antenna and this will greatly reduce the readability. This restricts the way tagged items can be placed on the shelf and requires multiple antennas to cover a three dimensional region.
The efficiency and performance of the antenna are greatly dependent on its design topology. The structure must be designed such that it can deliver maximum amount of energy to the desired destination in order to achieve efficient antenna performance.
The chapter is divided into four parts.
Part one gives a brief discussion on RFID technology and points out the design limitation for further improvement.
Part two describes the RFID operating principle and antenna design.
Part three details the design methodology of the reader antenna.
Part four concludes the paper and proposes future research.

2. Antenna Design

The RFID reader operates in the HF band at 13.56MHz and the reader antenna utilizes the magnetic field to transfer power to the battery-less tag during reading and writing operations. The associated electrical field is not used. The reader expects an antenna to be tuned to a center frequency of 13.56MHz and have an 50Ω impedance. For optimum performance, the reader matching should have a Voltage Standing Wave Ratio (VSWR) ratio of less than 1.2.

2.1 Design Guide

Of all the antennas that can be used to excite a magnetic field, loop antennas are recommended as the most suitable for generating the magnetic field that is required to transfer energy to the HF passive tag. But a number of issues have to be taken into consideration before the design of an antenna can begin.

2.1.1 Proximity of Metal

The presence of metal close to an antenna will reduce its performance to some extent . As the antenna is placed close to metal plate,, the metal will detune the parameters and also generate eddy current to cancel the EM wave generated by reader [14]. As a result, the read distance will drop. To overcome this, the antenna must be placed a certain amount of distance from the metal.

2.1.2 Quality Factor

The performance of an antenna is related with its Quality (Q) factor. In general, the higher the Q, the higher the power output for a particular sized antenna. However, the bandwidth is inversely proportional to the Q [15]. Hence, there should be a maximum value of Q. In the test bed, since the tag operates with a data rate of 70 kHz, the reader antenna circuit needs a bandwidth of at least twice of the data rate. Therefore, it needs:

$$B_{\min} = 140kHz \tag{1}$$

The Q can be can be determined by:

$$Q = \frac{f_0}{B} \tag{2}$$

Thus, the maximum attainable Q is obtained from formula (2),

$$Q_{\max} = \frac{13.56MHz}{140KHz} = 96.8 \tag{3}$$

2.1.3 Presence of Other Antennas

The presence of other antennas will alter the way a system performs because of coupling between the antennas. In this test bed where there are multiple shelves standing together, each embedded with one antenna, there would be interference among these closely placed antennas.

2.2 Topology Determination

The parameters of the loop antenna that can be chosen are shape, dimension, number of turns and wire diameter. These will be discussed in the following sections and will lead to a design methodology.

2.2.1 Shape

A novel 3-dimensional folded rectangular loop antenna is proposed to generate a magnetic field of at least certain strength within interrogation region. Figure 3 illustrates the schematic of designed antenna topology. The feed-point is located at the base where the electrical current enters and leaves the antenna. The two folded parts of the antenna helps to enhance the magnetic field to detect tags places in different orientations. After entering the feed-point, the current branches into two identical streams which flow through the rest part of the antenna and meet at the exit point. Since the currents flowing through two parallel folded parts are in the same direction, the mutual inductance is positive. Therefore, the magnetic field in x direction is strengthened.

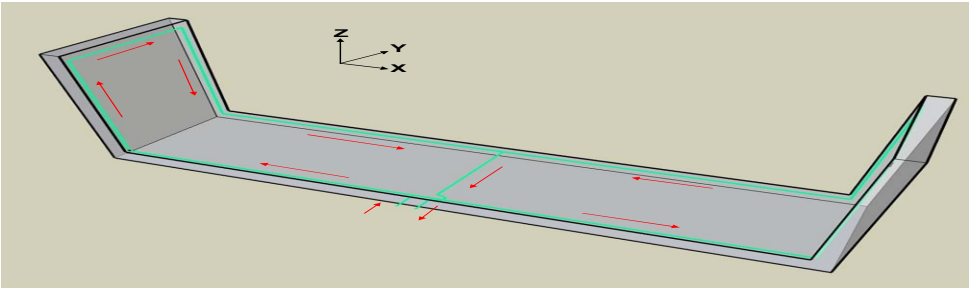


Fig. 2. Antenna topology

2.2.2 Number of Turns

For reading a loner ranger, one way is to achieve a larger B field . This can be done by increasing the number of ampere-turns (NI). However, there are drawbacks to this option as the more turns the antenna is, the less portable and more expensive the reader becomes. The consequence of increasing (NI) is a larger inductance in the reader antenna circuit that increases as the square of the number of turns. A high inductance load results in large amounts of reflected power (back EMF) as well as a large impedance that varies significantly. Therefore, it is important to keep (NI) as small as possible while achieving the minimum B field needed for the desired read range. For a loop antenna, fed by a voltage source, discarding the small ohm and radiation resistance with respect to ωL , the following formula is deduced:

$$\frac{V}{\omega L} = \frac{V}{\omega N L_{N=1}} \propto \frac{V}{N L_{N=1}} \tag{4}$$

where

N = number of turns

I = flowing current

$L_{N=1}$ = inductance of a loop of the same size with only one turn

Hence, to maximize the magnetic field strength B, we need to maximize $NL_{N=1}$, which means $N=1$ and keeping $L_{N=1}$ small.

2.2.3 Wire Diameter

Figure 4 illustrates the structure of loop antenna topology.

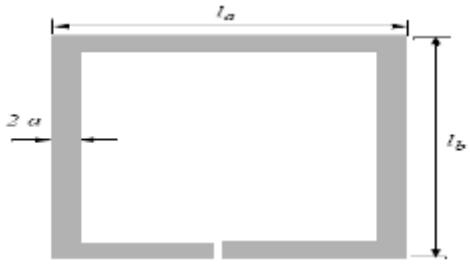


Fig. 4. Inductance of rectangular loop

If reader antenna is made of a rectangular loop composed of a thin wire, its inductance can be calculated by the following formula [16]

$$L_0 = 4\left\{ l_b \ln\left(\frac{2A}{a(l_b+l_c)}\right) + l_a \ln\left(\frac{2A}{a(l_a+l_c)}\right) + 2\left[a + l_c - (l_a+l_b)\right]\right\} \quad (nH) \quad (5)$$

Where the units are all in cm

a = radius of wire

$l_c = \sqrt{l_a^{\,2} + l_b^{\,2}}$

A = $l_a \times l_b$

Hence, using a wire with a large diameter helps to reduce L.

2.3 Impedance Matching

Since the reader antenna is designed for a smart shelf application, the dimension must fit the mechanical constraints of the bookshelf (100cm*30cm*40cm). The magnetic field created by the antenna is used to power the tags associated with the books within the bookshelf and the amount of energy induced in each tag is proportional to the strength of the magnetic field passing through the tag loop. The larger the reader antenna loop is, the more current carrying parts add a contribution to the magnetic field. If the loop becomes too large, however, these contributions become very weak due to the large distance from the current carrying part to the tag. Furthermore, if the total wire length of the loop becomes a considerable part of the wavelength of $22.12m$, the standing waves will cause multiple

resonances and decreases the total field [3, 11]. Hence, the antenna dimension is determined experimentally to provide the sufficient magnetic field.

2.3.1 Maximum Power Theorem

The antenna and the generator can be represented by an equivalent circuit [17] as shown in Figure 5. The impedance of the antenna is

$$Z_A = R_A + jX_A \tag{6}$$

where

Z_A = antenna impedance at terminals a-b (ohms)

R_A = antenna resistance at terminals a-b (ohms)

X_A = antenna reactance at terminals a-b (ohms)

In general the resistive part of (6) consists of two components; that is

$$R_A = R_r + R_L \tag{7}$$

where

R_r = radiation resistance of the antenna

R_L = loss resistance of the antenna

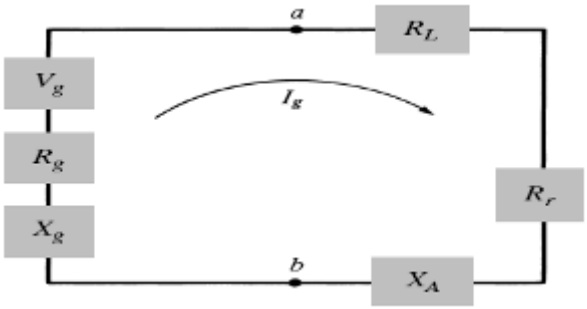


Fig. 5. Antenna equivalent circuit

To find the amount of power delivered to R_r for radiation and the amount dissipated in R_L as heat ($\frac{I^2 R_L}{2}$), we first need to find the current developed within the loop which is given by

$$I_g = \frac{V_g}{Z_t} = \frac{V_g}{Z_A + Z_g} = \frac{V_g}{(R_r + R_L + R_g) + j(X_A + X_g)} \quad (A) \tag{8}$$

and its magnitude by

$$\left|I_g\right| = \frac{\left|V_g\right|}{\left[(R_r + R_L + R_g)^2 + (X_A + X_g)^2\right]^{\frac{1}{2}}} \tag{9}$$

where V_g is the peak generator voltage. The power delivered to the antenna for radiation is given by

$$P_r = \frac{1}{2}\left|I_g\right|^2 R_r = \frac{\left|V_g\right|^2}{2}\left[\frac{R_r}{(R_r + R_L + R_g)^2 + (X_A + X_g)^2}\right] \quad (A) \tag{10}$$

and the dissipated as heat by

$$P_L = \frac{1}{2}\left|I_g\right|^2 R_L = \frac{\left|V_g\right|^2}{2}\left[\frac{R_L}{(R_r + R_L + R_g)^2 + (X_A + X_g)^2}\right] \quad (W) \tag{11}$$

The remaining power is dissipated as heat on the internal resistance R_g of the generator, and it is given by

$$P_g = \frac{\left|V_g\right|^2}{2}\left[\frac{R_g}{(R_r + R_L + R_g)^2 + (X_A + X_g)^2}\right] \quad (W) \tag{12}$$

The maximum power delivered to the antenna occurs when we have conjugate matching; that is when

$$R_r + R_L = R_g \tag{13}$$

$$X_A = -X_g \tag{14}$$

For this case

$$P_g = P_r + P_L = \frac{\left|V_g\right|^2}{8}\left[\frac{R_g}{(R_r + R_L)^2}\right] = \frac{\left|V_g\right|^2}{8}\left[\frac{R_r + R_L}{(R_r + R_L)^2}\right] \tag{15}$$

The power supplied by the generator during conjugate matching is

$$P_s = \frac{1}{2}V_g I_g^* = \frac{1}{2}V_g\left[\frac{V_g^*}{2(R_r + R_L)}\right] = \frac{\left|V_g\right|^2}{4}\left[\frac{1}{(R_r + R_L)}\right] \tag{16}$$

Of the power that is provided by the generator, half is dissipated as heat in the internal resistance (R_g) of the generator and the other half is delivered to the antenna. This only happens when conjugate matching is applied.

2.3.2 Two-Component Matching Network

Figure 6 displays the antenna impedance, which is $1.585 + j119.74(\Omega)$ at 13.56MHz. In order to achieve maximum power transfer, it is required to match the impedance of the load to that of the source (50Ω). This is accomplished by incorporating additional passive networks connected in between source and load.

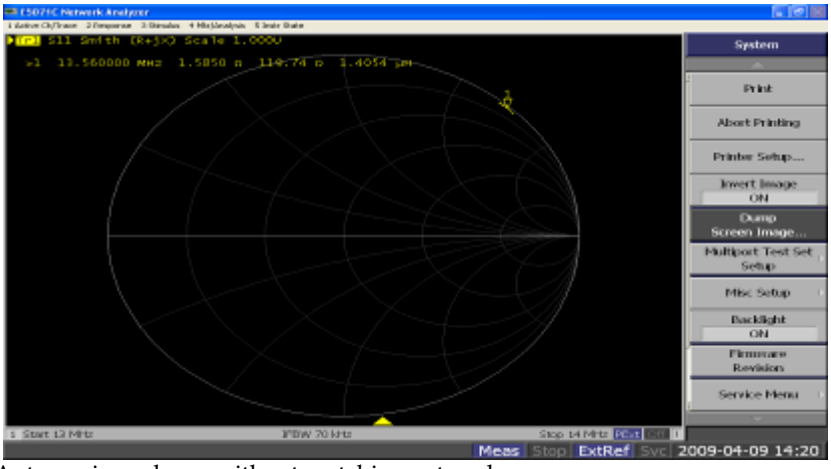


Fig. 6. Antenna impedance without matching network

The two-component network [18], also know as L-type due to the element arrangement, is used to transform the load impedance (Z_{load}) to the desired input impedance (Z_{in}). The components are alternatively connected in series and shunt configuration, as shown in Figure 7, which depicts eight possible arrangements of capacitors and inductors.

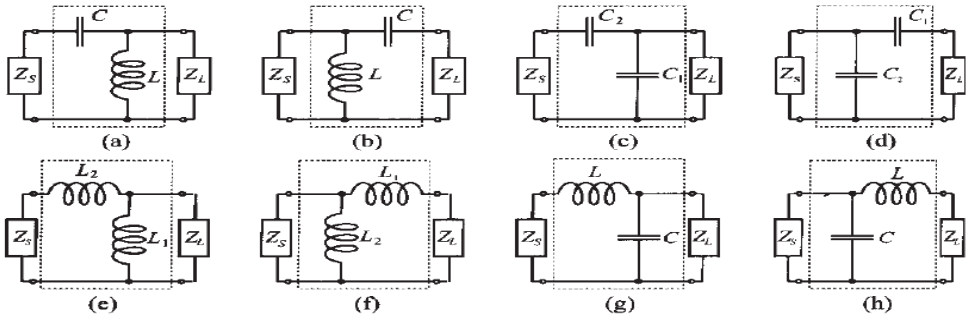


Fig. 7. Eight possible configurations of the two-component matching networks

There are two broad approaches in designing a matching network.

- To derive the values of the elements analytically
- To rely on the Smith Chart as a graphical design tool

The first approach yields very precise results. Alternatively, the second approach is more intuitive, easier to verify, and faster for an initial design, since it does not require complicated computations. Both approaches are applied as a cross-checking.

2.3.3 Analytical Approach

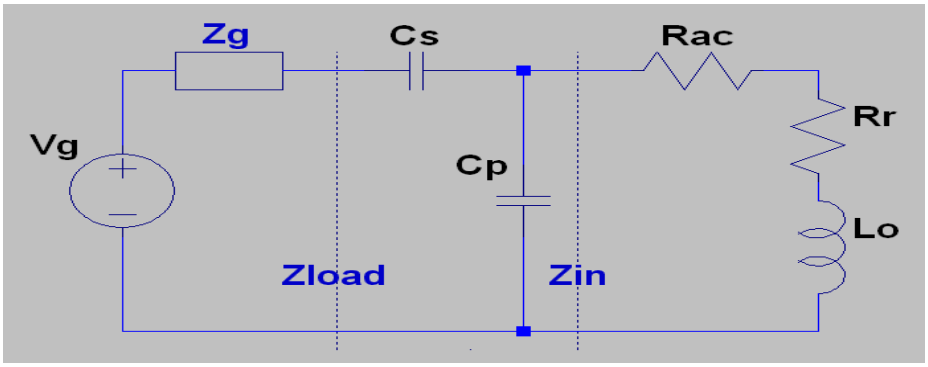


Fig. 8. L-type matching network design

As shown in Figure 8, the small loop antenna is primarily inductive and it can be represented by a lumped element equivalent circuit [17].

$$Z_{in} = R_{in} + jX_{in} = (R_r + R_{ac}) + jX_L \tag{17}$$

where

R_r = radiation resistance

R_{ac} = AC resistance of loop conductor

X_L = inductive reactance = ωL_0

The capacitors C_s, C_p are used to tune the antenna impedance to be 50Ω for maximum power efficiency. This is accomplished by choosing C_s, C_p according to

$$(-jX_s) + \frac{1}{\frac{j}{X_p} + \frac{1}{R_{in} + jX_{in}}} = 50 \quad (\Omega) \tag{18}$$

$$(-jX_s) + \frac{1}{\frac{jR_{in} - X_{in} + X_p}{X_p(R_{in} + jX_{in})}} = 50 \quad (\Omega) \tag{19}$$

$$(-jX_s)+\frac{X_p(R_{in}+jX_{in})(X_p-X_{in}-jR_{in})}{(X_p-X_{in})^2+R_{in}^2}=50 \quad (\Omega) \tag{20}$$

$$(-jX_s)+\frac{X_p^2R_{in}+j(X_{in}X_p^2-X_{in}^2X_p-X_pR_{in}^2)}{(X_p-X_{in})^2+R_{in}^2}=50 \quad (\Omega) \tag{21}$$

The real part is

$$\frac{X_p^2R_{in}}{(X_p-X_{in})^2+R_{in}^2}=50 \tag{22}$$

The imaginary part cancels out

$$\frac{X_{in}X_p^2-X_{in}^2X_p-X_pR_{in}^2}{(X_p-X_{in})^2+R_{in}^2}=X_s \tag{23}$$

Given $R_{in}=1.585, X_{in}=119.74$, it is calculated that $X_s=670.7, X_p=145.8$, which is $C_s=17.5pF, C_p=80.5pF$.

2.3.4 Graphical Approach

The Smith Chart is used for rapid and relatively precise designs of the matching circuits. The appeal of this approach is that its complexity remains almost the same independent of the number of components in the network. Moreover, the parameter choice and its value assignment can be instantaneously displayed at part of the Smith Chart on the computer screen.

Figure 9 illustrates the steps of L-type matching network design by using the Smith Chart as a graphical tool. The initial data point Z_L is $1.585+j119.74$. Since the first component is a shunt capacitor C_p, the total impedance of this parallel combination is positioned somewhere on the circle of constant conductance in green color that passes through the point Z_L. Next, the second capacitor C_s is added in series with the parallel combination of Z_L and C_p. The resulting impedance will move along the circle of constant resistance in red color. For maximum power gain it is required that an output impedance of the matching network connected to the antenna to be equal to the complex conjugate of the transmitter impedance. This circle has to pass through the center of the Smith Chart which is 50Ω.

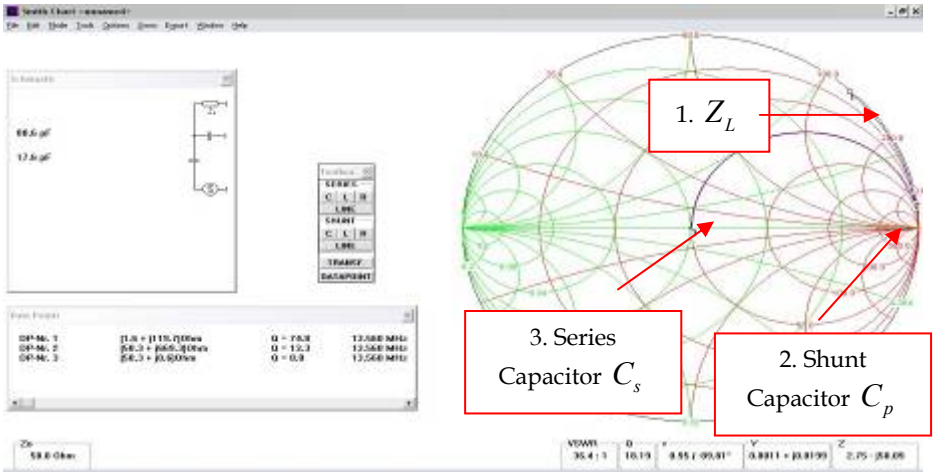


Fig. 9. Matching network design with Smith Chart

The intersection of two circles in the Smith Chart determines the impedance formed by the shunt connection of antenna and capacitor. Reading from the Smith Chart, it is found that this impedance is approximately

$$Z_{LC} = 49.4 + j666.86$$

The corresponding admittance is

$$Y_{LC} = 1.1 \times 10^{-4} + j1.5 \times 10^{-3}$$

The admittance of antenna is

$$Y_L = 1.1 \times 10^{-4} + j8.35 \times 10^{-3}$$

Therefore, the susceptance of the shunt capacitor is $jB_p = Y_{LC} - Y_L = j6.85 \times 10^{-3}$. The reactance of the series capacitor is $-jX_s = 50 - Z_{LC} = -j666.86$. Finally, the actual values are $C_s = 17.5pF, C_p = 80.5pF$.

The results of both analytical and graphical approaches were found to correlate quite well.

3. Antenna Methodology Development

The 3-dimensional single-loop antenna is built using coated copper wire as shown in Figure 10. The diameter of electrical wire is expressed as the American Wire Gauge (AWG) number. The gauge number is inversely proportional to diameter, and the diameter is

roughly doubled every six wire gauges. Since there is no coated wire for AWG 1-7, AWG 8 is the best choice available with largest diameter.
The antenna is then attached to a piece of foam base to give the antenna rigidity and mechanical support. Without the foam base, the structure of the antenna may not be uniform throughout the testing process (bending of wire etc) and this will have an undesirable impact on the inductance of the antenna and in turn may cause the antenna to be unreliable. Foam is chosen as it is a non-metallic material therefore it will not have any magnetic effect on the antenna. The feed point is made in the middle of the antenna and it will be used to provide interface with the RFID reader/writer.

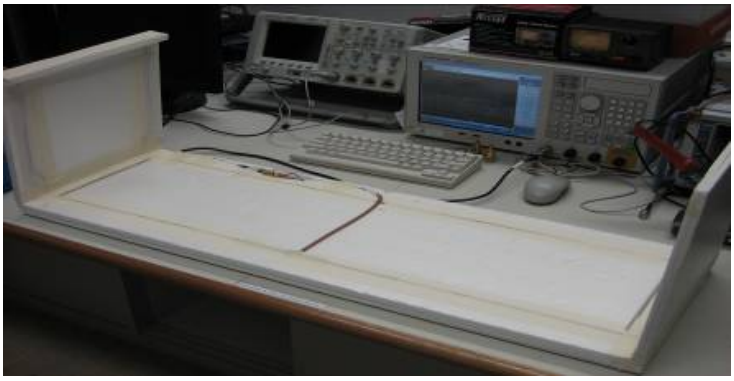


Fig. 3. Prototype of RFID reader antenna

The theoretical derivation provides a good guide to implement the matching network. In practice, a variable capacitor with a large range e.g. $12-80\,pF$ is used to achieve the desired range. Once the matching has been achieved, it is removed and measured and a larger fixed capacitor is used, together with a smaller variable capacitor, to allow for fine turning. Figure 11 shows the actual L-type matching network where the series branch consists of one fixed capacitor of $5\,pF$ with one variable capacitor of $3-11\,pF$ and the shunt branch is composed of two parallel capacitors of $47\,pF$ each with one variable capacitor of $3-11\,pF$ as well.

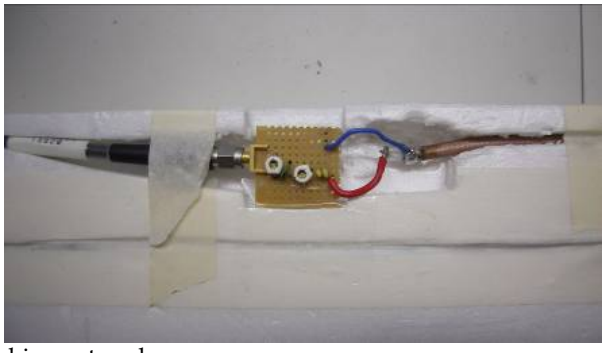


Fig. 4. L-type matching network

3.1 Antenna Measurement

The antenna performance is assessed by measuring following parameters.

3.1.1 Return Loss and SWR

After the antenna is connected to the matching circuit and fine tuned and matched to 50 ohms, we measured the impendence and SWR value of the antenna. Figure 12 displays the impedance is around $50\,\Omega$. And the antenna has a return loss is -14.785dB and SWR is 1.0828 as show in Figure 13.

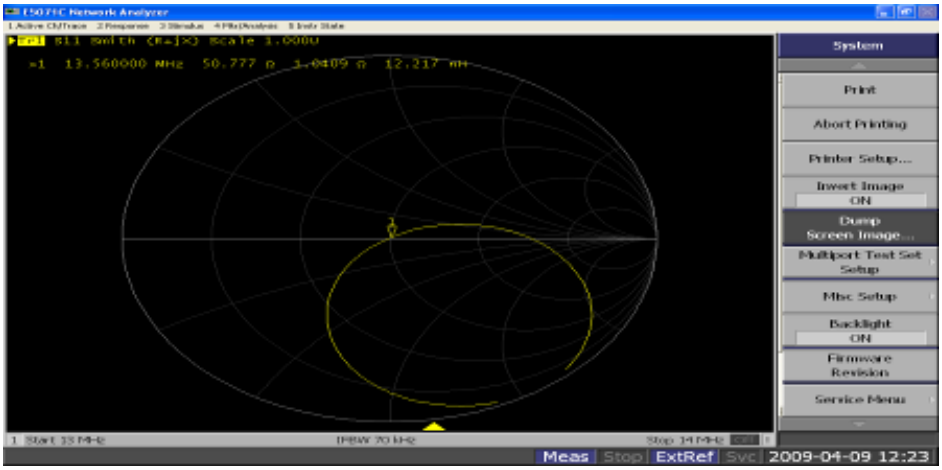


Fig. 12. Antenna impedance with matching network

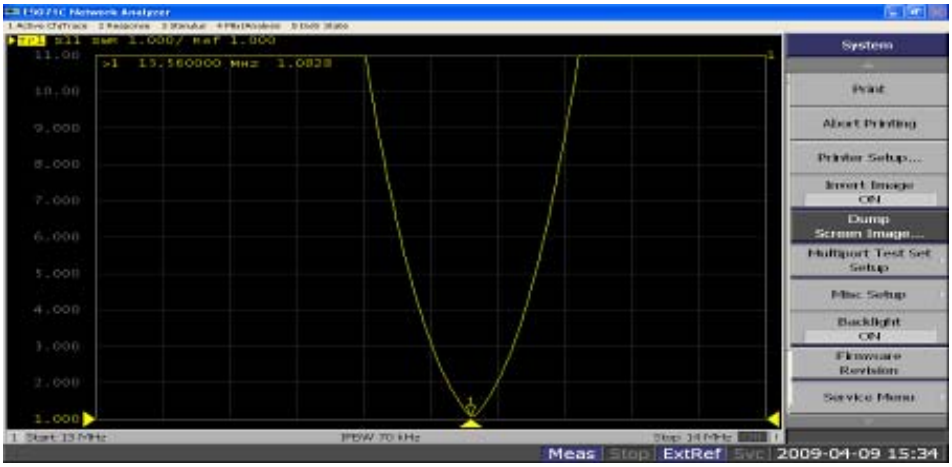


Fig. 13. Antenna SWR

Both results indicate that maximum amount of energy are transmitted to the antenna and this will ensure that data loss are reduced to the minimum.

3.1.2 Quality Factor

The Q factor represents the amount of AC resistance of a system at resonance and it can be determined by measuring the 3-dB bandwidth of the antennas' near field response, frequency sweep between 13 MHz and 14 MHz.

Figure 14 displays the measurement of Q factor. At 13.56MHz labeled by marker 2, the antenna exhibits the resonance. At 13.46MHz and 13.66MHz, the lower and upper -3dB points are found and recorded.

The three frequencies can be used in the formula (2):

$$Q = \frac{f_0}{f_2 - f_1} = \frac{13.56}{13.66 - 13.46} = 67.8 \tag{24}$$

The Q value is smaller than the upper limit 96.8.

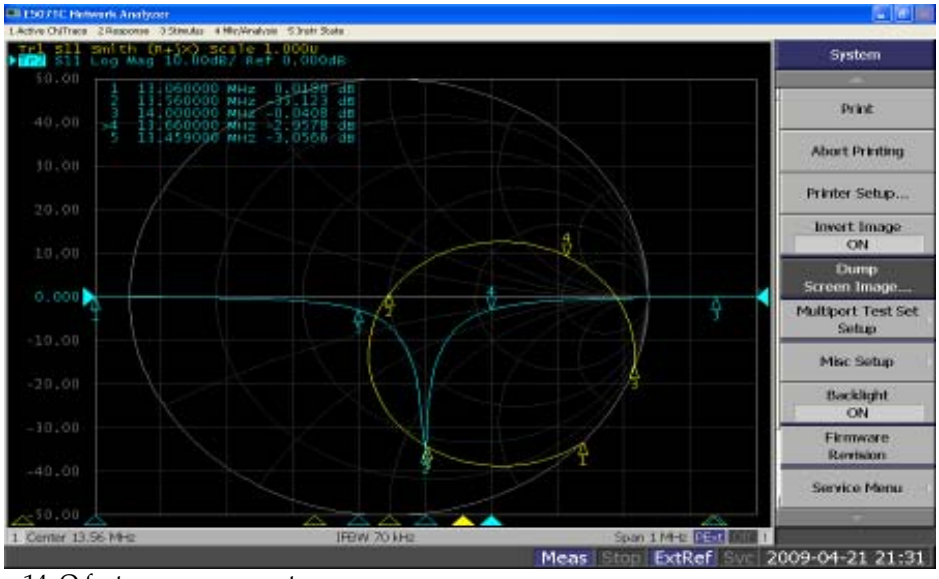


Fig. 14. Q factor measurement

3.1.3 Field Distribution Measurement

The antenna model must be able to produce an electromagnetic field having a magnitude of at least an interrogation threshold of a tag for the entire interrogation region.

The magnetic field strength of the loop antenna at a specified distance determines the reading range of an RFID system with prior selected RFID reader and tag. The stronger the magnetic field, the larger the detection range. Strength of the measured field is represented by the induced voltage. The tag utilized in the experiment is activated only when the induced voltage is greater than 200mV. Magnetic field intensity of the loop antennas is measured at varying test points [19]. The measurement is conducted by using oscilloscope as shown in Figure 15.

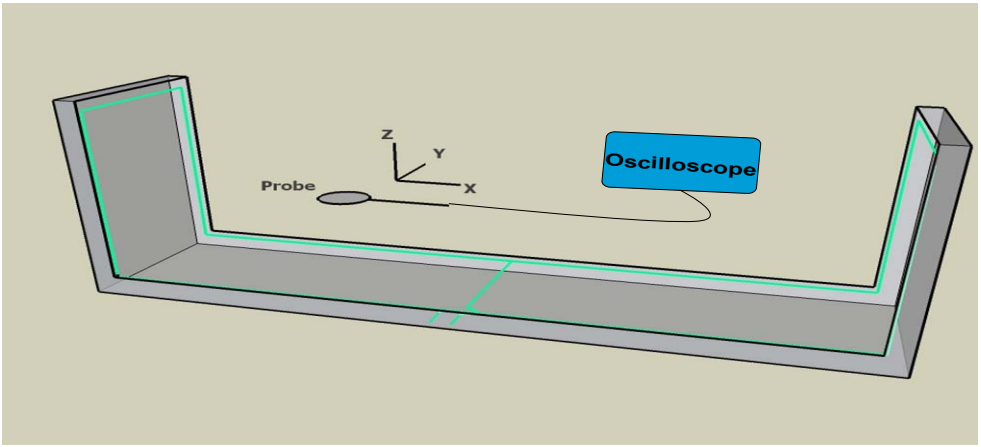


Fig. 15. Setup for field distribution measurement

Figure 16 shows that at the mid cross section x = 44 cm of the antenna, the field distribution in X-direction varies significantly with the z coordinate. The loop antenna generates strong magnetic field in the region closer to the antenna ($z \leq 20$ cm), while the field decreases as distance from the antenna increases. The antenna is able to detect the tag within 30 cm height of the antenna.

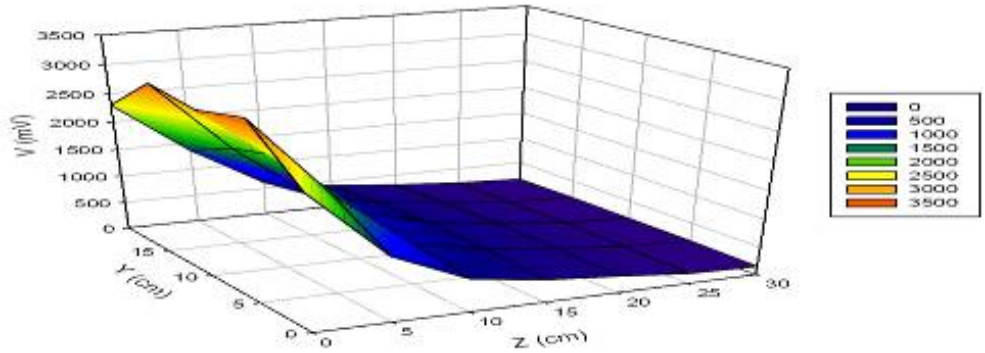


Fig. 16. Magnetic field distribution in x direction

Figure 17 presents that across the mid plane y = 20 cm of the antenna, the magnetic field in Y-direction slightly achieves the peak values at both edges of and slightly drops in the middle of the antenna for the same height. As distance from the antenna increases, the field decreases. There is a black region between x = 33 cm and x = 55 cm as $z \geq 15$ cm.

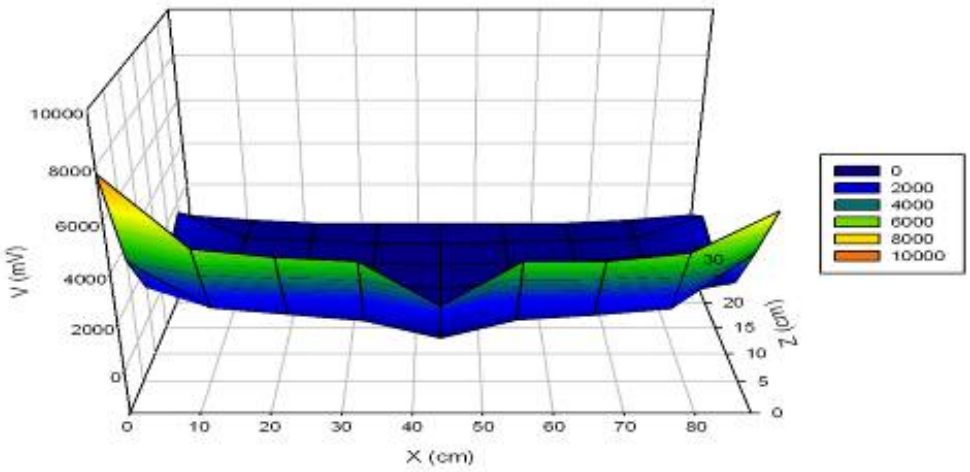


Fig. 17. Magnetic Fielddistribution in y direction

Figure 18 illustrates the magnetic field distribution in Z-direction across the plane z = 18 cm above the antenna. The antenna produces strong magnetic field at both edges. There is a black region in the middle of the antenna.

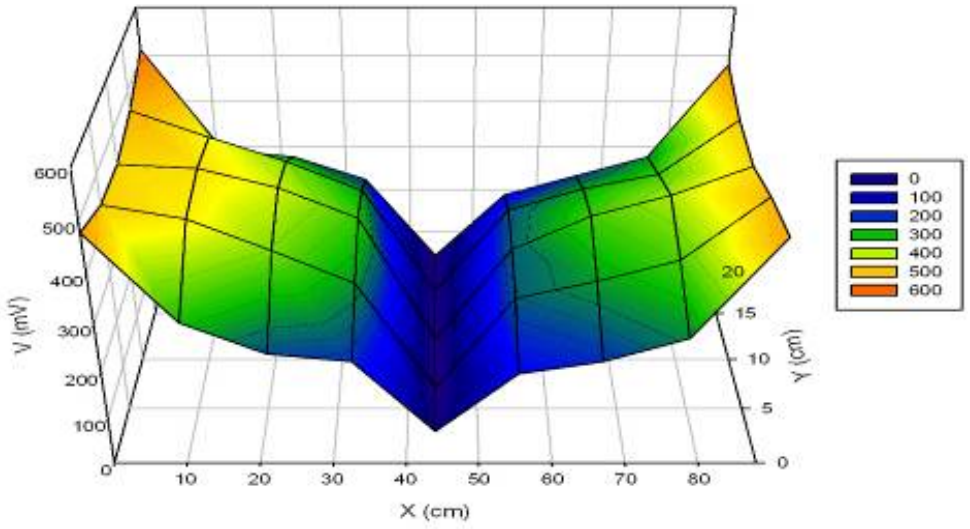


Fig. 18. Three-dimensional field distribution in z direction

3.1.4 Antenna Read Distance

Figure 19 illustrates the magnetic field intensity in x direction measured by moving the tag along three different lines, starting from points (y = 10, z = 5), (y = 10, z = 15), (y = 10, z = 25). These three lines are representative of all positions where the books are placed vertically. The magnetic field intensity is strong at both sides but drops rapidly as moving towards the center. In the middle of the antenna, the field becomes strong again.

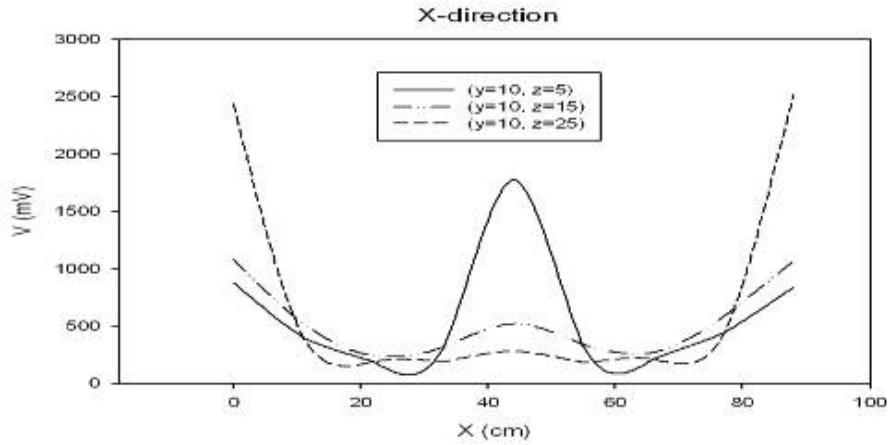


Fig. 19. Magnetic field intensity along different lines in x direction

Figure 20 shows the magnetic field intensity in y direction measured by moving the tag along three different lines, starting from points (x = 22, z = 15), (x = 44, z = 15), (x = 66, z = 15). These three lines are representative of typical positions where books are placed perpendicularly. The two lines in the upper region of the diagram differ in the trace of the line in the lower region.

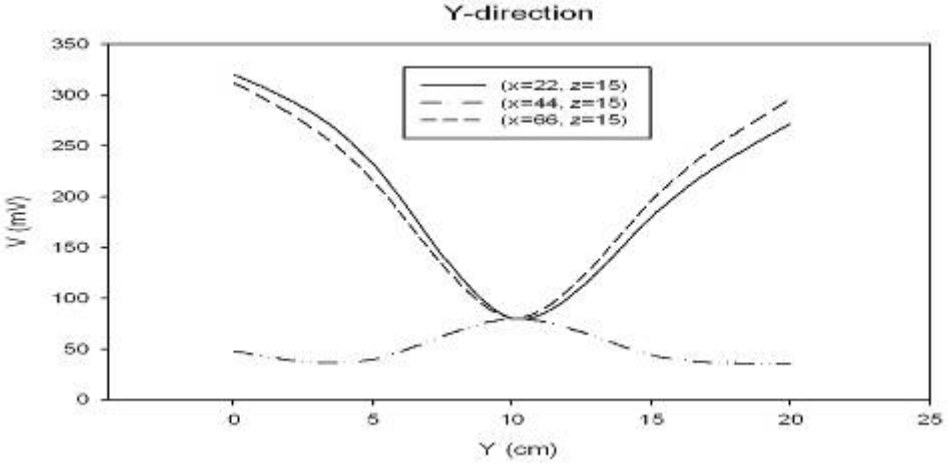


Fig. 20. Magnetic field intensity along different lines in y direction

Figure 21 presents the magnetic field intensity in z direction measured by moving the tag along three different lines, starting from points (x = 22, y = 10), (x = 44, z = 10), (x = 66, z = 10). These three lines are representative of typical positions where books are placed horizontally. The further away from the antenna, the weaker the field becomes.

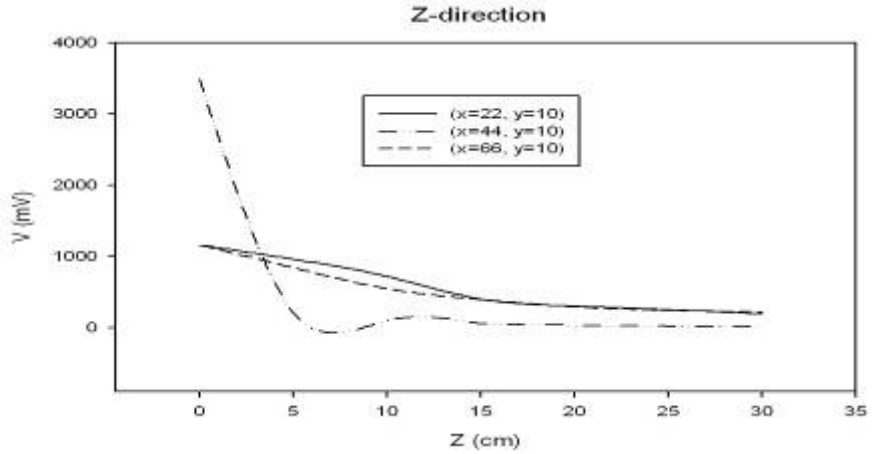


Fig. 21. Magnetic field intensity along different lines in z direction

3.1.5 Various Orientations

The results from the above sections were then used to develop a prototype for RFID book shelf application. The prototype was able to detect books in a number of different orientations within the 3D region with a range of 44 cm from the vertical antenna as shown in Figure 22.

Fig. 22. prototype of smart book shelf

4. Conclusion

This chapter has presented and investigated loop antenna and sensor circuit for HF RFID smart shelf application.
The issues including tag orientations, read range, proximity of metal and other antennas are approached during the antenna design. A 3-dimensional shape of single turn loop antenna is proposed. The optimum antenna size was determined as a trade-off between the magnetic field strength and mechanical constraint. The antenna model has been successfully used to configure an HF RFID smart shelf prototype. The performance test has shown that the antenna is capable of achieving the desired results.
The further research following this project would involve integrating more than one bookshelf into the smart shelf application.

5. Reference

[1] J. Landt, "The history of RFID," *Potentials, IEEE,* vol. 24, pp. 8-11, 2005.
[2] R. Want, *RFID Explained: A Primer on Radio Frequency Identification Technologies*: Morgan & Claypool Publishers, 2006.
[3] D. Miron, *Small Antenna Design*: Newnes, 2006.
[4] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. New York: Wiley, 2003.
[5] V. Chawla and H. Dong Sam, "An overview of passive RFID," *Communications Magazine, IEEE,* vol. 45, pp. 11-17, 2007.
[6] F. Fuschini, C. Piersanti, F. Paolazzi, and G. Falciasecca, "On the Efficiency of Load Modulation in RFID Systems Operating in Real Environment," *Antennas and Wireless Propagation Letters, IEEE,* vol. 7, pp. 243-246, 2008.
[7] H. K. Ryu and J. M. Woo, "Miniaturisation of rectangular loop antenna using meander line for RFID tags," *Electronics Letters,* vol. 43, pp. 372-374, 2007.
[8] S. C. Q. Chen and V. Thomas, "Optimization of inductive RFID technology," in *Electronics and the Environment, 2001. Proceedings of the 2001 IEEE International Symposium on*, 2001, pp. 82-87.
[9] J. Nummela, L. Ukkonen, L. Sydanheimo, and M. Kivikoski, "13,56 MHz RFID antenna for cell phone integrated reader," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, 2007, pp. 1088-1091.
[10] T. Instruments, "HF Antenna Design Notes Technical Application Report," 11-08-26-003, September 2003.
[11] W. Aerts, E. De Mulder, B. Preneel, G. A. E. Vandenbosch, and I. Verbauwhede, "Dependence of RFID Reader Antenna Design on Read Out Distance," *Antennas and Propagation, IEEE Transactions on,* vol. 56, pp. 3829-3837, 2008.
[12] C. Klapf, A. Missoni, G. Hofer, G. Holweg, and W. Kargl, "Improvements in Operational Distance in passive HF RFID Transponder Systems," in *RFID, 2008 IEEE International Conference on*, 2008, pp. 250-257.
[13] D. C. Yates, A. S. Holmes, and A. J. Burdett, "Optimal transmission frequency for ultralow-power short-range radio links," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 51, pp. 1405-1413, 2004.

[14] Q. Xianming and C. Zhi Ning, "Proximity Effects of Metallic Environments on High Frequency RFID Reader Antenna: Study and Applications," *Antennas and Propagation, IEEE Transactions on,* vol. 55, pp. 3105-3111, 2007.
[15] S. R. Best, "A discussion on the quality factor of impedance matched electrically small wire antennas," *Antennas and Propagation, IEEE Transactions on,* vol. 53, pp. 502-508, 2005.
[16] Y. Lee and P. Sorrells, "AN680: MicroID 13.56 MHz RFID System Design Guide," Microchip Technology Inc., 2004.
[17] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Hoboken, N.J.: Wiley-Interscience, 2005.
[18] R. Ludwig, P. Bretchko, and G. Bogdanov, *RF Circuit Design: Theory and Applications*, 2nd ed.: Prentice Hall, 2007.
[19] A. Cai, Q. Xianming, C. Zhi Ning, and L. Boon Keng, "Performance assessment of printed RFID reader antenna," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, 2007, pp. 301-304.
[20] Noax, "Experiment #52 — SWR Meters," *QST,* May 2007.
[21] C. Bowick, *RF Circuit Design*. Burlington, MA: Newnes, 1997.

16

Robustness enhancement of networked control systems

Michał Morawski and Antoni Zajączkowski
Technical University of Łódź
Poland

1. Introduction

In the last decade we can observe a growing interest in design and implementation methods for Networked Control Systems (NCS). Typically in an NCS, a plant is both controlled and monitored by a remote computer system connected with the plant via a communication channel performed by a computer network (Hristu-Varsekelis & Levine, 2005). This architecture of the control system differs significantly from the classical one, where all components of the system are attached directly to the control plant and exchange data using some wiring system. NCSs gain popularity due to the fact that they can be implemented more cost effective and can provide extended functionality as compared with classical control systems. Application of computer networks for a data exchange in automated production systems is not a new idea and during the last decades several industrial standards of computer networks have been developed. Industrial networks are typically based on asynchronous links (fieldbus) or technologies developed for specific areas. To this last category belong Profibus, CAN, ARINC, WorldFIP, and many others.
The most important feature of these industrial networks is that they guarantee bounded transmission delays. However, NCSs based on industrial standards suffer from some disadvantages, like high installation and maintenance costs, excessive weight of physical links and difficulties with scalability and redundancy. One of the solutions enabling partial avoidance of these problems is application of standard, inexpensive, easy to purchase and replace devices typically used for office networking. To this class belongs Ethernet which currently dominates general networking and seems to become an industrial standard in the nearest future (Decotignie, 2005; PROFIBUS Nutzerorganisation, 2006; Andersson, Brunner & Engler, 2003, IEC 61850, 2002).
The guided media, that form a basis of today plants, although quite reliable in nature, have several disadvantages like limited scalability, quite high installation and maintenance costs, low flexibility. Networks are more and more often applied in new domains, where usage of guided media is impossible or inconvenient. These domains include home or building automation, military systems, inter vehicle communication (VANET), sensor networks and others (Murthy & Manoj, 2004; Schoch, Kargl & Weber, 2008; Callaway, 2004).
Therefore more and more often the unguided (wireless) communication is seen as a significant complementary solution for wired communication (Willig, Matheus & Wolisz, 2005).

Such solutions based on 802.11 (Gast, 2005; WAVE), 802.15 (Gislason, 2008; Bluetooth Specification,ZigBee Specification, 2008), or proprietary technologies are relatively inexpensive, easy to deploy, and often allow to avoid installation of power supply cables. Some of them are commercially available like WISA (Frey, 2008). The wireless media however suffer from low reliability, limited capacity and are prone to noise and interferences.

Solving the problem of exploiting limited networks resources, one should keep in mind, that fulfilment of specific requirements for any kind of homogenous networks is unrealistic in real today plants. Simultaneous application of technologies which differ in their maturity of even more than one decade – thus heterogenous – requires to control a traffic at higher layers – network or even application (gateways).

It is well known that incorrect operation of the control system due to some hardware failure can be dangerous or even catastrophic. Therefore in order to increase reliability of computer controlled systems, redundancy realised by duplication or multiplication of sensors, actuators and selected processing units is applied, especially in the case of systems responsible for control of critical processes.

In network based control systems situation is additionally complicated by the possibility of network failures and stochastically time-varying transmission delays introduced into feedback loops by a network. Hence, resiliency of communication is crucial for satisfactory operation of these systems. It is obvious that multiplication of communication links is not sufficient to assure proper operation of NCSs and should be complemented by suitable traffic engineering and some procedures at the application level. While there is a need to keep network introduced delays below a certain bound, traffic engineering is uncomplicated when using point-to-point links, more complicated when using shared media and quite complex in the case of wireless links prone to collisions, noise, interferences.

In this chapter only an application layer resiliency is considered. The complement of solution in network layer was proposed previously (Morawski, 2005, Morawski, 2006b, Morawski, 2007), and is only briefly discussed here. The physical layer (i.e. modulation and coding) are defined by the appropriate standards and therefore cannot be easily altered. The significant influence of the link layer on latencies cannot be exploited in practice due to limitations of available hardware, firmware, and drivers. In particular, implementation of the advanced 802.11e substandards like burst acknowledgements, no acknowledgements, HCF is extremely seldom, and software support for wider deployed WMM extensions is manufacturer dependent, diverse and compatible in theory only. However, it is worth to notice, that some companies adjust lower (1, 2) layers to their needs (Frey, 2008). We have decided to not follow this path, due to cost reasons. Further, we discuss resiliency of NCSs with networks implemented using standard, inexpensive, off-the-shelf devices and systems. It is crucial to underline here, there is no "silver bullet" – none single method is enough to achieve robust NCSs.

Organization of the chapter is as follows: in Section 2 delays introduced by the computer network are discussed, in Section 3 the compensation of network introduced delays at the application layer is presented together with experimental results illustrating properties of the laboratory NCS with the Magnetic Levitation System. These results are next complemented by addition of hardware redundancy as discussed in Section 4. Section 5 presents a short description of an approach to the network layer compensation of latencies, and Section 6 contains conclusions.

2. Network introduced delays

In many publications (e.g. (Stallings, 2002; Welzl, 2005)) inevitable data transmission latencies introduced by computer networks are modelled statistically by some well-known distributions like exponential, Poisson, Pareto, gamma, Erlang, etc. Such models can be accepted in the case of the Internet network, where the main source of delays are queues in intermediate nodes. In the real-time traffic generated by the NCSs, esp. those using wireless channels, the significant influence of the media access procedure and layer 2 retransmissions can be observed and experimentally obtained latency distributions cannot be approximated by aforementioned models (Natori, Oboe & Ohnishi, 2008). This observation was confirmed by the number of experiments performed in our laboratory (Morawski, 2006a).
Fig. 1 presents the probability density functions (PDF) of round trip times (RTT) obtained by the measurements in the network with the Access Point 802.11bg without the "e" extension, two Ethernet switches and the PIX firewall. Three depicted graphs correspond to three independent measurements performed in three short time intervals shifted by about 15 minutes one from the other. Fig. 2 presents the same RTT data as the discrete-time functions. The considerable increase of RTTs during the first measurement can be interpreted as the effect of an unidentified traffic in the shared network or disturbances. It is necessary to notice, that reduction of the sampling period can induce a significant increase of delays due to retransmissions, a contention procedure for media access control and buffering in network card drivers (Fig. 3). These phenomena are exceptionally important for wireless media, but the problem exists in any shared media. This problem can be diminished in network layer using multiple transmission channels and suitable traffic engineering mechanisms (Morawski, 2005), where link costs (delays) are considered as uncertain numbers. From the above discussion follows that selection of the sampling period in network based control systems requires additional analysis as compared with that performed (Åström & Wittenmark, 1997; Wang & Liu 2008) in the case of classical computer controlled systems.
It is worth mentioning here, that in our experiments we have applied the general purpose inexpensive hardware typically used for office networking and the ultimate goal of our research was to investigate if application of such hardware enables successful deployment of NCSs with high dynamic controlled plants. In our experimental NCS as a plant served Magnetic Levitation System (MLS) which is structurally unstable system of high dynamics (Morawski & Zajączkowski, 2007). From analytic considerations and simulations of the stand-alone closed-loop control system with MLS follows that stable operation of this system requires frequent sampling of the measured output and fast floating point computations of the control feedback. Hence, compensation the influence of network introduced delays is critical for proper operation of NCS with MLS. Taking into account results concerning network introduced delays and requirements concerning the choice of the sampling period for digital control of MLS we have concluded that statistical approach to the design of network based controllers for MLS is less useful then the predictive control approach described in (Liu Xia & Rees, 2005; Yang, Wang & Yang, 2005).

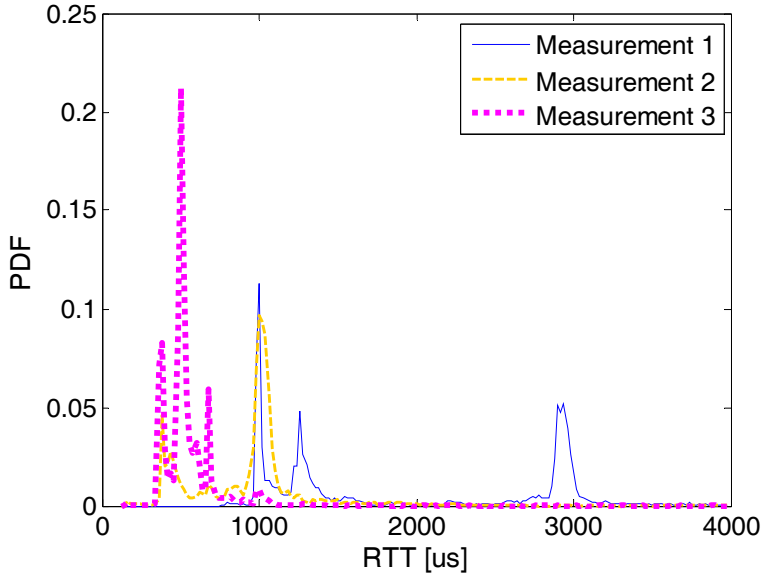


Fig. 1. Probability density functions of RTT obtained in the laboratory network

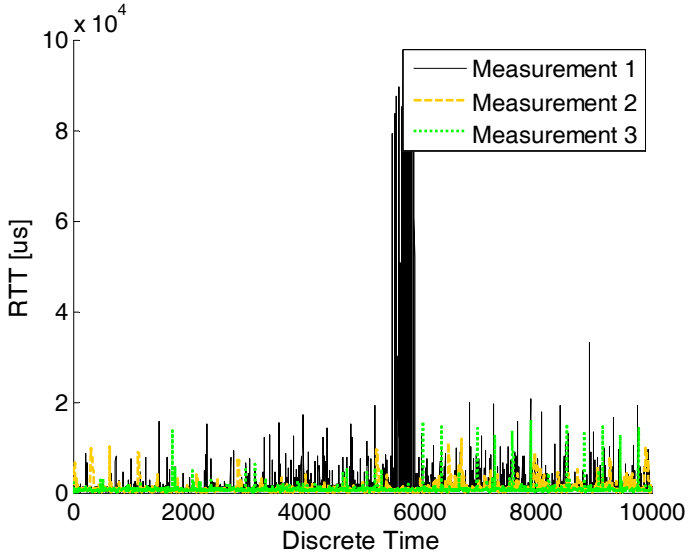


Fig. 2. RTTs time series in the laboratory network obtained for the same data as depicted above

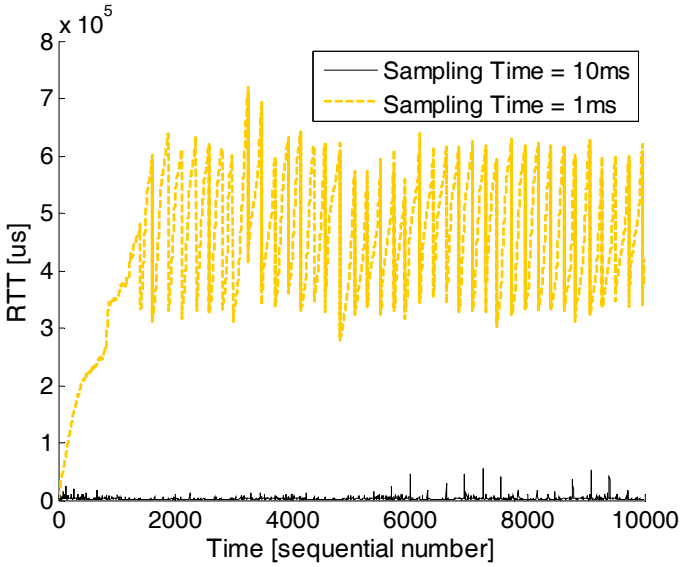


Fig. 3. Effect of a congestion in some (wireless) link. Long delays resulting in unstable operation of the system. Such system is useless in practice.

4. Application Layer Compensation

We consider a discrete-time, linear, time-invariant control plant modelled by the following state and output equations:

$$x\big(k+1\big) = Ax\big(k\big) + Bu\big(k\big) \tag{1}$$

$$y\big(k\big) = Cx\big(k\big) \tag{2}$$

where $x(k)$ is the state vector, $u(k)$ is the vector of control inputs, $y(k)$ is the vector of measured outputs and A, B, C are matrices of suitable dimensions.
Further we assume that the model of the plant is completely reachable and completely observable (Åström & Wittenmark, 1997; Wang & Liu 2008). In other words we assume that the pair (A,B) is completely reachable and the pair (A,C) is completely observable. From the first assumption follows that using the pole shifting method or the linear-quadratic optimal control theory (Åström & Wittenmark, 1997) we can design a linear state feedback

$$u(k) = K_{\mathrm{s}}x(k) \tag{3}$$

such that the closed-loop system

$$x(k+1) = (A + BK_{\mathrm{s}})x(k) \tag{4}$$

is asymptotically stable with desired dynamic properties.
Second assumption guarantees that we can design a Luenberger observer (Åström & Wittenmark, 1997) which estimates the state vector of the plant on the basis of the measured output and control input. Further we consider the implementation problem of the control law (3) using an NCS (Welzl, 2005; Srikant, 2003) shown in Fig. 4.

A networked control system consists of a plant, controller and computer network across which all sensor and actuator data must be sent (Hristu-Varsekelis & Levine, 2005). In this system, the plant is equipped with some computer (a local controller) responsible mainly for sending and receiving data to and from a remote controller using a communication channel provided by the computer network. If a computer network is used to exchange information between the local and remote controller then some transmission delays are introduced into the feedback loop. We distinguish two transmission delays: the sensor to controller delay τ_{sc} and the controller to actuator delay τ_{ca} and we assume that there exist nonnegative integers (Liu, Xia & Rees, 2005; Yang, Wang & Yang, 2005) n_{sc} and n_{ca} such that the total transmission delay satisfies the inequality

$$0 < \tau_{\mathrm{sa}} < \Delta n_{\mathrm{sa}} \tag{5}$$

where $n_{\mathrm{sa}} = n_{\mathrm{sc}} + n_{\mathrm{ca}}$ and $\tau_{\mathrm{sa}} = \tau_{\mathrm{sc}} + \tau_{\mathrm{ca}}$.

Remark. The delay τ_{ca} includes the time spent by the remote controller for execution of the control algorithm.

Second assumption concerning operation of the network is as follows: the number of lost packets on the route from the plant to the controller and on the route from the controller to the plant are less than n_{sc} and n_{ca}, respectively. The controller operates in the event driven mode. It means that that a control algorithm is started when the new packet with measured data is received. Sensors work in the clock (time) driven mode, that is samples of the measured output signals are computed in the time instants $t_k = k\Delta$, where k is the integer sample number, Δ is a sampling period. Therefore sensor is responsible for stringent time properties. Actuators operate both in the clock and event driven modes. It means that control values are received when a packet from the controller arrives and the new control value is applied in the immediate sampling instant.

Let $\mathrm{T}_k = [t_k, t_{k+1})$. In this time interval the local computer can receive zero, one or more – at most n_{sa} data packets with the new values of the control input. If during this interval no packet arrives then the preceding value of the control input will be applied in the next time interval $\mathrm{T}_{k+1} = [t_{k+1}, t_{k+2})$.

If in the interval T_k the local computer receives more than one data packet with the control values then the most recent value will be applied and earlier (i.e. based on the earlier samples, not on earlier received) values will be neglected. These scenarios are illustrated in figures 5.

Suppose now that on the basis of the plant model (1) the linear state feedback (3) was designed. If this control law is used in the NCS in which random time-varying transmission delays occur then control values are applied with the delay $r(k)$ which takes values from the set $R = \{1,2,\ldots,n_{\mathrm{sa}}\}$.

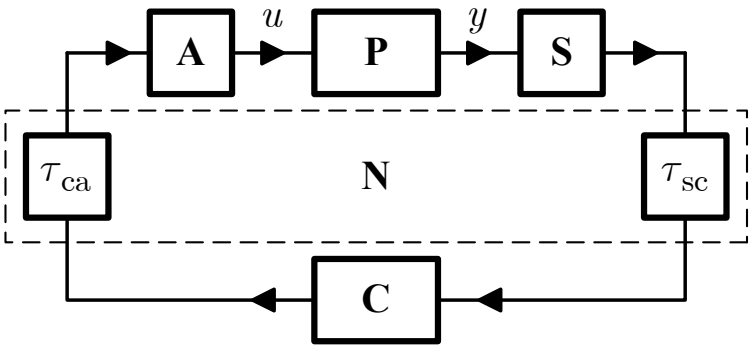


Fig. 4. One channel closed-loop NCS. P – plant, S – sensors, A – actuators, C – remote controller, N – computer network.

Note that $r(k)$ corresponds to the time instant $t_k = k\Delta$ and to the sample $y(k)$ of the output. It is easy to see that the best situation occurs when $r(k) = 1$, but even in this case application of the new control value by the actuator is delayed. If in an NCS no mechanisms exist responsible for compensation of the influence of the random delays introduced by the network, then the behaviour of the closed-loop system degrades significantly and in some cases such a system can suffer from stability loss. Hence, compensation of the influence of this delay phenomenon belongs to the most important problems encountered in the area of network based control systems.

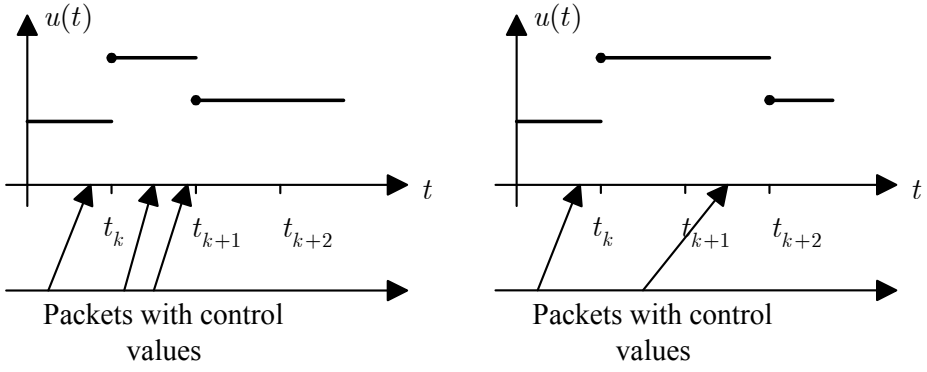


Fig. 5. Packets with control data are received in the interval $\mathrm{T}_k = [t_k, t_{k+1})$ (on the left). No packet with control data is received in the interval $\mathrm{T}_k = [t_k, t_{k+1})$ (on the right)

As was mentioned above the remote controller should be equipped with some mechanism compensating the influence of the transmission delays introduced by the computer network. One of the mechanisms proposed recently (Liu, Xia & Rees, 2005; Yang, Wang & Yang, 2005) is the method of prediction described in what follows. Consider the NCS depicted in Fig. 4

and assume that the local computer has a buffer for storing a finite number of control values. Assume also that the remote controller stores the model of the plant represented by the triple (A, B, C).

Suppose that the controller received the sample $x(k)$ of the plant state together with the time stamp k corresponding to this sample. If we know the state $x(k)$ then using (3) we can compute the control value $u(k)$. The next predicted value $\hat{x}(k+1)$ of the state can be computed according to the equation (1)

$$\hat{x}(k+1) = Ax(k) + Bu(k). \tag{6}$$

Applying (3) again we obtain the predicted value $u(k+1)$ of the control signal

$$u(k+1) = K_{\mathrm{s}}\hat{x}(k+1). \tag{7}$$

Repetition of these computations $n_{\mathrm{sa}} - 1$ more times with respective exchange of arguments gives the last predicted values of the state and control

$$\hat{x}(k+n_{\mathrm{sa}}) = A\hat{x}(k+n_{\mathrm{sa}}-1) + Bu(k+n_{\mathrm{sa}}-1) \tag{8}$$

$$u(k+n_{\mathrm{sa}}) = K_{\mathrm{s}}\hat{x}(k+n_{\mathrm{sa}}). \tag{9}$$

As a result we obtain the finite sequence of control values

$$\mathbb{U} = u(k), u(k+1), \ldots, u(k+n_{\mathrm{sa}}). \tag{10}$$

This sequence and the corresponding time stamp k attached is sent via the computer network to the local computer. This computer compares his current discrete time with the received time stamp and computes the value of $r(k)$. From the control sequence the element of index is selected and applied in the immediate sampling instant – figures 5.

The experimental MLS described in (Morawski & Zajączkowski, 2007) is connected to the typical PC (local computer) that reads data from the height sensor and controls the voltage applied to the winding of the electromagnet. The height of levitation is measured by the optical sensor connected to the 12-bit AD converter. The local computer sends to and receives data from the remote computer system responsible for computation of the control values. These two computers exchange data via a computer network using UDP packets. The local computer sends packets at the rate of 1024 packets per second. Each packet contains the current and past samples of the height of levitation, the corresponding time stamp and the respective control values. The packets produced by the remote controller are asynchronously sent to the local computer and can be considered as responses to the packets containing samples of the plant output. Each packet sent by the remote controller contains

the copy of the respective time stamp and the finite sequence of $n_{\mathrm{sa}} = 10$ [1] control values computed according to the presented method of prediction. From this follows that the size of the UDP packet is 72 B which results in the throughput estimate of 901120 bps.
The estimates of the bandwidth including media access procedure and gaps (<2Mbs) are sufficient to perform data transmission using typical modern Ethernet or WiFi (802.11) networks. The above approximation does not take into account repetitions, thus may be underestimated in shared media case. Nonetheless in any case the available bandwidth is far from saturation. The local computer evaluates the difference between its current time and the received time stamp. This quantity interpreted as the Round Trip Time (RTT) is expressed in milliseconds and is used as an index needed for selection of the suitable element from the received sequence of control values $\mathbb{U}$ (10).
Fig. 6 presents time variability of RTT (thus index in the table obtained using (6), (7)) seen by the local computer. Figures 7 present the graphs of the ball position when the network based control loop uses the Ethernet and WiFi networks respectively. The results of operation of the network based systems are compared with the standalone system.

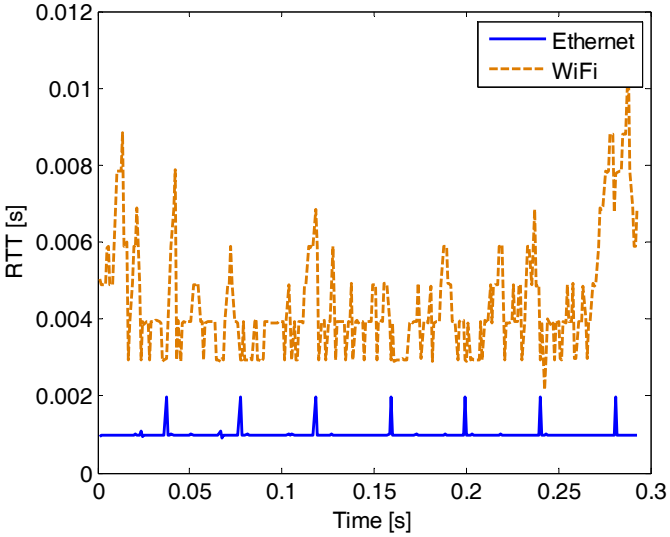


Fig. 6. RTT seen by the actuator. These values can be interpreted as indices in vector obtained using (10).

[1] In fact, the first element is included in this vector $\mathbb{U}$ (10) but is not used as a control value. This value can be used for the assessment of the closed loop system properties.

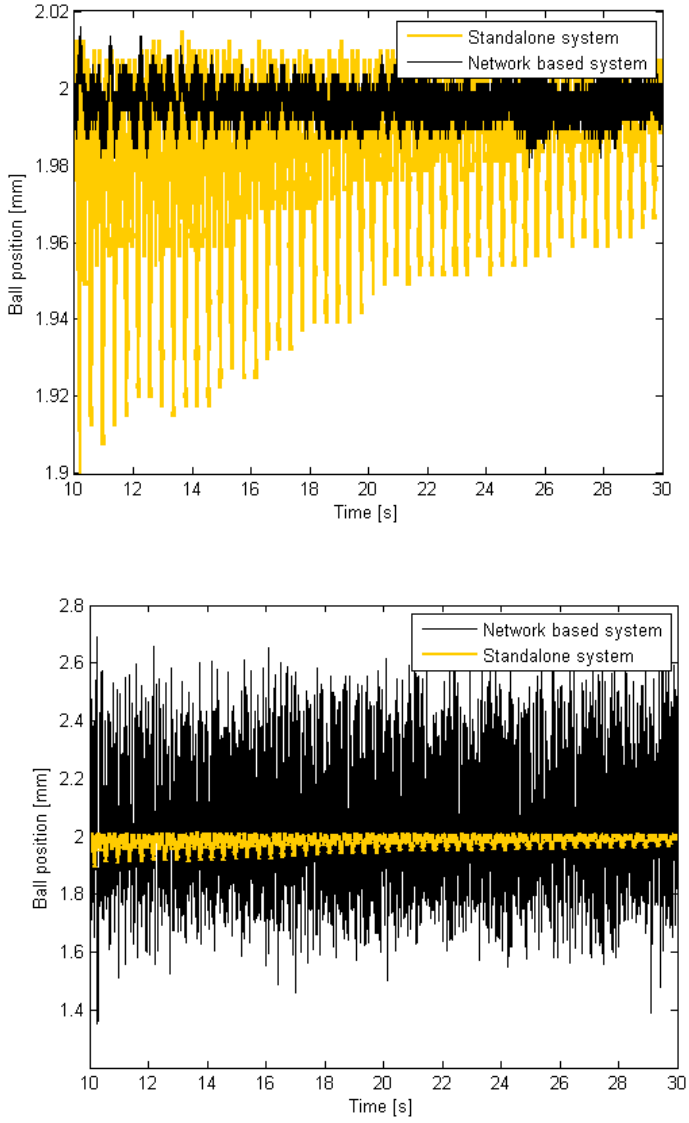


Fig. 7. Quality of control of magnetic levitation system obtained using methods described in the chapter using Ethernet network (above) and mixed (WiFi and Ethernet) network (below). The better results of NCS in case of Ethernet networks are the results of using observer to obtain ball velocity instead of the simple derivation via filtering. WiFi based controller without compensation does not work at all.

5. Application layer redundancy

The aforementioned results were obtained for the case of unicast communication. In order to increase robustness of the control system, we have proposed multicast communication originated from the local computer and responses as unicast communication, originated by the remote computer. In the case when the remote computer has multiple different interfaces, such solution does not increase the traffic volume – hence delays. However such redundancy concerns only communication links and does not concern the control process. If several computers are used in the same distributed control system the data traffic increases, but the redundancy level increases also – it concerns both links and the control process. The intermediate solution can be based on the traffic engineering algorithms very shortly described in the next Section, where traffic is limited at the "intelligent" network level (in middleware).
The scheme of the redundant system, and the results of experiments concerning the switching over the active remote computer are presented in Fig. 8. The glitch seen on this chart is the effect of the different internal state of the observer at the moment of this structural change.
The redundancy lowers slightly the quality of control due to an increase of delay statistics, but this phenomenon is negligible. However, the aforementioned impulse is too large to keep the system stable when switching from Ethernet to the WiFi network. In the remaining cases this switching over the networks is possible.

6. Network layer compensation

The described method of the compensation the network induced delays have limited capabilities because while increasing n_{sa} the quality of prediction and plant state estimation performed by the observer degrades significantly. Although theoretically n_{sa} can take any value, in practice it should not exceed 5 – 10 (depends on the particular problem, for MLS see Fig. 6 and 7). Usually occasional violation of this limit does not result in system misbehaviour, but in the case of bursty delays of a data series, a system failure is practically unavoidable. Therefore the network layer compensation should be applied as a complementary mechanism enhancing robustness of the NCSs.
Network layer observes the destination of a communication as some additive, multiplicative or convex metric built on the basis of some link cost attributes (Alkahtani, Woodward, & Al-Begain, 2003). The attributes can be considered as a vector of technical, economical and other properties of the link. In classical networks (or high performance networks) the quality of the link (i.e. cost of the link) is described by its existence only, or by its physical bandwidth. Other properties are used infrequently, esp. link delay – the most obvious attribute in considered areas of applications. Moreover forwarding packets based on the minimum delay is proven (Gallager, 1977) to be optimal. Unfortunately the information about delays is always outdated and former approaches exploiting this rule have been failed (Khanna & Zinky, 1989).

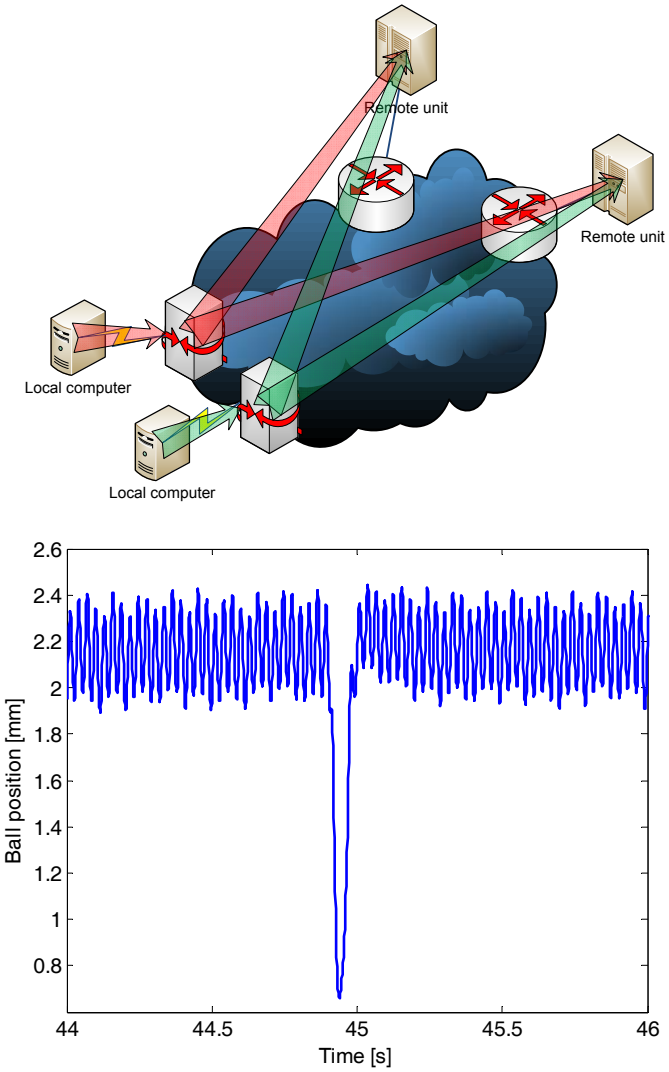


Fig. 8. Application layer redundancy. The scheme of the system (above). Process of switching the active remote controller (below).

Here we shortly discuss another approach developed recently by Morawski (Morawski, 2005; Morawski, 2006b; Morawski, 2007). His approach is based on the method of suitable utilisation of multiple communication channels that dissipates congestions (originated by media access procedure, noise, retransmissions, foreign flows interference, etc.) by balancing the traffic gradually in contrast to previous attempts. The algorithm was initially designed for general networks and adopted recently to the real time traffic (Morawski, 2007).

As was mentioned above, the quality of links can be expressed as overall transmission latencies associated with particular links. These latencies are highly variable quantities computed as the sum of propagation, transmission, media access, processing and queuing times. While the propagation, transmission and processing times are less or more constant, the remaining ones are not, and the media access time necessary for retransmissions influences directly the queue depletion ratio. Therefore, we use the sum of the queue depletion time and the constant components of the latency as the input $s \mapsto \eta(s)$ of the first-order, linear, low-pass IIR filter:

$$\xi(s+1) = \big(1-\alpha\big)\xi(s) + \alpha\eta(s) \tag{11}$$

where s is the discrete-time corresponding to the subsequent time instants, $\alpha \in \big(0,1\big) \subset R$ is the constant defining dynamics of the filter, and $s \mapsto \xi(s) \in R$ denotes the estimate of the average latency introduced by the link.
Additionally we take into consideration the average standard deviation v (variability) of ξ computed as follows:

$$\begin{aligned} v^{+}(s+1) &= \beta\big(\eta(s) - \xi(s)\big) + \big(1-\beta\big)v^{+}(s) \quad \eta(s) \geq \xi(s), \\ v^{-}(s+1) &= \beta\big(\xi(s) - \eta(s)\big) + \big(1-\beta\big)v^{-}(s) \quad \text{otherwise}. \end{aligned} \tag{12}$$

It is well known (Welzl, 2005; Stallings, 2002; Srikant, 2003) that the network model does not belong to the class of statistically invariant systems which, according to the definition of the standard deviation, fulfil the condition $v^{+}(s) = v^{-}(s)$. The results of approximations computed by (11) and (12) are presented on Fig. 9. The values of ξ, v^{+}, v^{-} create an uncertain quantity defining the link quality (or link cost), where ξ is the most likely (or central) value, $\xi + v^{+}$ is the estimate of the upper limit of variability of ξ, and $\xi - v^{-}$ is the estimate of the lower limit. Possible geometric interpretation of such uncertain value is presented on Fig. 9. The uncertain sum of the link costs gives the uncertain metric of the path (Hanss, 2005). The packets are forwarded randomly inversely proportional to the respective value of the path metric, that can be expressed as a degree of occlusion of particular metrics.
While $\eta(s)$ (11) is highly variable, the value $\xi(s)$ changes slower, but changes every time instant s also. The algorithms which decide when to report a new value of the link cost, and the new updated value of this cost, and corresponding adaptation mechanisms are discussed in detail by (Morawski, 2005; Morawski, 2006b, Morawski, 2007).
The proposed algorithm can be used with any routing protocol which works in event triggered fashion, i.e. except those ones, that have only timed updates. The algorithms has the same properties like the soft handover, therefore does not induce the route flapping phenomenon.
Therefore the quality of this algorithm is far better than in the classical version (Khanna & Zinky, 1989). The quality was measured in the simulations using Network Simulator ver. 2 by comparing the drop/receive ratio in the case of the connectionless traffic and by evaluat-

ing the variability of the main control value of TCP protocol (cwnd) for the connection oriented traffic. The variability of cwnd can be successfully evaluated only by highly equipped appliances, that can use sophisticated versions of TCP, that require many resources. Because the most of the TCP stacks available for microcontrollers are handicapped (due to necessity of resource conservation) and do not use aforementioned sophisticated flow control and, even in such case, usage of connectionless traffic results in closer to assessments observed in real networks. The quality of the connectionless traffic was assessed using an impulse or selfsimilar traffic with different statistical properties. In all cases the quality of the algorithm outperforms the standard solutions. Finally, the laboratory network was tested.

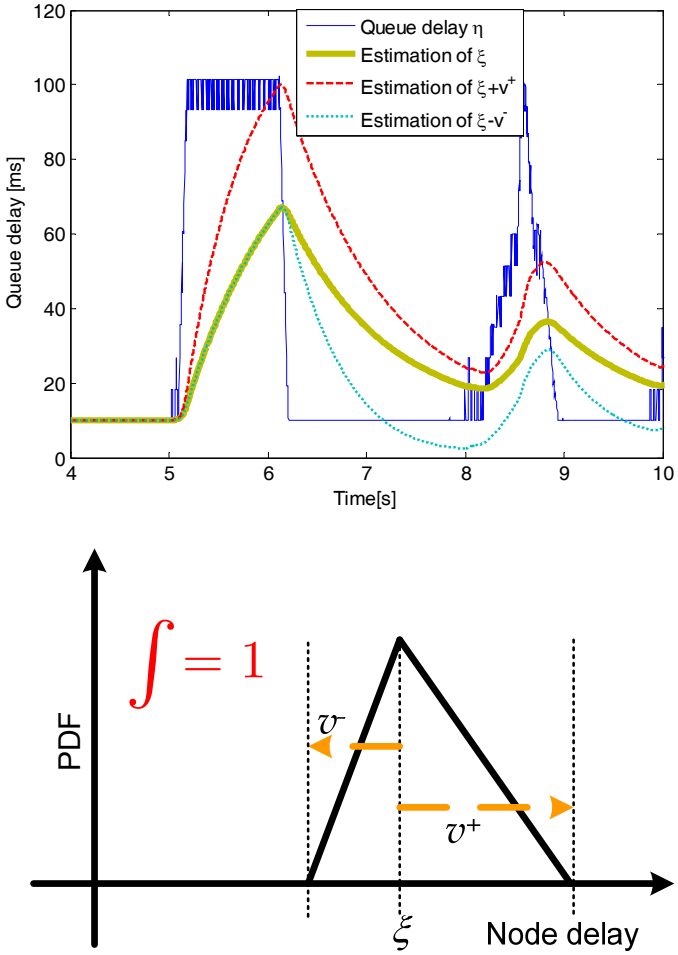


Fig. 9. Approximation of the link cost using equations (11) and (12) (above). Possible geometrical interpretation (below)

7. Conclusions

Neither suitable traffic engineering nor application layer compensation applied alone, cannot keep closed loop latencies of a NCS within desired bounds. However, proper combination of algorithms in the network layer and application layer can cause, that NCSs are applicable for network based control of highly dynamic systems.

8. References

Alkahtani, A.M.S, Woodward, M. E. & Al-Begain, K. (2003). Measurement Based Optimal Multi-path Routing, *Proceedings of the Fourth Annual Network Symposium PostGraduate Networking Conference (PGNet)*, Liverpool – England, 16-17 June 2003;

Andersson, L.; Brunner, C. & Engler, F. (2003). Substation Automation based on IEC 61850 with new process-close Technologies, *Proceedings of Power Tech Conference,* art 6. ISBN, Bologna, Italy, 23-26 June 2003, IEEE;

Åström, K.J. & Wittenmark, B. (1997). *Computer-controlled systems*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd ed.;

Bluetooth Specification, *Bluetooth Special Interest Group*, www.bluetooth.com, www.bluetooth.org;

Callaway, E.H. (2004). *Wireless Sensor Networks: Architectures and Protocols*, CRC Press;

Decotignie, J.-D. (2005). Ethernet-based real-time and industrial communications. *Proceedings of IEEE,* Vol. 93, No. 6, (Jun 2005), 1102-1117;

Frey J.-E. (2008). WISA – Wireless Control in Theory, Practice and Production, *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Keynote speech, Hamburg, Germany, 18 Sep. 2008, IEEE;

Gallager, R.G. (1977). A Minimum Delay Routing Algorithm using Distributed Computation, *IEEE Transactions On Communication*, Vol. 25, No. 1, (Jan. 1977), 73—84;

Gast, M. (2005). *802.11 Wireless Networks: The Definitive Guide*, O'Reilly Media Inc, Place, 2nd ed.;

Gislason, D. (2008). *ZigBee Wireless Networking*, Newness, 2008;

Hanss, M. (2005). *Applied Fuzzy Arithmetic*, Springer, 2005;

Hristu-Varsekelis, D. & Levine, W.S. (2005). *Handbook of Networked and Embedded Control Systems,* Birkhäuser, Boston;

IEC 61850 Communication Networks and systems in substations, IEC standard in ten main parts, first parts published in 2002;

IEEE Standard for Wireless Access in Vehicular Environments (WAVE), P1609.1, P1609.2, P1609.3, P1609.4;

Khanna, A. & Zinky, J. (1989). The revised ARPANET routing metric, *Proceedings of SIGCOMM*, 45-89, Aug. 1989, ACM;

Liu, G. P.; Xia, Y. & Rees, D. (2005). Predictive control of networked systems with random delays, *Proceedings of the 16-th IFAC World Congress*, We-M15_TO/2, Prague, Czech Republic, 4-8 July 2005;

Morawski, M. (2005). Uncertain metrics applied to QoS multipath routing, *Proceedings of 5-th International Workshop on Design of Reliable Communication Networks, DRCN'05*, 353-360, Island of Ischia, Naples - Italy, 16-19 October 2005, IEEE;

Morawski M. (2006a). Analysis of short Latencies in Industrial Network Environments, *Journal of Applied Computer Sciences (JACS)*, No. 2, (Feb. 2006), 65-78;

Morawski, M. (2006b). Optimal Adaptive Routing with Efficient Flapping Prevention, *Proceedings of 4th Polish-German Teletraffic Symposium PGTS*, 85-94, Wrocław, Poland, 20-21 Sep. 2006;

Morawski, M. & Zajączkowski, A.M. (2007). Predictive Control of the Magnetic Levitation System Using a Network Based Controller. Part 1: Control Algorithm, *Proceedings of VIII-th Conference on Control in Power Electronics and Electrical Drives SENE*, 335-340, Łódź, Poland, 21-23 Nov. 2007;

Morawski, M. (2007). Traffic engineering for industrial networks, *Theoretical and Applied Informatics (TAAI)*, Vol. 19, No. 4, (4thQ, 2007), 239-254;

Murthy, C.S.R. & Manoj, B.S. (2004). *Ad-Hoc Wireless Networks*, Prentice Hall Communication;

Natori, K.; Oboe R. & Ohnishi K. (2008) Stability Analysis and Practical Design Procedure of Time Delayed Control Systems With Communication Disturbance Observer, *IEEE Transaction on Industrial Informatics*, Vol. 25, No. 3, (Aug. 2008), 185-197;

PROFIBUS Nutzerorganisation, PROFINET Technology and Application, *PROFIBUS & PROFINET International (PI)*, 2006;

Schoch, E.; Kargl, F. & Weber, F. (2008). Communication Pattern in VANETs, *IEEE Communication Magazine*, Vol. 46 No. 11, (Nov. 2008), 119–125;

Srikant, R. (2003). *The Mathematics of Internet Congestion Control*, Birkhäuser, Boston;

Stallings W. (2002). *High-Speed Networks and Internets: Performance and Quality of Service*, Prentice Hall, 2002;

Wang, F. Y. & Liu, D. (2008). Networked Control Systems. Theory and Applications, Springer, New York;

Welzl, M. (2005). *Network Congestion Control. Managing Internet Traffic*, Wiley & Sons, 2005;

Willig, A.; Matheus, K. & Wolisz A. Wireless Technology in Industrial Networks, *Proceedings of IEEE*, Vol. 93, No. 6, (Jun 2005), 1130-1151;

Yang, Y.; Wang, Y. & Yang, S.-H. (2005). A networked control system with stochastically varying transmission delay and uncertain process parameters., *Proceedings of the 16-th IFAC World Congress*", We-M15_TO/3, Prague, Czech Republic, 4-8 July 2005;

ZigBee Alliance, Inc. (2008), *ZigBee Specification*, No. 053474r13, Nov. 2008;

17

The Role of Business-to-Control Agents in Next Generation Automation Enterprise Systems

Francisco P. Maturana and Eugene Liberman
Advanced Technology Lab
Rockwell Automation
1 Allen Bradley Drive, Mayfield Heights, Ohio 44124

Abstract
Control system level agent technology is a powerful environment that fosters cooperation of control level applications (agents) to solve a set of complex problems not easily solved with a standard control system programming such as ladder code, function blocks, etc. alone. The next logical step in the agent technology evolution is to extend the agent technology capability to the enterprise level. There are many benefits in spanning the agent capability beyond the control system level. The control system environment is not a resource rich environment and some complex large scale applications that require a great deal of computing power may not fit in it. The enterprise level agents can take advantage of the virtually unlimited resources at the enterprise level. This paper discusses the characteristic of intelligent technology and how the agents communicate and cooperate in the different levels of the automation systems.

1. Introduction

Systems in general are becoming more complex and intertwined (Shen et al., 2001) (Mařík et al., 2001). Classical programming and data management will not be able to cope with increased level of complexity. Computing platforms are required to operate at a faster speed to carry out more and more transactions per second. Information integration plays a fundamental role to achieve a more efficient data retrieval and management system. Discovery of information and establishing dependencies among the components of the system is a cumbersome task.
As we progress into the hyper information space realm, automation systems will become more and more involved in the information processing cycles. Automation processes are going to be another node in a global information infrastructure network of resources and, therefore, will need to be more intelligent, highly adaptable, discoverable and information friendly systems.
We envision that in order to fulfill a seamless integration of the distinct levels of the enterprise, we will need to move away from the typical centralized server approach into a

more fine-granular domains with software components acting as independent intelligent agents (Wooldridge and Jennings, 1995)(Brooks, 1986)(Shen *et al 2001*)(Christensen, 1994) (Mařík et al., 2001).

Control system level agent technology is a powerful environment that fosters cooperation of control level applications (agents) to solve a set of complex problems not easily solved with a standard control system programming such as ladder code, function blocks, etc. alone. Rockwell Automation had successfully demonstrated the power and ease of solving complex problems in industrial automation environment using control level agent technology (Maturana et al., 2008)(Gianetti et al., 2006)(Discenzo et al., 2001)(Staron et al., 2004)(Tichý ar al., 2002)(Maturana et al., 2004).

The next logical step in the agent technology evolution is to extend the agent technology capability to the enterprise level. There are many benefits in spanning the agent capability beyond the control system level. The control system environment is not a resource rich environment and some complex large scale applications that require a great deal of computing power may not fit in it. The enterprise level agents can take advantage of the virtually unlimited resources at the enterprise level.

Control systems are generally designed for the "factory floor" environment and integration of the "factory floor" functionality with the rest of the computing environment had always been a challenging task. Therefore, the creation and deployment of the Enterprise level agents that become full members of the control level agent community is a valuable addition to the agent-based control system technology.

Given this additional autonomy at the equipment level, the physical system effectively becomes more survivable and requires less maintenance. The equipment can operate autonomously independent of the rest of the system if a critical event interrupts communications. This property of continuing operation without central control is a must in industrial and military environments.

For example, the Office of Naval Research (ONR) and the US Navy were looking for a highly survivable robust control environment for the chilled water distribution application, a critical ship system (Maturana et al., 2000). One of the major requirements was that the chilled water system continues to operate even after a major disturbance such as an explosion or a missile strike somewhere on the ship occurs. Several approaches were investigated and a distributed intelligent multi-agent system was selected. The main goal was to have a fully distributed system with no single point of failure.

Agents were deployed in the automation controllers. These agents consisted of both reasoning and real-time control and were distributed among 23 controllers which were physically located near the controlled hardware. The reasoning part of the agents inside the controllers negotiated the control actions to accomplish specific missions and configurations.

Figure 1 shows the Navy's land-based water cooling system testbed that we prepared with controller enclosures to download the agents. The testbed included real plumbing, controls and communications, and electrical components that resembled the real ship systems but in a reduced scale. A typical control plan consisted of water routes to transport cold water from the cooling units into the heat loads (computers, radars, weaponry, etc.) and water routes to move hot water from the loads back to the coolers.

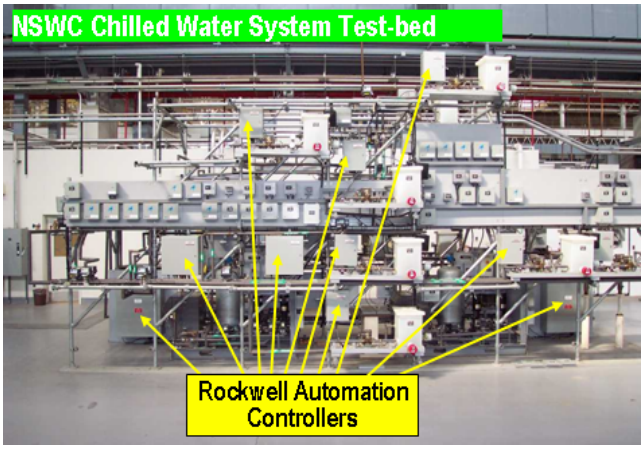


Fig. 1. Office of Naval Research chilled water land-based simulator

The agents evaluated a number of conditions that affected the physical device in order to create a feasible water route. Sometimes lines were obstructed to simulate missing capability, but the agents evaluate adapted their decisions to discover feasible alternatives without following prescribed configurations or pre built tables.
The land based simulator helped in understanding how to program intelligent software in embedded control devices. The agent offered a new dimension in adaptable control systems. However, this capability was limited to device level only. To date we confront a different set of requirements that involve a multi tier system architecture in which we are being asked to combine requirements and capabilities from different layer of the enterprise. Our intention is to explore and demonstrate the architecture of the business-to-control layers and how these will be constructed and interfaced to the different tiers of the manufacturing and information sysems, the industrial environment.

2. Intelligent agent architecture

An important objective of the distributed artificial intelligence and cognition research is to provide a foundation to enhance the capabilities of machines and to make machines more useful and intelligent (Nilsson, 1980)(Balasubramanian et al., 2000)(Brennan at al., 2003)(Charniak and Mc Dermott, 1985). Some of the techniques pursued include a suite of Artificial Intelligence (AI) techniques such as expert systems, fuzzy logic, genetic algorithms, reasoning, artificial neural networks, and model-based techniques. Many of the automation successes reported applied biologically inspired architectures (Brooks, 1986)(Christensen, 1994) and techniques to solve well-targeted automation problems such as adaptive control, defect classification, and job scheduling to name a few.
The capabilities which may be provided by intelligent machines may be categorized based on the degree of embedded knowledge with the most capable systems employing real-time goal adjustment, cooperation, pre-emption, and dynamic re-configuration. These capabilities may be effectively integrated in an agent-based system employing intelligent machines in a distributed automation system. This architecture is built on a foundation of a

society of locally intelligent cooperating machines. It provides an effective framework for an efficient and very robust automation of complex systems.
An agent-based control solution is built around a set of application rules and behaviors. As shown in Figure 2, an agent solution has a tree-like hierarchal shape in which each branch and sub branch can be made of agent components and attributes. The structure of an agent is made from three main components: (1) Reasoning, (2) Control, and (3) Data Table.

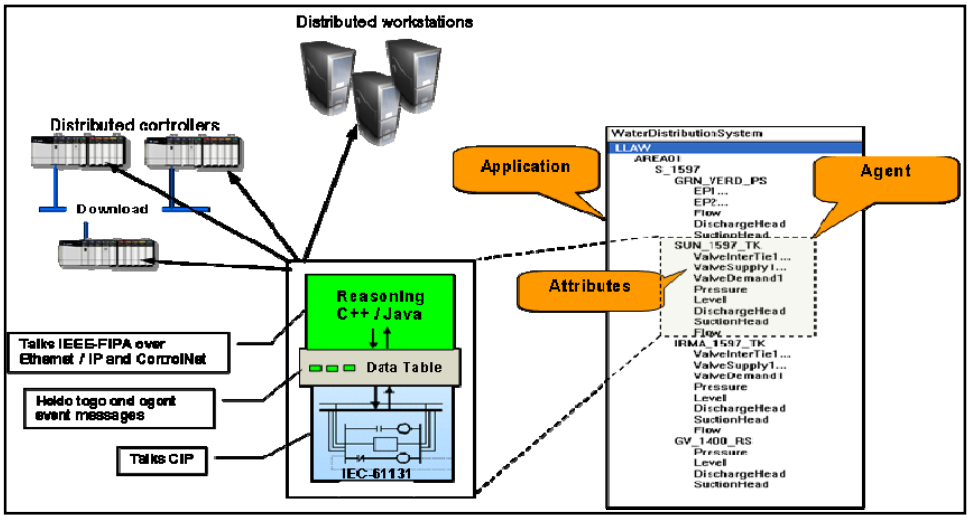


Fig. 2. Agent architecture

The reasoning part is a software component that conveys the agent's heuristics. This component defines the behavior of the agent according to the evolution of its internal rules and the interaction with the control level component. There are event-based transactions between the reasoning and the control level components that are defined during the agent programming phase.
The agent initiates reasoning about a particular event based on the arrival of a global message. A global message is an inter-agent communication that conveys requests or just information. The global message is associated with a particular capability of the agent that is specified as part of the agent behavior. Upon arrival of the global message, the agent behavior for an associated capability is fetched for execution. At this point, the options are diverse since the agent behavior can contain multiple steps that require internal actions as well as the initiation of more global messaging that the agent needs to complete its local goals. A goal is a plan that an agent constructs by pulling together local and combined capabilities. The combined capabilities are obtained via negotiation with other agents. Global messages are encoded according to the Foundation for Intelligent Physical Agents (FIPA, 2000) protocols.
Another way to drive the agent behavior is via the planner engine (see Figure 3). The role of the planner is to coordinate the events from the control level with the agent behavior. Again, there are multiple options on how to do this since the control level events can be associated

with a variety of steps in the reasoning layer. The extent of these associations is a system designer decision.
We use a distributed control architecture based on automation control devices with extended firmware. The extended firmware allows for the realization of component-level intelligence which converts the device into an intelligent node with negotiation capabilities. The intelligence of the application can now be distributed among multiple controllers as opposed to the traditional control system programming in which the concentration of functionality is more predominant.

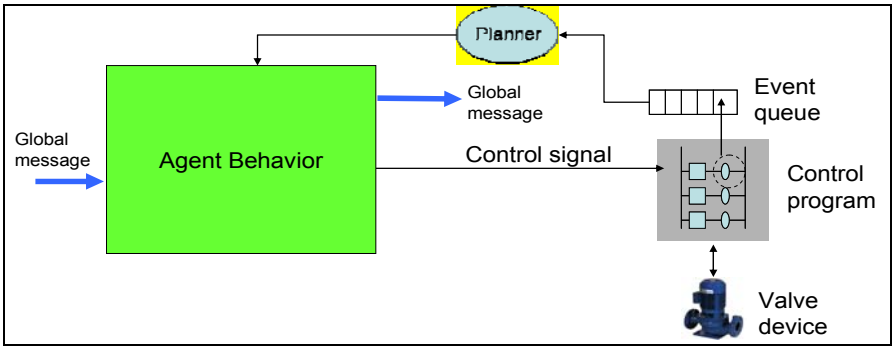


Fig. 3. Agent behavior fetching via the planner

Agents are goal-oriented entities that act autonomously and cooperatively when involved in a problem-solving task. Although an agent is an individualistic entity when pursuing local goals. They organize their individual capabilities around system goals. An agent capability is a description of the type of operations an agent can do. For example, a welding robot agent has a capability to weld specific spots on a structure. On the other hand, a system goal is abstract because there is no explicit declaration of it during the design of the system. A goal can be described as a dynamic social force that pulls agents together to solve a particular task. Thus, a system goal has the following social attributes: (a) System event or need, (b) A first and second level responders, (c) Plan of actions, and (d) Execution.

3. Organizing distributed agents

One of the key benefits of using agents is their ability to work in a distributed environment. Agents use social skills to overcome challenges of the distributed environment.

3.1 Agent design

The effort to program agents may become a difficult task if there are no well defined boundaries between the functions. However, it is always possible to force an initial partitioning to build a working model. The rule of thumb is to generate agents that encapsulate a physical device such as a valve or water pump; these are well defined devices with clear boundaries. But, as the implementation moves into the reasoning layer, it becomes even more difficult to define the boundaries. For example, the designer has to be prepared to decide the pump agent behavior and the context in which pumps negotiate and what they negotiate for.

Later we will duscuss the agent wrapper layer. In such a model, control related functions are kept in the control level and encapsulated with a rather simplistic agent. The higher level behaviors can then be modeled in the upper level as business processes that interact with its control counterpart. From a design point of view the latter approach is very appealing since business level processes will count with greater computing resources than the ones provided by the embedded devices. This brings more freedom into the programming of the functions and helps in deciding on performance issues relative to the agent partitioning.
From the agent technology point of view, industrial applications can be designed according to their decision making complexity and size. The decision making complexity of a complex machine has greater magnitude than a simple machine with a reduced number of actuation points and inputs and outputs connections. The other dimension relates to the size of the application which depends on the number of nodes that are needed to model to operations of the manufacturing plant. Thus, it is possible to categorize the control applications according two these axes, i.e., complexity and size. For example, a material handling system such as a distribution hub is made of a large number of conveyor belts. Each conveyor belt is a simple machine but the material handling operation requires a combination of multiple of these simple machines to carry out the transportation of the material. On the opposite side, we find systems with a reduced number of machines but the machines are very sophisticated in terms of configuration setup, inputs, and outputs. Anywhere in the middle we find a variety of applications that fluctuate between the two poles, as shown in Figure 4.

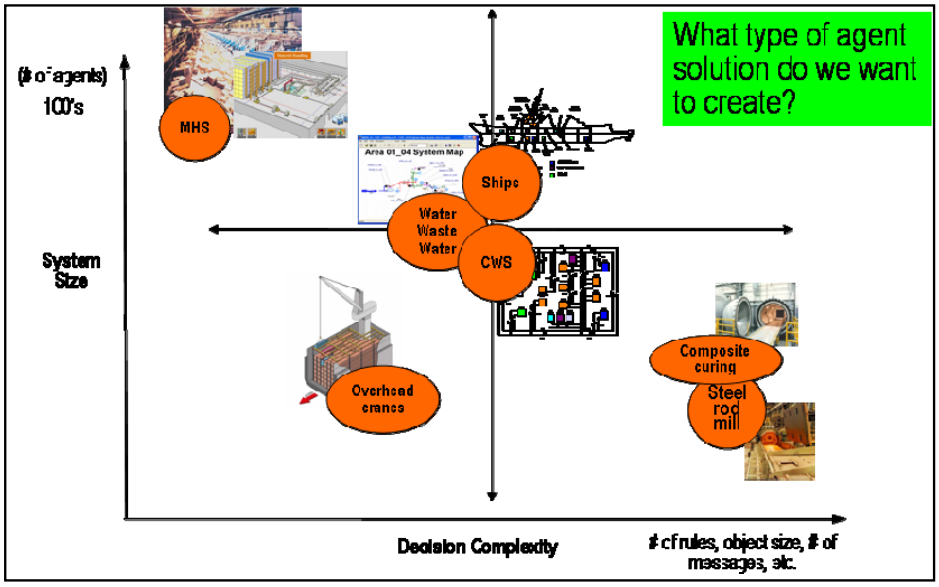


Fig. 4. Application domain categories

Application domain categories are very important in modelling distributed agent control since they frame a better division of functionality. There is a need for establishing rules to design this type of systems. One goal is to highlight one of the most difficult aspects of agent modelling which is the definition of the agent boundaries. Although it is possible to

create centralized, monolithic agents to handle all aspects of the manufacturing organization, it is not a recommended option. We will show that in order to bring greater flexibility and effective scalability into the enterprise, it is required to have a separation of the functions into different layers to make the system more distributed.

4. Enterprise level agent technology architecture

In the interest of simplicity, we will focus the description of our business-to-control architecture as a two layer interaction. In this description, there is an enterprise level and a control level or enterprise domain and control domain.

One of the properties of the control level agents is the capability to find all the peers and establish communication between them. To accomplish agent discovery social knowledge must be composed and stored in directory services. The directory services organize social knowledge using one of the techniques above to propagate agent information throughout the different layers. The social knowledge information can be propagated in an automated fashion or on demand. But, these directory actions take place naturally as part of the directory service functions.

The Enterprise level agents inherit the directory service information from the control level agents. All agents that register their social information with the control level directory services are also known in the enterprise level via social knowledge propagation policies, as shown in the Figure 5. These directory services contain agent properties such as: capabilities, functions, and input and output parameters, etc. This information is required for an application at the Enterprise level to utilize the agent's services in the control level. The LDAP server is fed with this agent information from the Enterprise level DS via an LDAP proxy (LDAP Enterprise agent) at the initialization phase.

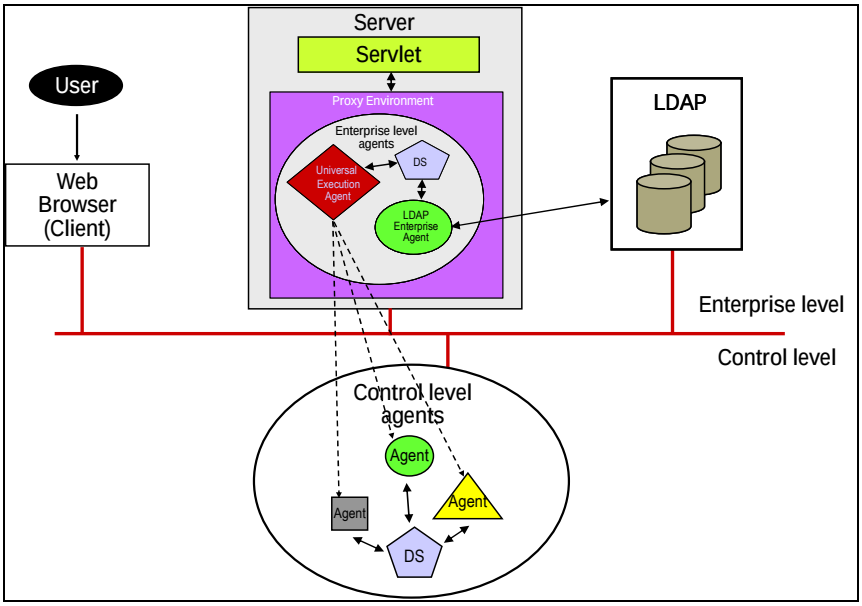


Fig. 5. Directory Service layers

The LDAP directory server was chosen due to its flexibility and accessibility. Any application on the network can access LDAP server, provided that the user or application, an LDAP client, has the proper credentials. LDAP is a standard protocol so every LDAP client adheres to the same standard that is portable and relatively easy to implement.
In this work, we moved the system integration in the direction of a universal model. We are interested in identifying the software components and terminology for the interfaces and communication between the enterprise level and the manufacturing floor. In our business-to-control architecture, we show an initial set of mechanisms that help in connecting the processes without having to be too specific about the information exchange details.
To this end, the agent infrastructure accommodates a proxy environment component that can be dynamically created to bridge the two layers. The proxy environment is a wrapper that knows how to move information across the layers, thereby supporting translation and interpretation of the information.

5. Benefits of the enterprise component to the agent infrastructure

Section 3 and 4 describe the technical requirements and implementation of the enterprise level agents and how the enterprise and control levels can be integrated. This section will show some practical applications of this integrated framework and specific examples will be provided to illustrate the benefits of the interacting layers. We will introduce a water distribution system as an example describing integration of control and enterprise level agents.
In the water distribution system, the control level agents can solve the problem without the "help" of enterprise level agents. But the introduction of the enterprise level agents can increase efficiency and reduce cost of the control level agent system alone. Since control-level agents have limited information about the system, their decision making scope lacks the desired level of optimality. Then, to compensate for the lack of knowledge, the control level agents have been programmed with cooperation protocols to allow them to explore the universe of discourse. Although the cooperative search for solutions may put the agents closer to a near optimum equilibrium, there is still the problem of partial knowledge and localized observation. Thus, the use of enterprise level agents is justified from the point of view of augmented system-level knowledge. Since an enterprise level agent has access to unlimited resources and, services, and databases, it is a much better location to program more exhaustive search nets to support a more global decision making process which can then be coordinated with local level agents.
Furthermore, the enterprise level agents can be launched to report the status of any set of control level agents and all the enterprise level agents can be engaged and controlled from a web browser, so the system status can be remotely obtained at any time from anywhere.

5.1 The BPEL orchestration role

Another benefit of enterprise level agents is the fact that they can be wrapped in web services or other enterprise applications. These web applications can be orchestrated into more complex functions that can run in the background. The orchestrated services then become business level processes that can be coordinated and deployed by a Business Process Execution Language (BPEL) engine (BPEL, 2002). BPEL offers a rich set of features to coordinate business process into an integrated process that oversees the combined

activity for all involve processes, as shown in Figure 7. Concurrency is a natural aspect of the BPEL orchestration permitting processes to execute and communicate in parallel. In our architecture, we can bring the BPEL activity into the agent world as another agent capability since we are able to encapsulate the BPEL process as another agent behavior. BPEL orchestration engines can then be brought into the decision making loops and knowledge exploration in parallel to support the high and low-level agents.
One of these orchestrated applications can be system status monitoring and fault notification. The process status and fault identification can be displayed in a web browser as well. This reduces the requirements on the number of personnel assigned to the role of monitoring the system's status. Monitoring can take place anywhere with Internet connectivity and this has a great potential to reduce costly infrastructure and software development.

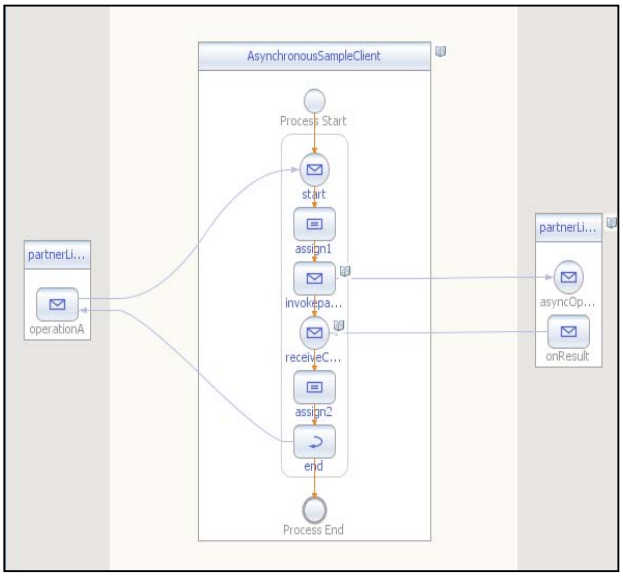


Fig. 7. BPEL orchestration example

The Enterprise level environment has virtually unlimited resources. Several instances of vital redundant agents can be launched and deployed at the enterprise level on various machines. A failure of a machine hosting an agent environment will not impact the system integrity as a whole. The agent system will automatically reconfigure itself and utilize services of available agents. The BPEL orchestration task can be programmed to carry out launching a new instance of an agent if the original instance does not respond or reports failure. The options are unlimited.

6. Sample application: Water distribution and electrical power negotiation

The ability to interconnect the control and the enterprise levels via the business-to-control infrastructure will allow for a consideration of a new breed of control scenarios. We looked

over different aspects of the water distribution domain in which we could demonstrate enterprise and control level agents (Giannetti et al., 2005). We found that in the domestic water distribution systems there is a mix of requirements and decision-making scenarios that map well to the two-level interoperability.

In a domestic water distribution system, there are quality and process requirements that cannot be completely contained in the automation controllers (aka, Programmable Logic Controller—PLC) (CIP, 2001)(IEC, 2001). For example, process-level requirements include water availability, chlorination ratios, residual ratios, etc. Higher level requirements include scheduling of water pumping to accommodate low-cost electricity pricing intervals or seasonal conditions, etc. These requirements shape the design of the agent system in terms of system partitioning and distribution of capabilities. Figure 8 shows the type of agents (green bubbles) that are used in a water distribution system model: pump station and pump, tank, utility company, city water. Each of these agents has the knowledge about how to operate a specific piece of equipment but they also depend on business level knowledge to make high value decisions. In our proposed solution, we include enterprise level services to contain the business level knowledge: utility company and city water. These services provide access to system's historical data (demand, consumption patterns, and electricity pricing policies).

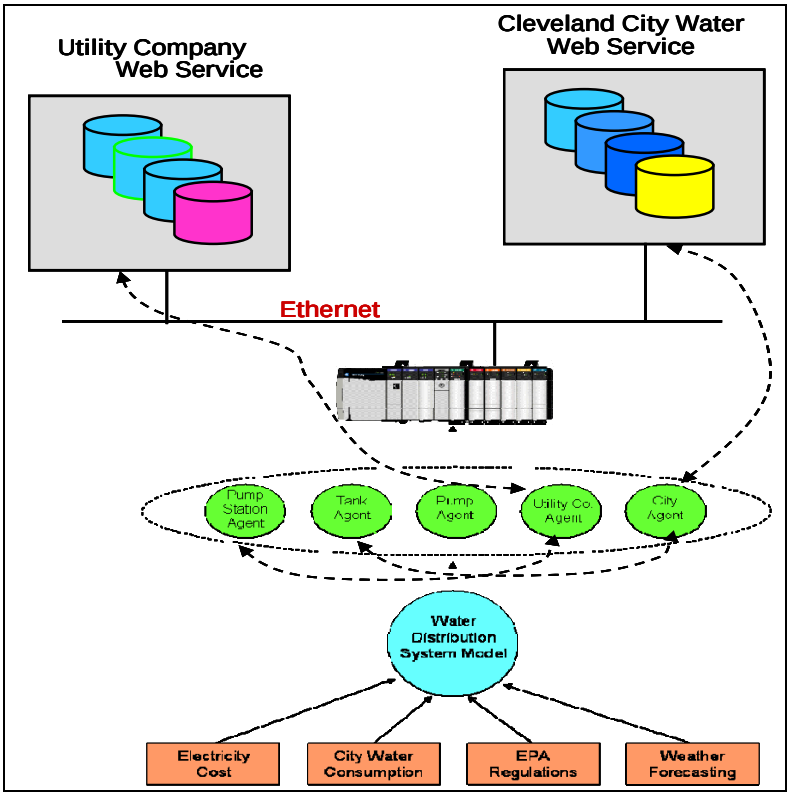


Fig. 8. Agent-based water system

The combined actions of the two levels allows for a complex decision making system. For example, a water tank agent will see the need for receiving additional water (*n*-gallons) to fulfill some near future demand (i.e., forecast demand). The water tank agent in the PLC needed to converse with the city water service to forecast the water demand for a city district in the near future. The city water service has the necessary computing capacity and access to databases to estimate the demand based on the actual, historical, and seasonal consumptions. Here is where the business-to-control engagement adds its value.

Another scenario also takes place between the water system and the utility company services, as shown in Figure 9. After the water tank agents calculate their future demand, they emit a request for pumping water to the pumping station agents. The pumping station agents need to carry out process level calculations to estimate the amount of power that is going to be needed to provide the water. This process-level evaluation takes place in between the pumping station and the pumps since this information gathering requires health assessment information that is known by the pumping devices themselves.

The pump stations then contacts the utility company services with a request for low cost electricity. The utility company services have capacity to do the calculations by contacting the pricing interval calculators and databases and perhaps humans to estimate a final price for the electricity and a valid time interval for the offering. The utility company service needs to interact with seasonal and historical databases to estimate the prices. This example describes a complex interaction among services and agents. In a classical framework without the considerations that have showed in this article, programming the interactions would be expensive and cumbersome.

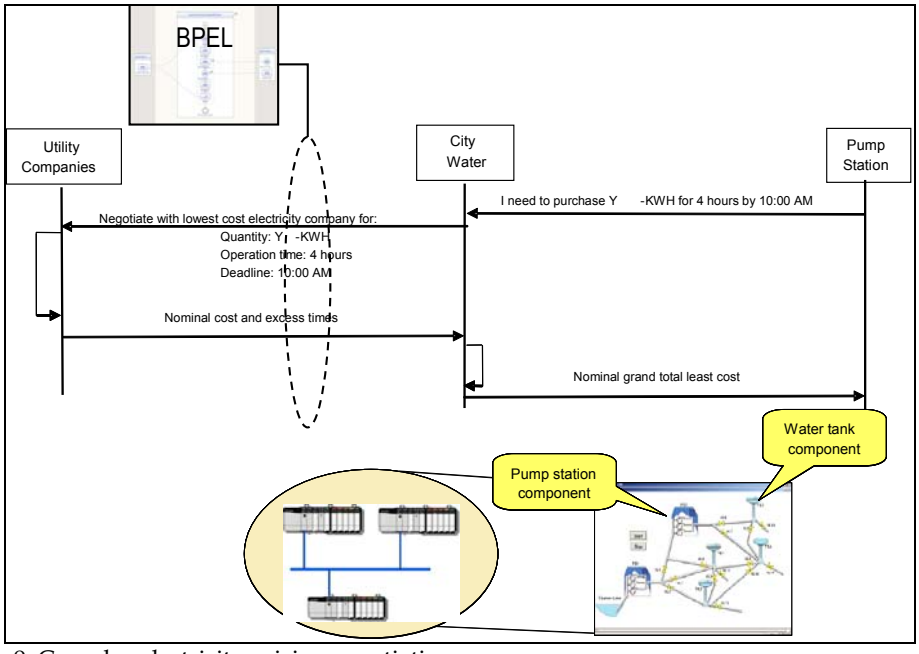


Fig. 9. Complex electricity pricing negotiation

The description above illustrates two representative scenarios that may occur in an agent-based water distribution system between the enterprise and the control levels. The role of the BPEL orchestration is very fundamental in the coordination of the different services and the selection of their responses. There will be multiple transactions going back and forth in multiple directions that need to be coordinated and synchronized in order to maintain the stability of the system. For example, the BPEL process that orchestrates the interaction between the city water agent and the utility companies needs to handle one-to-many transactions in one direction (city water to utility companies) and many-to-one transactions in the opposite direction. In the transition between communications, BPEL needs to listen for the responses while applying system-level rules to decide on the most suitable responses to be emitted to the agents. The scenarios can become more sophisticated and complicated. But the intention of this work is to show how to assemble the architecture for realizing the future vision of autonomous control systems.

7. Conclusions

The integration of the enterprise level agents and control level agents will make systems more robust and operate at lower cost. However, the right balance needs to be maintained between the control and the enterprise functionalities. Systems designers will have to make sure the loss of enterprise capability will not compromise the fundamental control level ability to carry out control tasks autonomously.

8. References

M. Wooldridge and N. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Eng. Rev., vol. 10, no. 2, 1995, pp. 115–152.

A. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE J. Robotics and Automation, vol. 2, no. 1, 1986, pp. 14–23.

Shen W., Norrie D., and Barthès J.P.: "*Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*". Taylor & Francis, London, 2001.

J. H. Christensen, "Holonic Manufacturing Systems: Initial architecture and standards direction," in *Proc. First European Conference on Holonic Manufacturing Systems*, Hanover, Germany, 1994, pp. 20.

Mařík, V., Pěchouček, M., and Štěpánková, O., Social Knowledge in Multi-Agent Systems. *Multi-Agent Systems and Applications*, LNAI 2086, Springer, Berlin, 2001, 211-245.

Francisco P. Maturana, Dan L. Carnahan, Donald D. Theroux, Kenwood H. Hall: Distributed multi sensor agent for composite curing control. ETFA 2008: 1236-1243.

Lucilla Giannetti, Francisco P. Maturana, Frederick M. Discenzo: Agent-Based Control of a Municipal Water System. CEEMAS 2005: 500-510.

Discenzo, F.M., Marik, V., Maturana, F.P., Loparo, K.A., 2001, *Intelligent Devices Enable Enhanced Modeling and Control of Complex Real-Time Systems*, International Conference on Complex Systems, (Irvine).

Staron, R. J., Maturana, F. P., Tichý, P., and Šlechta, P., Use of an Agent Type Library for the Design and Implementation of Highly Flexible Control Systems. The *8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004)*, Orlando, USA, 2004, 18-21.

P. Tichý, P. Šlechta, F. P. Maturana, and S. Balasubramanian, "Industrial MAS for Planning and Control," in (Mařík V., Štěpánková O., Krautwurmová H., Luck M., eds.) *Proc Multi-Agent Systems and Applications II: 9th ECCAI-ACAI/EASSS 2001*, AEMAS 2001, HoloMAS 2001, LNAI 2322, Springer-Verlag, Berlin, 2002, pp. 280-295.

F. Maturana, P. Tichý, P. Šlechta, F. Discenzo, R. Staron, and K. Hall, "Distributed multi-agent architecture for automation systems", Expert Systems with Applications 26, 2004, pp. 49-56.

Maturana F., Balasubramanian S., and Vasko D.: An Autonomous Cooperative Systems for Material handling Applications. ECAI 2000, Berlin, Germany, 2000.

Francisco P. Maturana, Raymond J. Staron, Kenwood Hall: Real time collaborative intelligent solutions. SMC (2) 2004: 1895-1902.

Charniak, E., & McDermott, D, 1985, *Introduction to Artificial Intelligence*, Addison-Wesley.

Nilsson, N., 1980, *Principles of Artificial Intelligence*, Morgan Kaufmann (Los Altos).

S. Balasubramanian, R. W. Brennan, D. H. Norrie, "Requirements for Holonic Manufacturing Systems Control," in *Proc DEXA Workshop 2000*, 2000, pp. 214-218.

R. W. Brennan, K. H. Hall, V. Mařík, F. P. Maturana, and D.H. Norrie, "A real-time interface for holonic control devices," in: V. Mařík, D. McFarlane, P. Valckenaers (Eds.): *Holonic and Multi-agent Systems for Manufacturing, Lecture Notes in Artificial Intelligence*, No. 2744, Springer-Verlag, Berlin, 2003, pp. 25-34.

FIPA, The Foundation for Intelligent Physical Agents (FIPA), 2000: http://www.fipa.org.

CIP: Common Industrial Protocol, Available, 2001: http://www.ab.com/networks/cip_pop.html.

IEC (International Electrotechnical Commission), TC65/WG6, 61131-3, 2nd Ed., Programmable Controllers - Programming Languages, April 16, 2001.

Business Process Execution Language for Web Services Version 1.1., July 2002, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf.

18

A Multiagent Architecture Based in aFoundation Fieldbus Network Function Blocks

Vinicius Ponte Machado
Federal University of Piaui
Natal – RN – Brazil
Dennis Brandão
University of São Paulo
São Carlos – SP – Brazil
Adrião Duarte Dória Neto
Federal University of Rio Grande do Norte
Natal – RN – Brazil
Jorge Dantas de Melo
Federal University of Rio Grande do Norte
Natal – RN – Brazil

1. Introduction

The industrial automation is directly related to the technological development of information. Better hardware solutions, as well as improvements in software development methodologies have made possible the rapid development of the productive process control. In this Chapter, it is proposed an architecture that permits to join two technologies in the same hardware (Industrial Network) and software context (Multiagent Systems – MAS). We show a multiagent architecture which uses an algorithm-based Artificial Neural Network (ANN) to learn about fault problem patterns, detect faults, and adapt algorithms that can be used in these fault situations. We also present a dynamic Function Block (FB) parameter exchange strategy which allows agent allocation in fieldbus. This proposed architecture reduces the supervisor intervention to select and implement an appropriate structure of function block algorithms. Furthermore, these algorithms, when implemented into device function blocks, provide a solution at fieldbus level, reducing data traffic between gateway and device, and speeding up the process of dealing with the problem. We also present some examples for our approach. The first one introduces FBSIMU which simulates Foundation Fieldbus function blocks architecture. This software has a controlled process and allocates the MAS to detect and correct faults. The second example shows a multiagent architecture that implements the neural network change in a laboratory test process which imitates fault scenarios.

2. Theoretical Foundation

2.1 Foundation Fieldbus Protocol

The term FOUNDATION Fieldbus (FF) indicates the protocol specified by the Fieldbus Foundation and standardized by IEC[1] standards number 61158 (IEC, 2000) and 61784 at profile CPF[2] - 1/1 (IEC, 2003). It is a digital, serial, bidirectional, and distributed protocol, which interconnects field devices such as sensors, actuators and controllers. Basically, this protocol can be classified as a LAN (Local Area Network) for instruments used in process and industrial automation, with the ability to distribute the control application through a network.

This protocol is based on the ISO/OSI (International Organization for Standardization/Open System Interconnection) seven layer reference model (ISO, 1994). Although being based on the ISO/OSI model, the FF does not use the network layer, the transport layer, the section layer, nor the presentation layer, because it is restricted to local applications. The entire network structure of the FF concentrates on the physical layer, the data link layer (DLL) and the application layer. Besides these three implemented layers, the protocol defines an additional layer named User Application Layer.

The FF Physical Layer, named H1, uses a shielded twisted pair cable as communication medium. The H1 specifies a 31.25 kBit/s bit rate with Manchester codification over a bus powered channel. The network topology configuration is flexible: it is typically configured with a trunk and several spurs, attending certain physical and electrical limitations regarding maximum spur lengths and number of transmitters.

The DLL carries the transmission control of all messages on the fieldbus and its protocol grants to the FF network temporal determinism for critic process control data. The communication is based on a master–slave model with a central communication scheduler (master), named Link Active Scheduler or LAS. This node performs the medium access control (MAC). Two types of DLL layer are standardized: Basic and Link Master. A Basic DLL transmitter does not have LAS capabilities, it operates passively as a communication slave. A Link Master DLL transmitter, on the other hand, can execute LAS functions and thus, if the active LAS node fails, become the LAS node. The FF Data Link Layer supports two transmission policies: one addressed to scheduled cyclic data, and another to sporadic (unscheduled) background data. These two communication policies share the physical bus, but they are sequentially segmented in cyclic time slots or periods. In the scheduled communication period, most process variables generated by periodic processes are transmitted cyclically according to a static global schedule table loaded on the LAS node. This cyclic transmission mode has higher priority over acyclic transmission modes. A periodic process can be defined as a process initiated at predetermined points in time, also called a time-triggered process.

The period for the network cyclic process is typically from tenths to hundreds of milliseconds, and it is mandatory to consider that the generated data must be delivered before the next data is available. This type of periodic data is usually related to measurement and control variables (Cavalieri et al., 1993).

---

[1] International Electrotechnical Commission

[2] Communication *Profile* Family

Sporadic or unscheduled communication is used to transmit non periodic, or aperiodic, data, generated by sporadic processes not directly related to the control loop cycles, but to user configuration actions and data supervision efforts. The unscheduled transmissions are dispatched under a token pass scheme. A token that circulates among all active nodes on the bus is used in FF protocol.

Once a transmitter receives the token, it is granted the right to send pending aperiodic messages with a minimum priority for a specific time period. Non periodic (or event-triggered) processes are initiated as soon as specific events are noted (Pop et al., 2002). The event-triggered processes are unpredictable and usually related to alarm notifications, configuration data and user commands as cited before. Although acyclic traffic is less frequent than the cyclic one, the acyclic data should also be delivered prior to a given deadline, according to the system requirements. For a description of the MAC operation on both cyclic and acyclic phases, refer to Hong & Ko (2001), Wang et al. (2002), Petalidis & Gill (1998).

The FF User Layer is directly related to the process automation tasks, and it is based on distributed control or monitoring strategies composed of Function Blocks (FB). Function Blocks are User Layer elements that encapsulate basic automation functions and consequently make the configuration of a distributed industrial application modular and simplified (Chen et al., 2002). Distributed among the transmitters, the FBs have their inputs and outputs linked to other blocks in order to perform distributed closed control loop schemes. When blocks from different transmitters are linked together, a remote link is configured and mapped to a cyclic message. Considering that all cyclic messages should be released in a predetermined instant defined on a schedule table, and that they carry data generated by the FBs, it is adequate to synchronize the execution of the FB set on the system with the referred cyclic transmissions schedule table. This solution leads to the concept of joint scheduling (Ferreiro et al., 1997).

The Foundation Fieldbus standardized a set of ten basic function blocks (Fieldbus Foundation, 1999a), a complementary set of eleven advanced control blocks (Fieldbus Foundation, 1999b), and a special flexible function block intended to be fully configurable by the user, i.e., internal ladder logic and parameter set (Fieldbus Foundation, 1999c). The standard and advanced block sets provide mathematical and engineering calculations necessary to configure typical industrial control loop strategies, while the flexible function block can be applied to custom or advanced controls or to complex interlocking logics based on ladder nets. It is important to mention, however, that the standard is open at this point, permitting the integration of ''user-defined'' custom function blocks in order to enhance the capabilities of FF control system, and make the integration of novel control techniques possible.

2.2 Multiagent systems

An agent is a computer system, a paradigm to the development of software applications that is situated in some environment and is capable of autonomous action in that environment in order to meet its design objectives (Russell & Norvig, 2003). In a few words, a multiagent system (MAS) is a problem of placing the agents together (organized as a society). The application components of a multiagent system are agents. Several different multiagent architectures can be found in the literature including applications in automation (Weyns et al., 2005; Weyns & Holvoet, 2007; Seilonen et al., 2002; Feng et al., 2007). In a MAS, control is

decentralized, i.e., none of the system components has global control over the system or global knowledge about the distributed system.
The main reason for the use of agents in these environments is that these applications need distributed interpretation and distributed planning by means of intelligent sensors.
Furthermore, distributed multiagent systems are an appropriate concept for many fields of industrial automation like monitoring, fault diagnosis, simulation and control, as they give several advantages for these applications. They allow distributed data collection while maintaining a high level of scalability and flexibility, once they keep network load low through an adequate pre-processing. They also provide on-site reactivity and intelligence that is required in various remote control scenarios, since the network channel is not capable of transporting each and every control command. Finally they offer an abstraction level when accessing proprietary devices for monitoring and control, and they are often easier to integrate into existing applications than, for example, a service oriented architecture (Theiss et al., 2008).
Applications of agent technology in the research of process automation systems have not been as numerous as many other industrial application domains. Neither the way to apply agent technology in process automation nor the possible utility of it has been so evident in process automation than in other fields. However, some promising research has been reported and some experiences from other fields might also have applicability in process automation (Seilonen et al., 2002).
Autonomy, high encapsulation and reactivity of agents motivate their usage in large automation systems. The application area of multiagent systems includes power supply systems, manufacturing systems, building automation and mobile applications (Jennings & Bussmann, 2002). The agent functionality comprises monitoring and diagnosis (e.g., Taylor & Sayda, 2003; Albert et al., 2003; Pirttioja et al., 2005), control, scheduling, modelling and simulation of these applications. The agents primarily operate on management level (Schoop et al., 2002) and use web-based technologies like web services and OSGi (Fei-Yue et al., 2005). This allows them to use the PCs and servers as hosts, making the performance, memory usage and real-time issues negligible.
As a conclusion to the current research about agent applications in process automation and other control applications, one could state that agents have generally been applied either for higher-level, non real-time and event-based operations, or for integration purposes. The state-of-the-art research regarding MAS applications in process automation leaves some questions unanswered behind. Research has mainly focused on control functions and the functional role of MAS in process automation. Other functions, e.g., monitoring and information access, have received less attention (Seilonen, 2006). Furthermore, in many cases these models do not address the issue of deterministic response times. Unlike the aforementioned studies, we have shown MAS architecture which enables the implementation of control configuration at the fieldbus level. We believe this is the main contribution of our work. A similar study was proposed by Brennan et al. (2002). However, in our study we aggregate machine learning through an Artificial Neural Network (ANN) and FIPA (Foundations of Intelligent Physical Agents) compliant agents. Moreover, our implementation raises a basic agent feature: adaptation. The function block allocation will change to adapt to a type of problem, without user intervention.

3. Function Block Intelligent Algorithms

Smart configuration strategies are implemented by intelligent algorithms that are incorporated to the sensors using the standard function blocks. These blocks have basic functions that, when combined, are able to implement the artificial neural network for example. The organization of function blocks is essential to the success of this type of process.

The protocol found to better suit these demands was the Foundation Fieldbus protocol, because its system is gifted with the capacity of distributing the control of the process in the field, i.e. the sensors and actuators have embedded processors which can execute the algorithms in a distributed way.

Many projects have been developed using Foundation Fieldbus protocol and function blocks. In ours laboratories (LAMP - Petroleum Measurement and Evaluation Laboratory in Federal University of Rio Grande do Norte) some intelligent algorithms were implemented using mainly neural networks.

In Silva et al. (2006), we can see a solution to execute artificial neural network algorithms in the environment of networks to Foundation Fieldbus industrial automation, based on standardized function blocks. This strategy involves two function blocks: arithmetic and characterizer. They must be configured and linked in such a way that the set behaves as an artificial neuron (Haykin, 1999). In the arithmetic block, the Algorithm Type parameter must be chosen as Traditional Adder and the gains of the inputs must be filled according to the training performed, as well as the bias values.

By linking the output of an arithmetic function block to the input of a signal characterizer function block, configured as described above, we have an artificial neuron in the FF environment. And by linking of these neurons, neural networks are built.

With another neural network function block, configuration of the agents can compensate the error as it is seen in Cagni et al. (2005). In his work the implementation of the self-calibration, self-compensation and self-validation algorithms for Foundation Fieldbus sensors are presented using standard function blocks.

The deterioration of the sensors can make the sensor measurement precision decrease in time, until another calibration of the sensor is made. The lack of precision and the calibration process can be economically disadvantageous to the industries. Determining what is the best calibration period for a sensor is the main focus of interest of this research. With this purpose, some based-neural network algorithms were created to increase the precision and the reliability of data collected by the sensors and to optimize the calibration periods. These algorithms are the self-compensation, self-calibration and self-validation ones.

The addition of noise is another very common problem during the process of extracting information generated by a sensor installed in a field network. In Costa et al. (2003), the implementation of a system is proposed so that, beginning from software embedded in a DSP (Digital Signal Processor), interacts with fieldbus devices connected through a Foundation Fieldbus network. This approach, based on the technique of Independent Component Analysis (ICA), presents an efficient solution to the problem of extraction of the noise derived from the sensor. In other work, Fernandes et al. (2007) presents an approach to process fault detection and isolation (FDI) system applied to a level control system connected with an industrial network Foundation Fieldbus. The FDI system was developed using artificial neural networks (ANN). Basically, the FDI system was divided in two parts:

the first corresponds to neural identification of the plant model; and the second, to the detection and isolation of faults in process.

4. Foundation Fieldbus Simulated Environment

The basic concept of the FBSIMU (Foundation Fieldbus Simulated Environment) architecture is to map each Function Block, as well as the plant, in an independent LabVIEW[3] application, also named Virtual Instrument (VI). The configuration of the whole system is centralized in the FBSIMU.CONF module. This module's graphical user interface is inspired by commercial fieldbus configuration tools. As mentioned before, the FBSIMU is focused on the function block application layer and it is composed exclusively of software according to a modular and extensible architecture. The simulator was developed in LabVIEW using the G graphical programming language, ''native'' language in this environment. Each FBSIMU module or software unit simulates an element or a structure of a real FOUNDATION Fieldbus system (Brandão, 2005).

4.1 Function Block simulation

The Function Block modules are programmed into the FBSIMU according to the FF specifications directions and, consequently, the usage and configuration of a simulated control loop on the FBSIMU environment is identical to a real FF system. A ''LabVIEW Foundation Fieldbus Tool Kit'' library has been developed (Pinotti et al., 2005) to provide a range of typical Foundation Fieldbus control and acquisition functions, according to the standards. These functions encapsulate different FF calculations and data type manipulations necessary to build standard or custom Function Blocks. A Function Block seed module is also used to accelerate the process of developing and integrating new projects. The seed has the whole FB module structure (an empty structure) and directions to proceed with a FB project from the design to the final test procedures.

Each FB module is built in two different versions that share the same FB core: stand-alone and process. The stand-alone FBs are executed by user commands and controlled by its graphical user interface. Its execution can be performed independently of any other module, so the user is able to test the FB and simulate its operation under a controlled condition of inputs and outputs. The graphical user interface is intuitive and enables the user to execute the FB continually or in a step-by-step mode. The process version of a FB, on the other hand, is controlled remotely likewise real FBs. Each process FB has a unique identification and its operation is controlled by the user through the following commands:

- FB_Read: this service allows the value associated with a block parameter to be read.
- FB_Write: this confirmed service allows the value associated with a block parameter to be written.
- FB_Exec: this service triggers the block algorithm to be executed.

[3] LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench), a platform and development environment for a visual programming language (National Instruments)

- FB_Reset: this service allows default values associated with all block parameters to be written.

Process FBs do not have graphical user interface, they are instantiated by the FBSIMU.CONF in each simulation process. The communications between process FBs and the FBSIMU.CONF are performed programmatically and dynamically by the LabVIEW function ''Call by Reference Node''. It is important to note that the industrial transmitters are not considered in the FBSIMU architecture, i.e., function blocks are instantiated on the simulation without being allocated in specific ''virtual'' transmitters. The FBSIMU.CONF module graphical user interface for fieldbus configuration is shown in Figure 1.

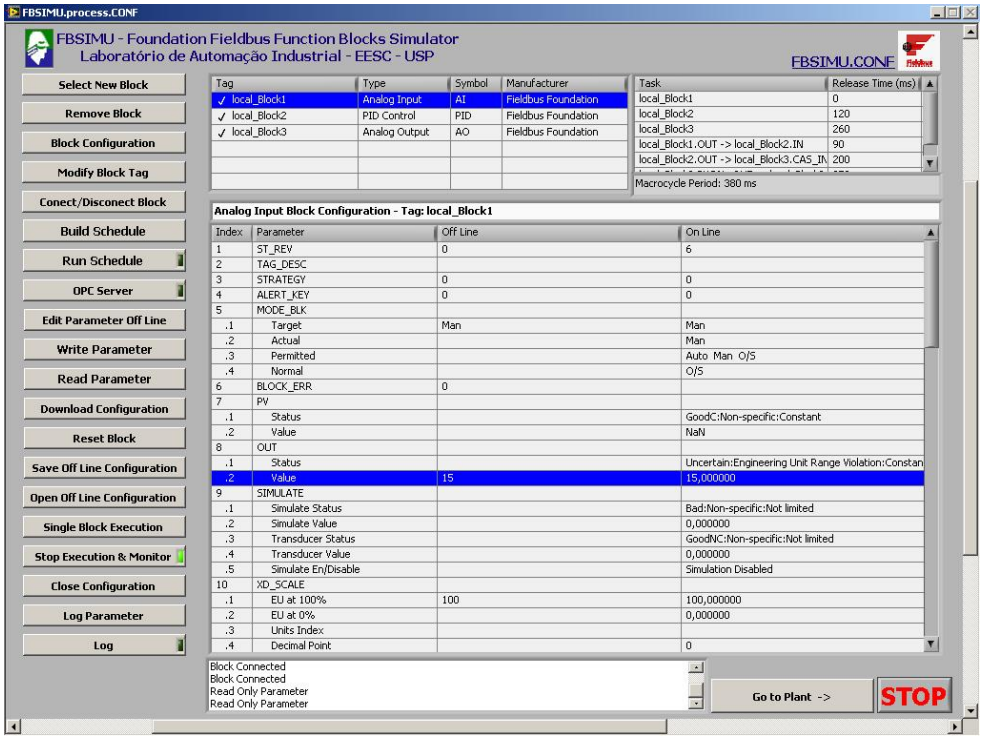


Fig. 1. FBSIMU.CONF graphical user interface for fieldbus configuration

4.2 Physical plant simulations

The plant module cyclically executes a discrete single variable (SISO) linear ARX (Auto-Regressive with Exogenous Inputs) mathematical structure (Ljung, 1999). This module is configured on the FBSIMU.CONF and simulates the controlled plant. The adopted ARX structure is represented by Equation 1, where k is the discrete time instant, Y is the output vector, U is the input vector, i is the number of MIMO plant inputs and outputs, na is the number of output regressors, and nb is the number of input regressors. In the current version, i is set to 1 (one) to reflect a SISO model.

The simulated plant dynamic behavior is modeled on the dynamic matrixes A and B. It must be observed that the number of regressors limits the model dynamic order and that all regressors must be initialized prior to starting the simulation.

$$Y_{ix1}(k) = \sum_{s=1}^{na} As_{ixi} \otimes Y_{ix1}(k-s) + \sum_{s=1}^{nb} Bs_{ixi} \otimes U_{ix1}(k-s) \quad (1)$$

As the user chooses the plant order (1st, 2nd or 3rd) and dynamics (gain for 1st and 2nd order systems, damping ratio, natural frequency and time constant), the selected plants' Bode Magnitude Chart, Pole-Zero Map, Root Locus Graph and the Step Response are instantly presented on the graphical user interface.
A white noise generator function adds a simulated acquisition noise to each plant output bounded by user configurable amplitude. Figure 2 shows the FBSIMU.CONF module graphical user interface for plant configuration.

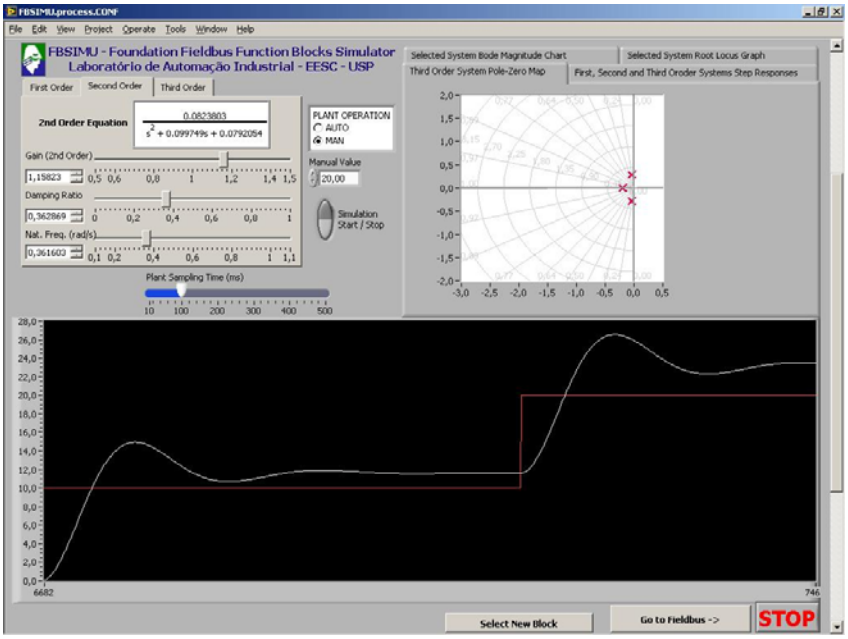


Fig. 2. FBSIMU.CONF graphical user interface for plant configuration

4.3 Simulation Architecture

The proposed execution model for the fieldbus simulation on FBSIMU is considered hybrid, because some tasks are event-driven while other tasks are time-triggered, according to table 1. All tasks related to the user interface are event-driven, they are executed after a user action such as selecting a new block, configuring schedule table, saving a configuration or starting the execution.

On the other hand, the tasks related to executing FBs according to a schedule table, plant simulation, and online monitoring of FBs are time-triggered.

Module	Priority	Execution	Timeout	Determinism
GUI & User commands	Low-Low	Event driven	1 sec.	No
FB Schedule	High-High	Time triggered according to the schedule table	No	Yes
Plant Execution	High	Periodic with configurable period	No	Yes
Online FB Parameters Monitoring	Low	Periodic with period = 500ms	No	Yes

Table 1. FBSIMU task set

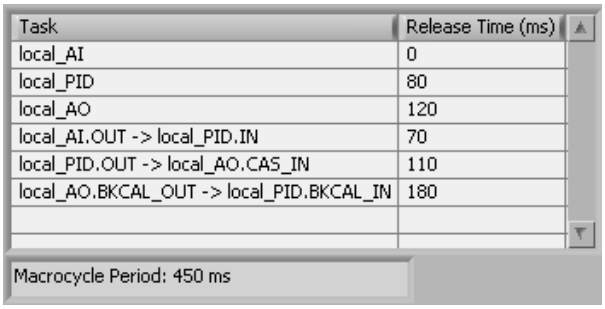
Once all tasks are performed on a single microprocessor they are, naturally, concurrent. The proposed solution for preventing unexpected delays of time-triggered tasks (considered critical) due to executing event-driven tasks (considered non-critical) is adopting priority levels for each task and preemptive execution mode.
In the preemptive execution mode, a higher priority task that is ready to execute preempts all lower priority tasks, which are also ready to execute or actually during execution. Table 1 summarizes the FBSIMU task set and its timing and execution characteristics.

4.4 A typical simulation experiment

The intrinsic flexibility of simulation tools opens a wide range of FBSIMU experiments where users can exploit the effect of important communication parameters and configurations found in industrial FF systems. Practical experiments consist in comparing given simulated fieldbus system performances over different operation conditions. The results can be analyzed via log files or graphically on charts.
For typical simulation sessions, a specific FB control strategy should be defined. Then, the period of the macrocycle must be set in milliseconds, and all the release times (in milliseconds) of each FB execution and FB link must be defined regarding the macrocycle start instant. Figures 3 and 4 present an example of these configurations on the FBSIMU.

Tag	Type	Symbol	Manufacturer
✓ local_AI	Analog Input	AI	Fieldbus Foundation
✓ local_PID	PID Control	PID	Fieldbus Foundation
✓ local_AO	Analog Output	AO	Fieldbus Foundation

Fig. 3. FBSIMU block list



Task	Release Time (ms)
local_AI	0
local_PID	80
local_AO	120
local_AI.OUT -> local_PID.IN	70
local_PID.OUT -> local_AO.CAS_IN	110
local_AO.BKCAL_OUT -> local_PID.BKCAL_IN	180

Macrocycle Period: 450 ms

Fig. 4. – FBSIMU schedule table

The configuration parameters from all FBs on the schedule table must be set in a parameter table, as shown in Figure 5, to support the proposed strategy (for example, Block Mode, Scaling, Gains), exactly likewise a real block strategy configuration.

Index	Parameter	Off Line
1	ST_REV	0
2	TAG_DESC	AI_TP_1
3	STRATEGY	0
4	ALERT_KEY	0
5	MODE_BLK	
.1	Target	Auto
.2	Actual	
.3	Permitted	
.4	Normal	
6	BLOCK_ERR	0
7	PV	
.1	Status	
.2	Value	

Fig. 5. FB parameter table

The last step is to link the input and output parameters from the AI and AO blocks to the plant simulation module as represented in Figure 6. The connection between an Analog Output block (AO) and the plant input (manipulated variable - MV) and the connection between the plant output (primary value - PV) and the Analog Input block (AI) are configured by the user for close loop experiments. Alternatively, the user may connect only the plant PV to the AI block for an open loop simulation, or manually load the plant PV with a given numeric value.

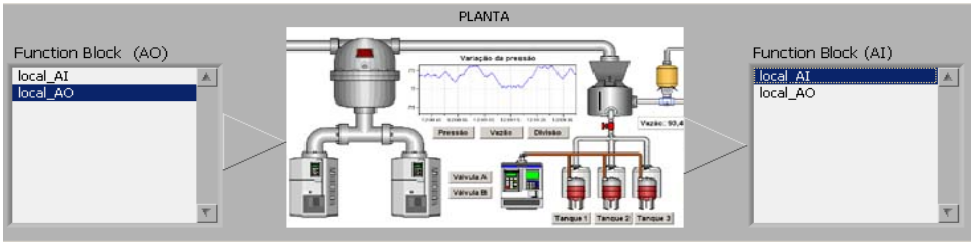


Fig. 6. - Plant simulation module

Finally, the user downloads the configuration to each FB and starts executing the schedule. During the execution, it is possible to monitor the parameter table with online parameter values and register the parameters on text files for further analysis. With the FBSIMU architecture, the FF operation scenarios can be configurable and different sequences of practice training can be defined to embrace fundamental concepts of fieldbus control systems, as well as practical situations of alarms or events handling. This characteristic is considered important because most of the traditional pilot plants equipped with fieldbus instrumentation offer just one or a few scenarios where a full sequence of practice experiments should be based on. Thus, the use of a simulated fieldbus system enables a flexible evaluation of the contribution and effect of the communication protocol on the overall system dynamics, which is an impossible goal considering that, in pilot plants equipped with real instrumentation, most communication configurations are fixed and, in most cases, inaccessible to end-users.

5. Multiagent Architecture

There is a number of requirements that ensures control of the production process. These include process variables, that is, data collected by the sensors that are often used for actuator actions. An incorrect interpretation and analysis of current data can result in the malfunctioning of the productive process. Thus, the functioning of a process can occur by combining these two items: collected data (sensors) and actions (actuators). Accordingly, we propose multiagent architecture which enables the analysis, interpretation and correction of data and events occurring in the fieldbus to improve production processes at the fieldbus level.

This architecture is composed of a multiagent system that generates inspection routines of the collected data in the plant's sensors. The aim is to induce the agents to evaluate field device data and investigate inconsistencies that may impede the productive process, such as lack of precision, external noise, alarm interpretation etc. During the process, the agents start functioning and perform tasks such as analyzing and correcting events through intelligent algorithm, as shown in Section 3.

Figure 7 shows that our architecture is composed of observation, diagnostic, and execution agents. The Observation Agent (OA) is responsible for monitoring field devices and checking for inconsistencies and faults. The Diagnostic Agent (DA) attempts to identify the type of fault (detected by OA) occurring in the field devices. Once the problem is diagnosed, the Execution Agent (EA) tries to correct it by implementing an intelligent algorithm. Another component in this architecture is a layer (LABVIEW/FIPA Layer) that allows

agents to interact with the Foundation Fieldbus model. We use LABVIEW framework to develop based-agents FIPA (Polaków & Metzger, 2007). The main reason for using FIPA-compliant agents is their capacity to aggregate other FIPA agents in the architecture. We also use an OPC (OLE for Process Control) interface, such as that used by Seilonen et al. (2002b) to integrate agents with the fieldbus. In our study, we change the function block connections to perform a desired control algorithm and to make these function block interconnections act as agents.

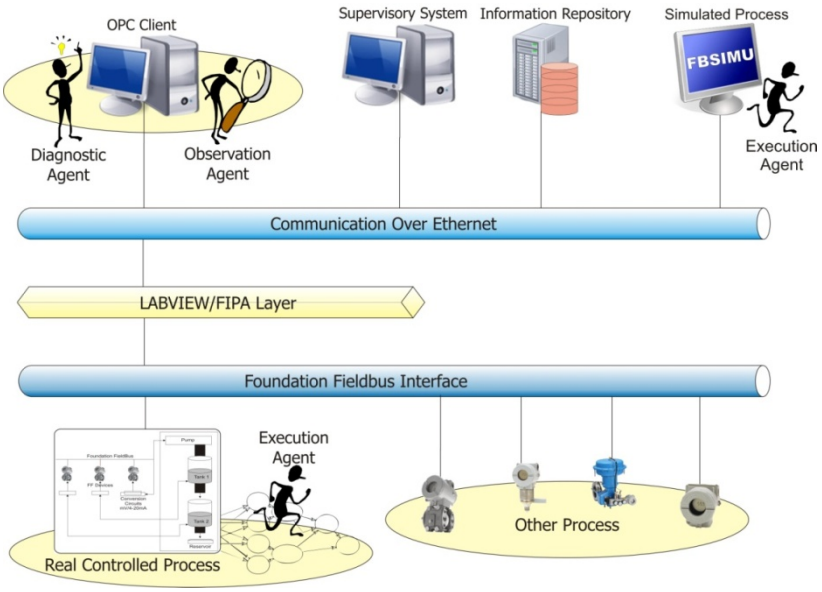


Fig. 7. – Proposed Multiagent architecture environment

The LABVIEW-FIPA layer allows communication with field devices through the OPC. Agents can access field devices and allocate function blocks. Agent learning occurs at higher levels (supervisory) which communicate with field devices (through the LABVIEW-FIPA/OPC layer). The gateway is responsible (physically) for establishing this communication. What is learned by the agent is stored in the Information Repository. This information is useful to other agents and reused in similar situations. Learning is implemented in computer supervisory machines as part of the agent. The agent action is performed at fieldbus levels, i.e., device function blocks (FB). The FB interconnections form an algorithm which controls a process. In our agent, this algorithm (ANN) is used in some instances to monitor devices (OA), and in others to correct faults (EA).

One of the OPC client and Foundation Fieldbus restrictions is the inability to allocate and deallocate function blocks in execution time. This operation is conducted by supervisors in plant control planning and any modification discontinues its operation. The operator performs the new configuration using proper software such as Syscon . Our strategy is to create a macro FB configuration from which others can be derived. In other words, changing the interconnections between allocated function blocks, it also changes control strategies. Figure 8 shows a number of possibilities of function block changes caused by a

predetermined macro allocation. The advantage of this approach is the use of more than one ANNs. Agent actions are performed by an artificial neural network. A change in the interconnections between function blocks also leads to a change in ANN structure, which, in turn, changes agent structure. Thus, we exchange fieldbus agents through the function block interconnection configuration.

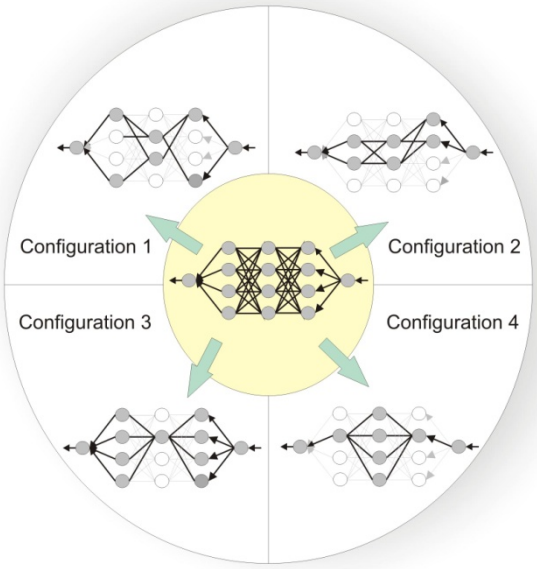


Fig. 8. - Change in Function Block structure

We have also apllied our architecture in FBSIMU to compare a real approach to a simulated one. It is important to underline that, in the FBSIMU, is possible to change all function blocks allocation in execution time, what is not possible in real fieldbus environment.
As an ANN, the agents go through two phases. The agents must learn (ANN train) and act (trained ANN use). They learn about the device (e.g. the OA learns how to predict device output values and the EA learns how to compensate for noise in device measurement). This training is conducted at the supervisory level and the device communicates with the fieldbus through a LABVIEW-FIELDBUS layer. The learned data (neural network weights) are stored in an information repository (IR).
In the learning phase the agents will train an artificial neural network to learn about fieldbus behavior. In the execution phase, the observation agents are able to monitor the field device and detect any faults therein. The OA and EA learns about the device or FBSIMU output (i.e., the OA learns how to predict device output values and the EA learns how to compensate for noise in device measurement, for example). The learned data (neural network weights) are stored in the information repository (IR). The EA uses this learned data to monitor devices. The system output is monitored In FBSIMU.
When a problem (malfunction) is detected, it must be correctly diagnosed to ensure a proper correction. The diagnostic agent (DA) consults the information repository to identify the

type of problem that is occurring in the fieldbus. As soon as the problem is detected (by the observation agent), and identified (by the diagnostic agent) from the information repository, the execution agents decide the best configuration to resolve the problem. The execution agents are function blocks that is used in the devices. The organization of these blocks characterizes the way the EA solves the problem (algorithm). At the end, we have an error-free signal (or an output), for example. In the FBSIMU environment we replace the original schedule table for one which simulates an ANN function block configuration. A new table means a new process control. As previously discussed, the implementation of these agents is based on a structure formed by function blocks. Each function block executes a different kind of algorithm, and together they are capable of meeting a specific application, such as an ANN.

5.1 Observation Agents

Observation Agents, launched in a fieldbus, monitor the variable values of a number of devices. These agents aim to detect measurement anomalies in sensor values or actuator inaccuracies. Previous information about the system behavior is important to properly detect and diagnose faults. Thus, automation engineers can associate the faults with signal patterns. In recent years, research carried out in fault detection and isolation systems (FDI) has shown procedures that use computer intelligence procedures, such as the Fuzzy Logic system and Artificial Neural Networks.

In this work the observation agents use an ANN to predict the measured value of a device. Therefore, the OA must learn how to predict the measured signal behavior before it is launched in a fieldbus. This learning is accomplished at the supervisory level, that is, neural network training for a prediction problem. Thus, the observation agents know the expected signal behavior. When it is launched in fieldbus (ANN with trained weights), the OA tries to predict the next signal and compares it to the real signal. A difference in signals indicates a problem.

5.2 Diagnostic Agents

When a problem (fault) is detected, it must be correctly diagnosed to ensure proper correction. The diagnostic agent (DA), like an AO, uses neural networks to correctly diagnose which fault is occurring in the fieldbus. it is necessary identify which problem is occurring in the process. The DA is based on previous work (Fernandes et al., 2007), and it identifies the type of fault is occurring in the system. Like other agents, the DA is based on an ANN. It is implemented at the supervisory level which communicates with the fieldbus through the OPC client. The neural identification system is defined as a two-step identification process, signifying the existence of an ANN to evaluate a system output value. The general scheme of the functioning system is shown in Figure 9. In this case, while the level system is in execution, a system from the ANNs tries to find its identification using its $(x(k))$ inputs. Each time, output level system $(y(k))$ is compared to output identification system $(y'(k))$, generating a residue value $(r(k) = y(k) - y'(k))$ that is used later in the fault isolation/classification system. Then, one system analyzes the residue values and indicates the occurrence or not of faults. When faults are detected, it indicates which type is occurring. In Section 6, we show this approach applied in a real example.

5.3 Execution Agents

As soon as the problem is detected by the observation agents, the execution agents change function block interconnections to allocate an algorithm (ANN) which can fix the problem. The organization of these blocks determines how the EA solves the problem (algorithm). The execution agents can act (configure) in different ways to correct the errors. As it was previously discussed, the implementation of these agents is based on a structure formed by function blocks.

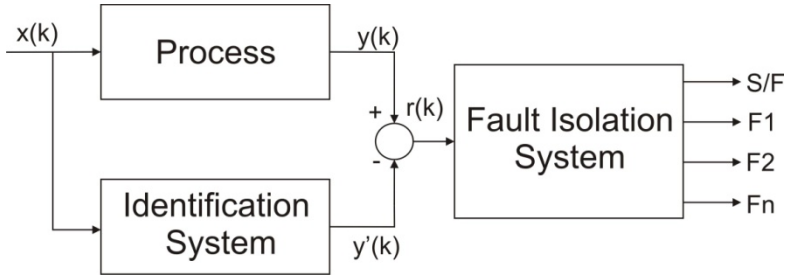


Fig. 9. - General scheme of FDI system (Fernandes et al., 2007)

A type of EA is illustrated in Figure 10. In this case, the structure formed by the function blocks contains a neural network-based noise filter. This algorithm (ANN) is able to remove noise from a measured signal. This structure is explained in Section 6.

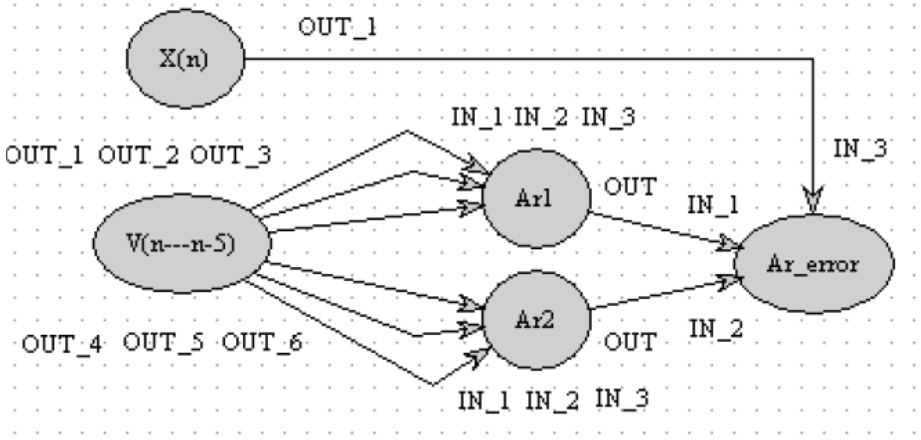


Fig. 10. - Noise Filter implemented as a Neural Network in Function Blocks as seen in SYSCON

The flexibility given by the application layer (represented by the function blocks) in the Foundation Fieldbus protocol enables different implementations at the field level. The combining of arithmetic and function characterizer blocks can produce a configuration similar to a neural network neuron (Silva et al., 2006). Thus, these examples show the variety of applications generated by different execution agents. For example: if the problem is noise

interference collected by the sensor, the execution agents combine function blocks to form a neural network. With training values former acquired, this network can function as a filter (Costa et al., 2005), decreasing the value measured by the sensor. In the event of a tendency toward loss of accuracy in the measured values, detected by the observation agents, the EA can act as a prediction system, anticipating the presumed supervisory faults that may occur. The EA can also act as a recalibration algorithm in detecting a decalibrated sensor (Cagni et al., 2005).

6. MultiAgent Architecture Example

6.1 FBSIMU Example

In this example, we show the proposed multiagent architecture which uses the function blocks configuration exchange approach as it was previously showed. A simulated process automation was implemented for testing the feasibility of the architecture. The test environment consists of a simulated fieldbus-based automation system (FBSIMU) and a prototype agent application. The simulated process is controlled by a AI-PID-AO function blocks represented in Figure 4.

This environment is used in our example to detect and remove noise. First, the agents (ANN) undergo a learning process. The Observation Agent is trained to predict a FBSIMU output at given moment, based on its past outputs, considering that the simulated signal is noise free. The neural networks used by the Diagnostic Agent are trained to identify a kind of problem that may occur in the simulated environment. The EA is trained to act as a noise filter. The information acquired by the agent is stored in the Information Repository and can be used by other agents, in other situations, if necessary.

When the learning phase is over, the Observation Agent starts monitoring the FBSIMU output. Indeed, this agent is a prediction ANN. It monitors the output signal and predicts its corresponding value in the next step.

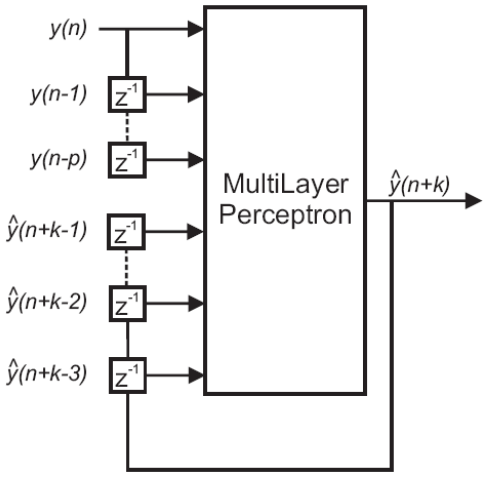


Fig. 11. - Recurrent Neural Network as Predictor

The type of prediction architecture used in this study is shown in Figure 11. It is a model with overall refeeding, resulting from a multiple-layer perceptron. The model has a single input, which is applied to delay line memory with *n*-units. It has a single output which is refed from the input of another delay line memory. The contents of these two memories are used to feed the input layer of the network.
In our test, FBSIMU introduces a simulated noise signal to the signal monitored by the agent (Figure 12). In certain moment, the output signal starts to exhibit different behavior as it was predicted. The difference between the measured and predicted signals is considered a problem by the OA.

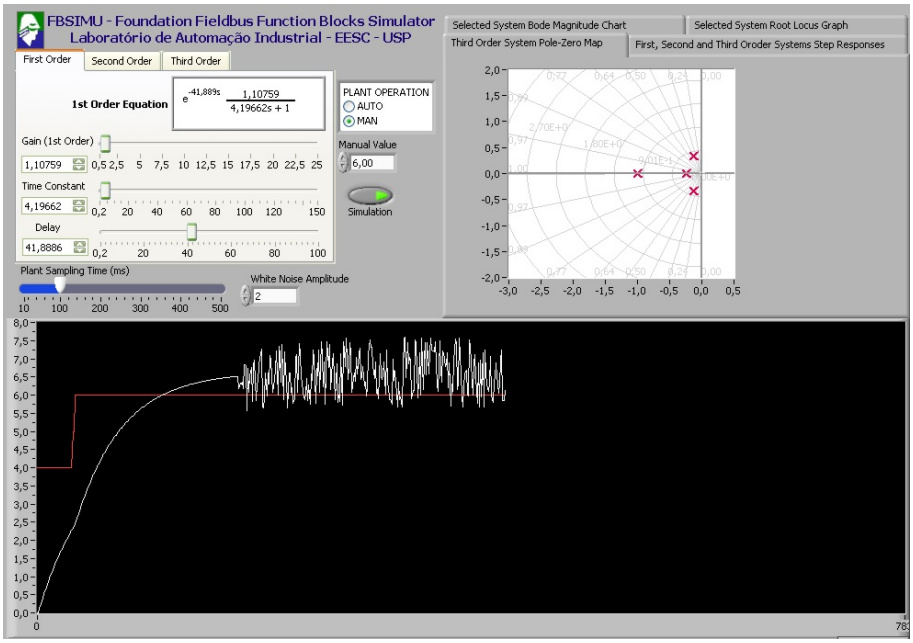


Fig. 12. – Output signal with noise in FBSIMU

At this moment, it is necessary identify which problem is occurring in the process. The diagnostic agent (DA) is responsible for identifying the problem. Like other agents, the DA is based on an ANN. It is implemented at the supervisory level which communicates with the fieldbus through the OPC client.
As mentioned in Section 5.3, the DA tries to find its identification using its inputs. Each time, FBSIMU (y(k)) is compared to predicted output (y'(k)), provided by AO, generating a residue value ($r(k) = y(k) - y'(k)$). This residue value indicates the occurrence or not of faults. If it is closer to zero, this indicates no faults. Otherwise, it indicates which type is occurring. If residue is positive, it indicates a positive noise, if not, it indicates negative noise. In this particular example, we simplify the detection. The Diagnostic Agent determines noise presence or not.
In this example, the DA detects positive. This means there is noise in the FBSIMU output. At this moment, the function block parameters change to allocate the EA as an ANN acting as a

noise filter, trained recursively until reasonable noise extraction is achieved. In the FBSIMU the EA exchanges the schedule table (Figure 13). The normal function block schedule is replaced by a new table which represents the EA allocation (trained ANN to remove noise). This function block allocation (EA) continues until the problem is solved (removing noise), then, the function block parameters (schedule table) change again to allocate the normal control and OA, and it restarts tracking the FBSIMU output.

6.2 Real Environment Example

In this second example, we show the proposed multiagent architecture which uses the function blocks configuration exchange approach, as it was previously showed. A prototype version of process automation was implemented for research purposes, and a laboratory test environment was used for testing the feasibility of the architecture. The test environment consisted of a simulated test, a fieldbus-based automation system and a prototype agent application. The test process contained parts that were similar to industrial processes. This environment was presented for the first time in our previous work (Machado et al., 2008a), and the results were showed in Machado et al. (2008b).

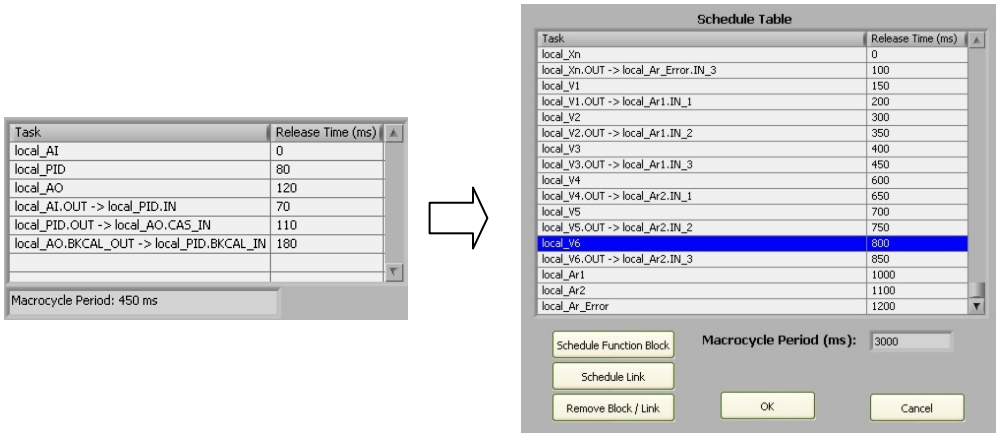


Fig. 13. – Schedule Table Exchange

As we can see in Figure 14, the plant level is composed of two cascading tanks. The water that flows out from the small hole of tank 1 falls into tank 2. This tank also has a small hole through which the water falls directly to the reservoir. A pump impels the water from the reservoir to tank 1. In each tank there is a Foundation Fieldbus pressure sensor, used to measure the corresponding levels connected to the Fieldbus network. Besides the pressure sensors, an FF/ loop of current from a 4 to 20 mA converter is used to send signals to the water pump. The industrial network Foundation Fieldbus is connected to a supervisory computer through Ethernet network interfaces. All the device configuration processes are carried out from this computer, and later supervised. This system transmits signals to the pump input to allow for water injection (or not) in tank 1 and to control the water level in both tanks. There is also a PC (OPC client) that sends a simulated noise signal (red dotted line) to a device.

Like previous FBSIMU example, this environment is used by agents to detect and remove noise. The Observation Agent is trained to predict a sensor output at given moment, based on its past outputs, considering that the signal is noise free. The neural networks used by the Diagnostic Agent is trained to identify a kind of problem that may occurs in the tanks. The EA is trained to act as a noise filter. When the learning phase is over, the Observation Agent starts monitoring the field device, which is allocated in the function blocks. This agent is allocated as a prediction ANN in field devices. It monitors the output signal and predicts its corresponding value in the next step. The ANN model for prediction is the same showed in Section 6.1 (Figure 11).
In our test, a PC (OPC client) sends a simulated noise signal to the device monitored by the agent. In Figure 15 A, we can observe both real and predicted signals. In half of samples, the sensor output signal starts to exhibit different behavior from that predicted (noise added by OPC Client). The difference between the compared signals is considered a problem by the OA. This difference can be seen in Figure 15 B.

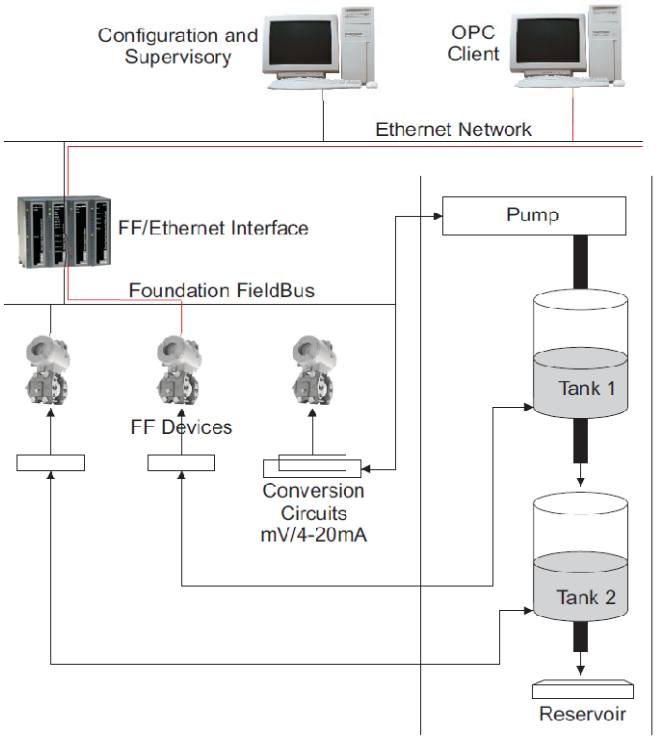


Fig. 14. – Laboratory environment

At this moment, the Diagnostic Agent (DA) is responsible for identifying a problem and selecting the best function block allocation to solve it. The DA is based on the previous described work (Section 6.1) which identifies what kind of fault is occurring in the tanks system. The DA is implemented in supervisory level and communicates with the fieldbus

through an OPC client. The neural identification system was defined as identification in two steps, which means the existence of an ANN to evaluate the level of tank 1, and another to evaluate the level of tank 2.

With identification in two steps, it is possible to get two residues: r(1) and r(2), where $r(1) = l(1) - l'(1)$ and $r(2) = l(2) - l'(2)$. $l(1)$ is the real measured signal, and $l'(1)$ is the predicted signal. In this case, an ANN, named ANN 3, is trained by receiving as input data the values from $r(1)$ and $r(2)$. The networks output corresponds to a vector of $n + 1$ numbers, numbers, where n is a quantity of faults that the network is able to classify.

Considering the two residues to detect and isolate the faults, the FDI system could detect in maximum 8 different faults, in which the situation $r(1) = 0$ and $r(2) = 0$ would be a normal behavior of the system. In view of the test, we managed to foresee only six types of faults with joint distinct residues.

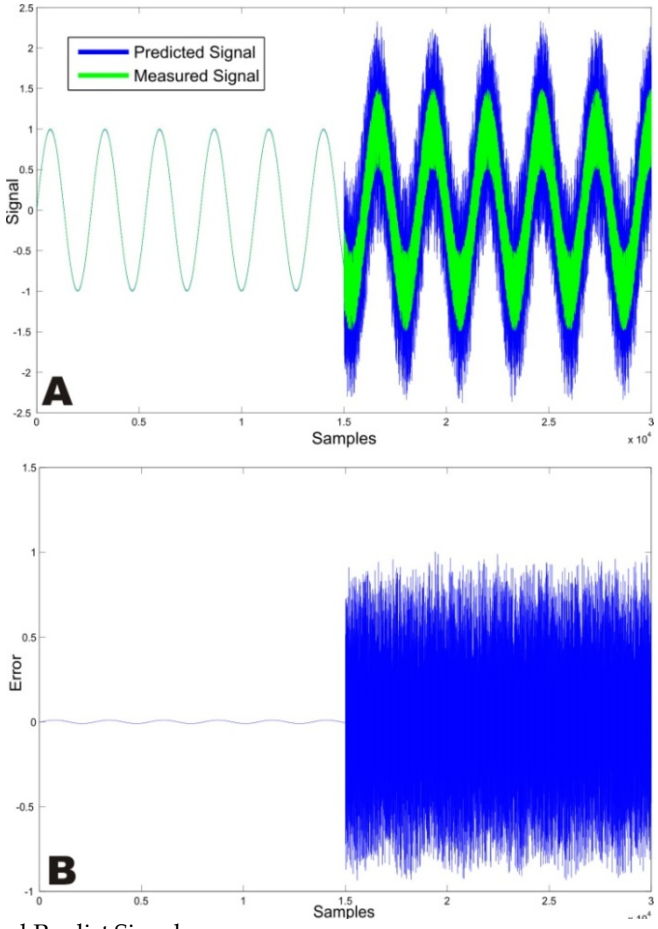


Fig. 15. – Real and Predict Signals

Table 2 shows the five types of faults selected for the result analysis, where (+) represents the positive residue, (-), the negative residue, and (0), the equal residue or very close to zero. In this example, the DA detects positive residue for $r(1)$ and null value for $r(2)$. This means there is a noise in tank 1 output sensor. At this moment, the function block parameters change to allocate the EA as an ANN, acting as a noise filter, once it was trained recursively until a reasonable noise extraction.

Fault	Description	R(1)	R(2)
	Absence of fault	0	0
1	New hole in tank 1. No fall of water in tank 2	-	0
2	Decrease of hole in tank 1	+	-
3	Decrease of hole in tank 2	-	+
4	Increase of hole in tank 2	0	-
5	Read error in Sensor 1. Positive Bias. (Noise Added)	+	0

Table 2 – Dispositions of the 6 fault residues

The EA (acting as a filter) acts immediately after noise detection. This function block allocation (EA) continues until the problem is solved, then, the function block parameters change again to allocate the OA and restart device tracking. It is important to emphasize that for experimental purposes the function block structure is replaced by another when the agents are exchanged (e.g., Observation Agents to Execution Agents). If the number of devices is substantial, the architecture can allocate many OA and EA at the same time, monitoring and acting on different devices and fault situations.

7. Conclusions

Nowadays, problems in the field devices are detected by supervisors through alarm triggers. From the proposed SMA architecture, the agents are able to detect and apply intelligent algorithms to solve these problems without user intervention. This article shows multiagent architecture which is able to detect and correct an undesired noise in a Foundation Fieldbus device and simulated environment (FBSIMU) by implementing function block intelligent algorithms. The intelligent algorithms use Artificial Neural Network (ANN) to find out about the noise and remove it. The agents encapsulate these ANNs and, using a LABVIEW-Fieldbus layer, can directely interact with field devices through an OPC client. In this approach the algorithm is implemented in the device function blocks providing a solution at fieldbus level.

This chapter provides two innovations in fieldbus research mentioned in our previous works. First, we have a dynamic function block interconnection exchange that allows the allocation of different neural network structures at fieldbus level. Accordingly, we have several control strategies (agents) allocated in field devices, which can monitor, detect, and correct a number of faults. The second innovation is the reusing of function block configurations. The same ANN structure can be used in different situations. That is, the same agent can act in other processes. The use of Information Repository allows us to share the ANN structure with other agents.

However, the main novelty in this chapter is the use of our approach in a function block simulated environment (FBSIMU). Thus, we can compare two types of function block intelligent algorithm implementation. The first one, in FBSIMU, is able to exchange all function blocks configuration in execution time. This approach is not permitted in real fieldbus environment. In this (second) case, we opt for a function blocks interconnections exchange. So, we can observe, despite second approach is well suited for a real Foundation Fieldbus environment, the full function block allocation and deallocation is more appropriate for exchange control algorithms in fieldbus devices function blocks. For safety reasons and respecting the industries patterns, the real time function block exchange was not implemented by foundation fieldbus manufacturers. We believe that our approach can supply this gap.

8. References

Albert , M.; Längle, T.; Wörn, H. (2003). Generic Diagnostic Functionalities Encapsulated within a Software Agent. *IAR-ICD/IFATIS/ MAGIC Workshop on Advanced Control and Diagnosis*.

Brandão, D. (2005). Ferramenta de simulação para projeto, avaliação e ensino de redes fieldbus, *Doctorate thesis*. Escola de Engenharia de São Carlos, USP.

Brennan, R. W.; Zhang, X.; Xu, Y.; Norrie, D. H. (2002). A reconfigurable concurrent function block model and its implementation in real-time java, Integr. *Comput.-Aided Eng.*, vol. 9, no. 3, pp. 263–279, 2002.

Cagni, E.; Pereira, D.; Pereira, A.; Doria Neto, A. D.; de Melo, J. D.; Guedes, L. A. (2005).The implementation of the self-calibration, self-compensation and self-validation algorithms for foundation fieldbus sensors are presented using standard function blocks, IEEE International *Conference on Computational Intelligence for Measurement Systems and Applications*, pp. 220–225.

Cavalieri, S.; Di Stefano, A.; Mirabella O. (1993). Optimization of acyclic bandwidth allocation exploiting the priority mechanism in the fieldbus data link layer. *IEEE Transactions on Industrial Electronics*, pp. 297–306.

Chen, J.; Wang, Z.; Sun, Y. (2002). How to improve control system performance using FF function blocks. In: *IEEE international conference on control application*. Glasgow, Scotland, pp.1022-1026.

Costa, I.; Doria Neto, A. D.; de Melo, J. D.; de Oliveira J. (2005). Embedded FASTICA algorithm applied to the sensor noise extraction problem of foundation fieldbus network, Neural Networks, 2005. IJCNN *'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 4, pp. 2217–2221.

Fei-Yue, W.; Haitao, Z.; Yunfeng, A. (2005). An OSGi and agent based control system architecture for smart home, *Proc. Networking, Sensing and Control*, pp. 13–18.

Feng, Q.; Bratukin, A.; Treytl, A.; Sauter, T. (2007). A flexible multi-agent system architecture for plant automation, in *5th IEEE International Conference on Industrial Informatics (INDIN 2007) Industrial Informatics*, IEEE Transactions on, pp. 1047–1052.

Fernandes, R. G.; Silva, D. R.; Guedes, L. A.; Doria Neto, A. D. (2007). An implementation of a fault detection and isolation system on foundation fieldbus environment. *International Journal of Factory Automation*, Robotics and Soft Computing, vol. 3, pp. 130–136.

Ferreiro, R.; Vidal J.; Pardo, C.; Coego, J. (1997) Fieldbus: Preliminary design approach to optimal network management. In: *IEEE international workshop on factory communication systems*, pp. 321 – 325.

Fieldbus Foundation. (1999b). *Foundation specification function block application process,* Part 3: FF-892 – FS1.4. Austin, USA.

Fieldbus Foundation. (1999c). *Foundation specification function block application process*, Part 5: FF-894 – DPS0.95. Austin, USA.

Fieldbus Foundation. FF-890-1.3. (1999a). *Foundation specification function block application process*, Part 1. Austin, USA.

Haykin, S. (1999). Neural Networks - A Comprehensive Foundation. *Prentice Hall*.

Hong, S.H.; Ko, S.J. (2001) A simulation study on the performance analysis of the data link layer of IEC/ISA fieldbus. SIMULATION, pp. 109–18.

International Electrotechnical Comission. (2000). IEC 61158: *Digital data communications for measurement and control – fieldbus for use in industrial control systems.* Switzerland. CD- ROM.

International Electrotechnical Comission. (2003). IEC 61784: *Digital data communications for measurement and control - Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems.* Switzerland. CD-ROM. 2003.

International organization for standardization. (2009). ISO/IEC 7498-1: Information technology – open systems interconnection – basic referencemodel: The basic model. Switzerland. CD-ROM.

Jennings, N.R.; Bussmann, S. (2003) Agent-based control systems: Why are they suited to engineering complex systems?, in *IEEE Control Systems Magazine*, v. 23, Issue 3, pp. 61-73.

Ljung, L. (1999). *System identification- theory for the user*. Englewood Cliffs: Prentice Hall.

Machado, V.; Doria Neto, A. D.; de Melo, J. D.; Ramalho, L.; Medeiros, J. (2008a). Multiagent architecture for function blocks: Intelligent configuration strategies allocation, in *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, 2008, pp. 1377–1382.

Machado, V.; Doria Neto, A. D.; de Melo, J. D.; Ramalho, L.; Medeiros, J. (2008b). A neural network multiagent architecture applied to fieldbus intelligent control. in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. IEEE, pp. 567–574.

Petalidis, N.; Gill, D. S. (1998) The formal specification of the fieldbus foundation link scheduler in E-LOTOS. In: *International conference on formal engineering methods*.

Pinotti Jr., M.; Brandão, D. (2005). A flexible fieldbus simulation platform for distributed control systems laboratory courses. *The International Journal of Engineering Education*, pp. 21(6):1050–8. Dublin.

Pirttioja, T.; Pakonen, A.; Seilonen, I.; Halme, A.; Koskinen, K. (2005). Multi-agent based information access services for condition monitoring in process automation, in *Proc. INDIN '05, 3rd IEEE International Conference on Industrial Informatics*, pp. 240 – 245.

Polaków, G.; and Metzger, M. (2007). Agent-Based Approach for LabVIEW Developed Distributed Control Systems. *In Proceedings of the 1st KES international Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. N. T. Nguyen, A. Grzech, R. J. Howlett, and L. C. Jain, Eds. Lecture Notes In Artificial Intelligence, vol. 4496. Springer-Verlag, Berlin, Heidelberg, pp. 21-30.

Pop, T.; Eles, P.; Peng, Z. (2002). Holistic scheduling and analysis of mixed time/event-triggered distributed embedded systems. In: *10th international symposium on Hardware/software* codesign. pp. 187 – 192.

Russell, S.; Norvig, P. (2003) Artificial Intelligence: A Modern Approach, 2nd ed. *prenticeort: prentice*.

Schoop, R.; Colombo, A.W.; Suessmann, B.; Neubert, R. (2002). Industrial experiences, trends and future requirements on agent-based intelligent automation, in Proc. IECON 02, *28th IEEE Annual Conference of the Industrial Electronics Society*, pp. 2978–2983, vol.4.

Seilonen, I. (2006) An extended process automation system: An approach based on a multi-agent system. *Ph.D. dissertation*, Helsinki University of Technology, Espoo, Finland.

Seilonen, I.; Appelqvist, P.; Koskinen, K. (2002a). Agent-based approach for faulttolerance in process automation systems, in *Proceedings of the 3rd International Symposium on Robotics and Automation* (ISRA 2002).

Seilonen, I.; Pirttioja T.; Appelqvist, P. (2002b). Agent technology and process automation, pp. 31–35.

Silva, D.; Guedes L. A.; Doria Neto, A. D.; de Melo, J. D. (2006) "Neural networks implementation in foundation fieldbus environment: A case study in neural control", *International Journal of Factory Automation, Robotics and Soft Computing,* pp. 48–54.

Taylor, J. H.; Sayda A. (2005). An Intelligent Architecture for Integrated Control and Asset Management for Industrial Processes, in *Proc. IEEE International Symposium on Intelligent Control*, Limassol, Cyprus, pp. 27-29.

Theiss, S.; Vasyutynskyy, V.; Kabitzsch, K., (2008). AMES - a resource-efficient platform for industrial agents, *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on* , pp.405-413, 21-23.

Wang, Z.; Yue, Z.; Chen, J.; Song, Y.; Sun, Y. Realtime characteristic of FF like centralized control fieldbus and it's state-of-art. In: IEEE international symposium on industrial electronics. 2002.

Weyns, D.; Holvoet T. (2007). Architectural design of a situated multiagent system for controlling automatic guided vehicles, *International Journal on Agent Oriented Software Engineering*, vol. 1, no. 4, pp. 1–39.

Weyns, D.; Schelfthout, K.; Holvoet, T.; Lefever, T.; (2005). "Decentralized control of e'gv transportation systems," in Autonomous Agents and Multiagent Systems, pp. 67–74. [Online]. Available: citeseer.ist.psu.edu/weyns05decentralized.html

19

Production System's Life Cycle-Oriented Innovation of Industrial Information Systems

Dencovski Kristian, Löwen Ulrich,
Holm Timo, Amberg Michael,
Maurmaier Mathias and Göhner Peter
Dept. of Systems Engineering, Siemens AG Corporate Technology
Dept. of Information Systems III, Friedrich-Alexander-University Erlangen-Nuremberg
Institute of Industrial Automation and Software Engineering, Universität Stuttgart
Germany

1. Introduction

Global companies of various industries – for instance the automotive industry – increasingly compete for key market shares. This frequently leads to an innovation race between competitors which often centers around certain trends or changed basic conditions (see Kiefer et al., 2006). Examples for these conditions are (see also Figure 1):

1. Increase of complexity of production systems due to higher demands on flexibility and the need to minimize necessary resources
2. Increase of number of mechatronic components within the production system
3. Reduction of startup / ramp up times
4. Increase of production life cycle

For Siemens AG as supplier of automation solutions for production systems the postulated changed basic conditions as well as trends result in challenges, which need to be addressed in an adequate manner:

1. Increased complexity of production systems leads to an increased complexity of the engineering process of production systems. The complexity must be handled during the engineering phase by means of efficient engineering processes and methods.
2. The increased number of mechatronic components demands new engineering methods, which efficiently support the overall approach of mechatronic components.
3. Reduction of startup as well as ramp up times (time-to-operation) demands standardization of solutions including efficient concepts of reuse.
4. Increase of production life cycles demands a change from craft- and phase-specific approaches towards an integrated view on production systems over the whole life cycle (Preiss et al., 2001) in order to reduce total cost of ownership and increase total value. It also demands an increasing importance of offerings for technical services and modernization.

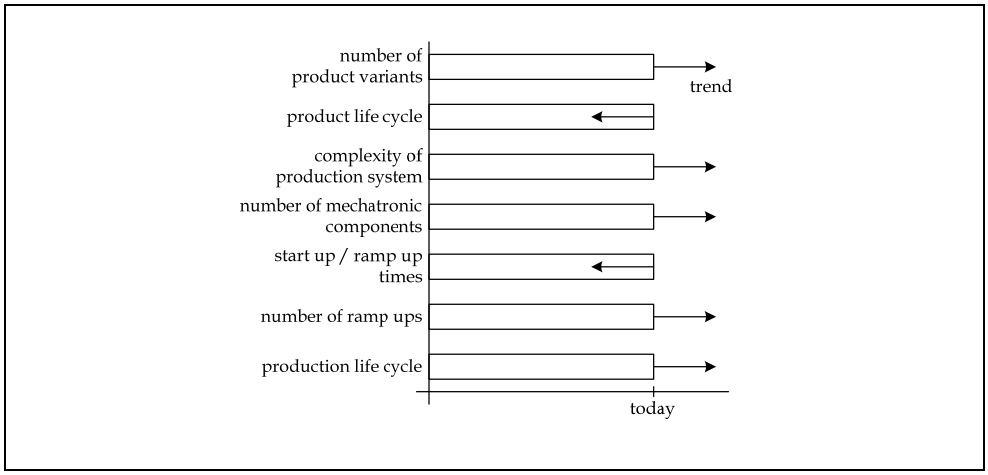


Fig. 1. Changed Basic Conditions (based on Kiefer et al., 2006)

Industrial Information Systems are becoming a more-and-more important instrument to increase the productivity of production systems by supporting engineers during all phases of the production system's life cycle adequately. But the Industrial Information Systems currently available on the market are often not designed adequately in order to cope with challenges posed by above mentioned trends and conditions. Instead they support craft-specific tasks and automation device specific functions, which are mostly craft- and device-centric. Furthermore Industrial Information Systems often lack integration along a production system life cycle and are therefore not capable to affect the overall life cycle cost effectively.

From the automation solution supplier's point of view new approaches are needed, which are capable of coping with the altered basic conditions. In order to reduce the total cost of ownership, i.e. demanded reduction of cost and time over the whole life cycle (Schott, 2007) on the one hand and to increase the total value of ownership on the other, the focus must be set to (Abel et al., 2003)

- integration of automation with other crafts
- integration along the complete production system life cycle,

while at the same time considering the implications for Industrial Information Systems.

The vision pursued by Siemens AG to address above mentioned challenges is the *Digital Factory* (Wucherer, 2006). Digital Factory is the generic term for a comprehensive network of digital models, methods and tools. Its aim is the holistic planning, evaluation and ongoing improvement of all the main structures, processes and resources of the real factory (VDI 4499, 2008). The term Digital Factory is well established in the field of factory automation and especially within the automotive industry. Behind this term stands the idea to digitize all information belonging to a production system and represent it by models - especially mechanical-, electrical-, as well as automation related information - and to provide it during all life cycle phases using suited tools, i.e. Industrial Information Systems (example see Figure 2).

For instance, engineering the production system based on mechatronic components in the Digital Factory takes place in a virtual manner using Industrial Information Systems.

According to (VDI 2206, 2008), mechatronics is an interdisciplinary field in which the following crafts - respectively corresponding systems - interact: Mechanical systems […], electronic systems, information technology. According to Siemens AG a mechatronic component can be understood as a collection of mechatronics aspects (especially mechanical, electrical and automation related) that represent a functional intent. So the mechatronic component is a digital representation in the production system life cycle and contains all information that is needed during production system life cycle means. The aim is to plan, check and control a production system entirely from the initial engineering to operation in order to make production as a result more flexible and cost effective. Only when all elements of the production system's life cycle are seamlessly linked to each other, flexible production can be achieved to the fullest degree. According to Siemens AG, integrated engineering for mechanics, electrics and automation will be available in 10 to 15 years not only but especially for automotive industry (Hiesinger, 2008). In subsequent life cycle phases of the production system the Digital Factory provides benefits by visualization and simulation of the processes taking place in the real production system.
In order to face the described changed basic conditions and trends as well as the resulting challenges for automation solution suppliers, powerful Industrial Information Systems are needed. But nowadays Industrial Information Systems do not provide the concepts necessary to realize the vision of the Digital Factory. Therefore an evolutionary and step-by-step method to innovate Industrial Information Systems towards the idea of the Digital Factory is needed. Such a method for life cycle-oriented innovation of Industrial Information Systems is introduced in the following chapters.

Fig. 2. Digital Factory (Source: Siemens AG)

2. Industrial Project Business

Aim of every production system is to implement a technical process in a cost and resource efficient manner so that a certain target corridor of quality is reached for the produced goods. The production system owner's intent is to maximize the Total Value of Ownership (TVO). Of course the cumulated cost across all life cycle phases has to be considered when calculating the TVO. The life cycle can be divided into six phases: Engineering, commissioning, operation, technical service, modernization and decommissioning.
Although engineering is one of the shorter phases of the production system's life cycle, a major part of the total cost that emerges in later phases - like for instance operation or service - is determined during engineering. As one can easily reason when looking at Figure 3, it is not sufficient to consider only one particular life cycle phase when trying to maximize the TVO.

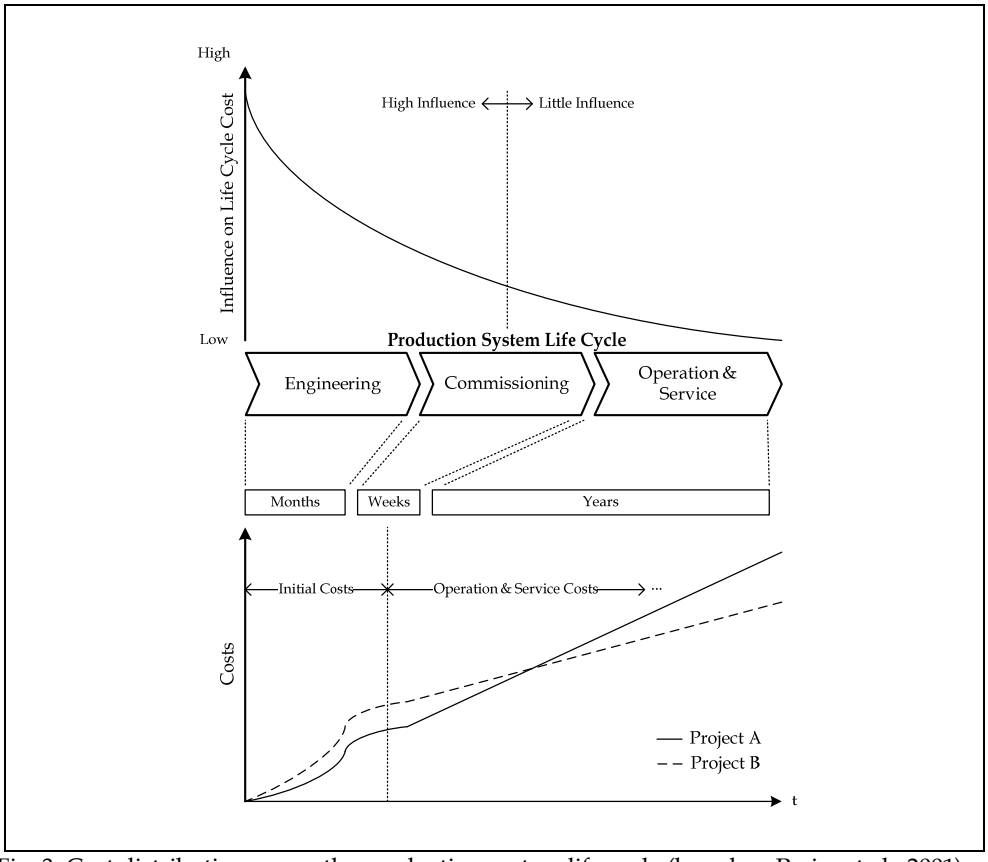


Fig. 3. Cost distribution across the production system life cycle (based on Preiss et al., 2001)

Therefore Siemens Corporate Technology, together with Friedrich-Alexander-University Erlangen-Nuremberg and Universität Stuttgart, have developed a method for targeted

innovation of Industrial Information Systems (IIS) which is introduced below. This method explicitly considers interrelations between single life cycle phases and thereby supports the innovation of IIS so that their application can help enhancing the TVO. Those IIS are not only used by production system's owners but also by other stakeholders like those who provide engineering, procurement and construction - so-called EPC - or suppliers. Both generate a major part of the TVO during for instance operation or modernization (see Tayeh, 2009). The IIS have to support their users when executing tasks while at the same time they have to embed themselves into the user's workflows to maximize the TVO. In order to assess the latter property first of all the activities IIS have to support during the entire production system's life cycle have to be examined.

2.1 Activities during a Production System's Life Cycle

When analyzing the activities different stakeholders engage in during the life cycle of a production system it can be observed that two fundamentally different kinds of activities exist:

- *Order-dependent activities* describe tasks, that are executed by a particular stakeholder as part of a dedicated customer project for which the stakeholder accepted a specific order from a customer. Examples for these activities are the engineering of a particular production system as well as its commissioning, operation, service, and modernization.
- *Order-independent activities* are carried out independently from a customer's particular order to prepare order-specific activities in the future. Especially during engineering the costs of customer projects can be reduced significantly due to targeted order-independent development of reusable sub-solutions (Fay et al., 2009). A systematic execution of order-independent activities includes an analysis of the application domain, the business strategy, planning activities as well as the implementation and test of reusable work results (VDI 3695, 2009).

Examples for typically order-independent activities during engineering are the development of a technological production system structure as well as the preparation of mechatronic components that can be used repeatedly. But systematic preparation of reusable work results can increase efficiency during other life cycle phases, too. An example for a life cycle phase-spanning as well as order-independent activity is the provision of an integrated chain of IIS to support order-dependent activities in particular customer projects.

Interdependencies between order-independent and order-dependent activities of a production system's life cycle are visualized in Figure 4. The work results gathered during order-independent activities or during earlier projects can be put into libraries, standards or IIS in order to be reused in later projects. To ensure that these reusable work results comply to customer, market as well as project requirements, the experiences gathered in order-dependent activities have to be fed back.

These activities aren't executed by one single organization. Other stakeholders beside the production system's owner are EPC as well as a multitude of suppliers, which are either responsible for particular activities or might assign them to external suppliers. To maximize the TVO it is again not sufficient to optimize every single activity regarding its cost-value-ratio but to instead consider the interdependencies between particular activities in a global context.

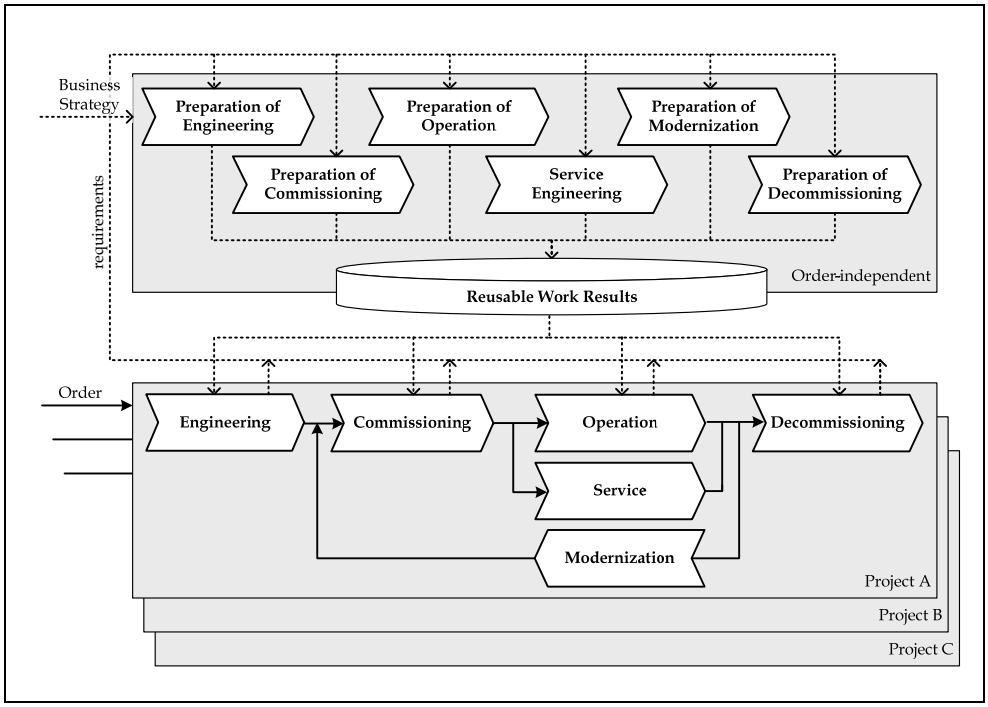


Fig. 4. Order-independent and order-dependent activities in industrial project business

2.2 Challenges of Enhancing Efficiency and Quality

Primary tasks of the activities described in the previous section are of technical nature; after all their intent is to engineer, operate, maintain, modernize, and (de)commission a production system. Challenges of trying to efficiently provide tasks arise primarily from technical activities and belong to one of the following two types:

- Life cycle phase-dependent – Since during every life cycle phase-specific technical tasks have to be accomplished, different challenges arise from particular phases of the production system's life cycle.
- Life cycle phase-spanning – Due to above mentioned properties of the industrial project business common challenges arise, which have to be addressed in all life cycle phases.

During engineering, commissioning, operation, service and modernization of a production system experts of different crafts, for instance mechanical construction, fluid technology, electrical engineering, automation, as well as software engineering, have to work together. Especially technical information has to be shared and interdependencies between individual crafts have to be taken into account. Since every craft uses specific methods, abstractions and modeling languages, integrating the participating crafts is a special challenge.

During the life cycle of a production system a huge amount of technical information emerges. Sometimes this information is needed in only one life cycle phase. Other information is relevant for several life cycle phases and evolves during this process.

Pursuing the idea of the Digital Factory, the relevant digital information has to be accessible in all life cycle phases and needs furthermore to be expandable as well as changeable.
Beside the aspect of life cycle phase integration, integration among different abstraction layers - for instance those of the automation pyramid - is crucial. Although the tasks involved with the particular layers of automation differ in nature, they all access the same technical information of a production system. But this information varies intensely in granularity.
The third aspect of integration is the challenge to integrate information among different crafts. The three aspects of integration are visualized in Figure 5.

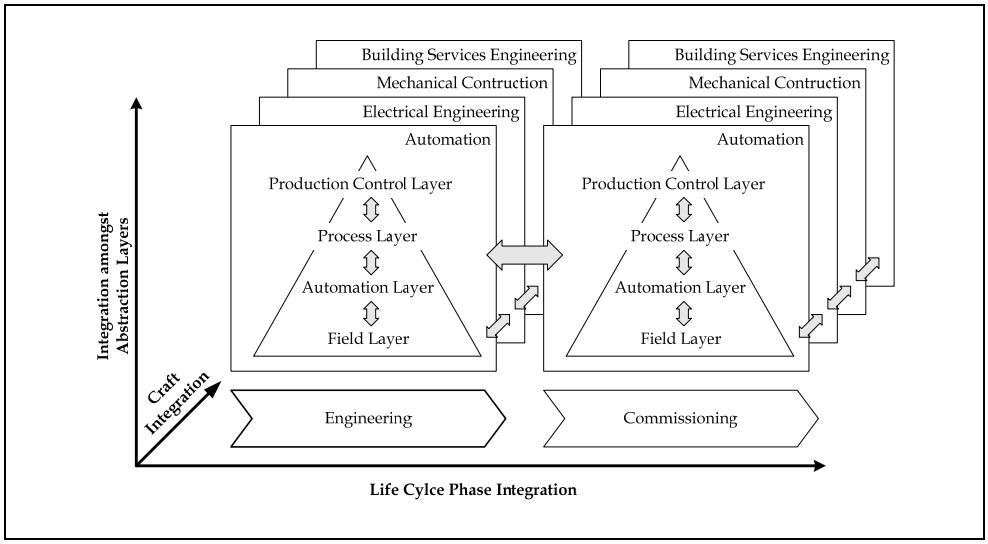


Fig. 5. Dimensions of Integration in Industrial Project Business

Production systems are complex systems which consist of large amounts of often similar components like for instance sensors and actuators. Consequently a generic challenge arises from the efficient handling of large amounts of data sets. These data sets have to be generated, saved, analyzed as well as changed in an efficient manner.
Due to the multitude of stakeholders and the complexity of a production system, it is sometimes necessary to change the planned or even already built production system during particular life cycle phases. These changes must not lead to inconsistencies within the digital information or even worse safety-critical malfunctions during operations. This challenge becomes more meaningful when changes influence multiple crafts or even multiple abstraction layers.
When considering the different aspects of integration it is always important for digital information to represent the real plant correctly. Hence the quality of this digital information, and thus the benefit of a Digital Factory, can be determined by measuring the fraction of digital information, which can be processed by a computer and represents the production system correctly, to the overall information available on the production system. This quality is visualized by the distance between the two curves shown in Figure 6. Due to

undocumented tasks and modifications during for instance commissioning or service the real plant diverges from it's digital shadow. This leads to a reduction in quality of the digital information which consequently necessitates in additional efforts to maintain the digital information.

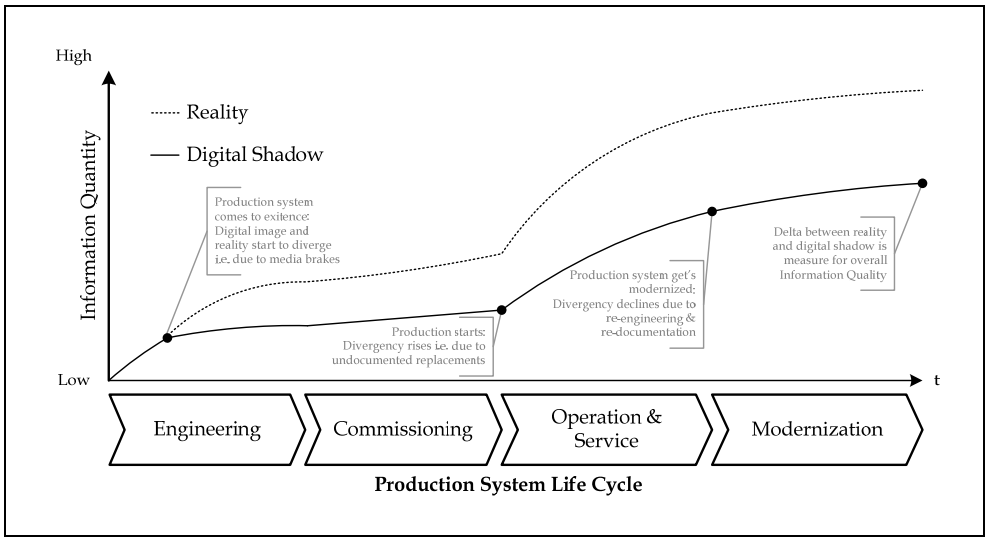


Fig. 6. Digital and computer-processable Information alongside the Production System's Life Cycle

2.3 Concepts to handle Challenges

In order to handle the above mentioned order-independent challenges, as well as challenges associated with individual life cycle phases, it is necessary to systematize the industrial project business (Löwen et al., 2005). This systematization is enabled by means of targeted application of concepts. A concept is a systematic approach to solve a specific problem. By systematically using concepts, which address above mentioned challenges, it is possible to actively cope with the challenges of industrial project business.

Table 1 shows an example of selected concepts as well as the appropriate challenges. Additionally the life cycle phase is given, in which the concept can be applied.

The concept of mechatronic components for example can be used to face the challenges crafts integration and life cycle integration during all life cycle phases. The encapsulation and integration of all information belonging to one mechatronic component facilitates the synergistic cooperation of all involved crafts and the provision of consistent information on the mechatronic component during the whole life cycle of the production system.

Every stakeholder uses a specific set of concepts to address the challenges when executing his tasks. Depending on the task the usable concepts can vary heavily. Often several concepts exist, which all support coping with a particular challenge – of course often having different degrees of performance.

The concept's potential to cope with challenges in industrial project business can be utilized if these concepts are supported adequately by Industrial Information Systems. This aspect is covered within the next chapter.

Concept	Life Cycle Phase	Challenge
Use of Mechatronic Components	All	Crafts integration Life cycle integration
Filing of all information in standardized file formats	All	Data integration Crafts integration
Library of reusable templates	Engineering	Efficient execution of engineering Quality of engineering results
Standard for configuration of technical devices	Commissioning	Integration of devices from different suppliers
Views with different abstractions on diagnostic data	Service	Integration of abstraction layers (i.e. layers of automation pyramid)

Table 1. Concepts associated with Challenges (Examples)

3. Industrial Information Systems

3.1 Definition

Industrial Information Systems are combined hard- and software systems, which support users when executing tasks with primarily technical focus within the industrial project business. These systems consist of at least a software tool executed on a computer. Examples are software tools used for the parameterization of field and safety devices (e.g. Siemens SIMATIC Step 7) as well as motion control units like for instance Siemens' Simotion Scout. Other IIS support their users within several life cycle phases and provide means to be customized to specific application cases. An example for this class of IIS is Siemens' COMOS®. IIS might also bring dedicated hardware components with them, specialized to be coupled to the production process while simultaneously complying to special pro-tection /safety requirements. Examples for this type of IIS are Manufacturing Execution Systems (e.g. Siemens' SIMATIC IT) or Process Control Systems (e.g. Siemens SIMATIC PCS 7) including process-oriented components as well as service and diagnostic tools with corresponding hardware units for the logging of process data (e.g. Siemens' SIPLUS CMS).

3.2 Supporting the Industrial Project Business with Industrial Information Systems

In order to support the user when executing tasks as part of the industrial project business, IIS must implement the user's concepts, in order to cope with above mentioned challenges. Users of IIS are therefore heavily interested in using those IIS, which support the concepts they are employing as accurately as possible. Especially when choosing an IIS but also during its customization, the user needs knowledge regarding the concepts the particular IIS supports. The set of concepts supported by an IIS determines the *philosophy* of the IIS. The

user's aim is to choose not only the one IIS, which offers all functions necessary for the tasks but which also fits well into his workflows and business strategy. Consequently the IIS-supplier needs detailed knowledge regarding the concepts which might be used within a certain craft and also life cycle phase. On the other hand the IIS-supplier needs to know the life cycle phase-spanning concepts which are needed to support the challenges in industrial project business. If the supplier's IIS does not address both the life cycle phase-specific and the life cycle phase-spanning concepts, it does not support the users adequately.
Especially if the IIS-supplier wants to enhance and innovate IIS, a choice must be made which concepts are going to be integrated in which upcoming version of the IIS. To support the IIS-supplier within these decisions, the next chapter introduces a concept catalog, that allows targeted innovation of IIS and can be easily integrated into common information systems development processes.

4. Targeted Innovation of Industrial Information Systems

Knowing the concepts used in industrial project business is necessary but not sufficient for targeted innovation of IIS. The concepts need to be operationalized in order to be integrated into IIS. The aim is to align further development strategically – especially in comparison to competing IIS-suppliers - to increase productivity. Therefore it is necessary to analyze as a first step the underlying philosophy of an IIS in detail. In a second step, measures for a targeted innovation of the IIS can be planned. It is self-explanatory that the development phase is the place to insert this step.

4.1 Information System Development Process

The first process model formally describing the development of not only IIS but information systems in general was introduced by (Royce, 1970) and is called Linear Sequential Model. It describes the information system's life cycle as well as its realization - with a focus on development. It is considered to be the origin that almost all other models - for instance the spiral model or the V-Model - are derived from (Green & DiCaterino, 1993).
Multiple points of criticism exist when it comes to the Linear Sequential Model, for instance:

- it doesn't reflect the possibility to incorporate late changes
- resulting documents are not specified sufficiently regarding their granularity
- the process is – at a whole – unidirectional.

All these points have been addressed in the past and where enhanced piece-by-piece within later models. But almost all of them are based on the Linear Sequential Model, which is why it can be seen as the lowest common denominator. This is why it will be used as an example to demonstrate the insertion of the concept catalog into the development process of IIS.
In the simplest case developing an IIS consists of five phases (see Figure 7). In the requirements analysis phase, the problems addressed by the IIS are specified along with the desired service objectives (goals) and underlying constraints are identified. During the specification phase the IIS' specification is produced from the detailed definitions of the first step. The resulting documents should clearly define the information system's function. In the IIS' design phase, the system specifications are translated into a real life system. The system developer at this stage is concerned with aspects like data structure, architecture, algorithmic detail and interface representations. By the end of this stage the system engineer should be able to identify the relationship between hardware, software and associated

interfaces. Any faults in the specification should ideally not be passed down stream. During the implementation and testing phase the designs are translated into a working system. At this stage detailed documentation from the design phase can significantly reduce the implementation effort. Testing at this stage focuses on making sure that any errors are identified and that the IIS meets its required specification. In the integration and system testing phase all the parts are integrated and tested to ensure that the complete IIS meets the requirements. After this stage the IIS is delivered to the customer. Feed back loops allow for corrections to be incorporated into the model. For example a problem / update in the design phase requires a revisit to the specifications phase. When changes are made at any phase, the relevant documentation should be updated to reflect that change.

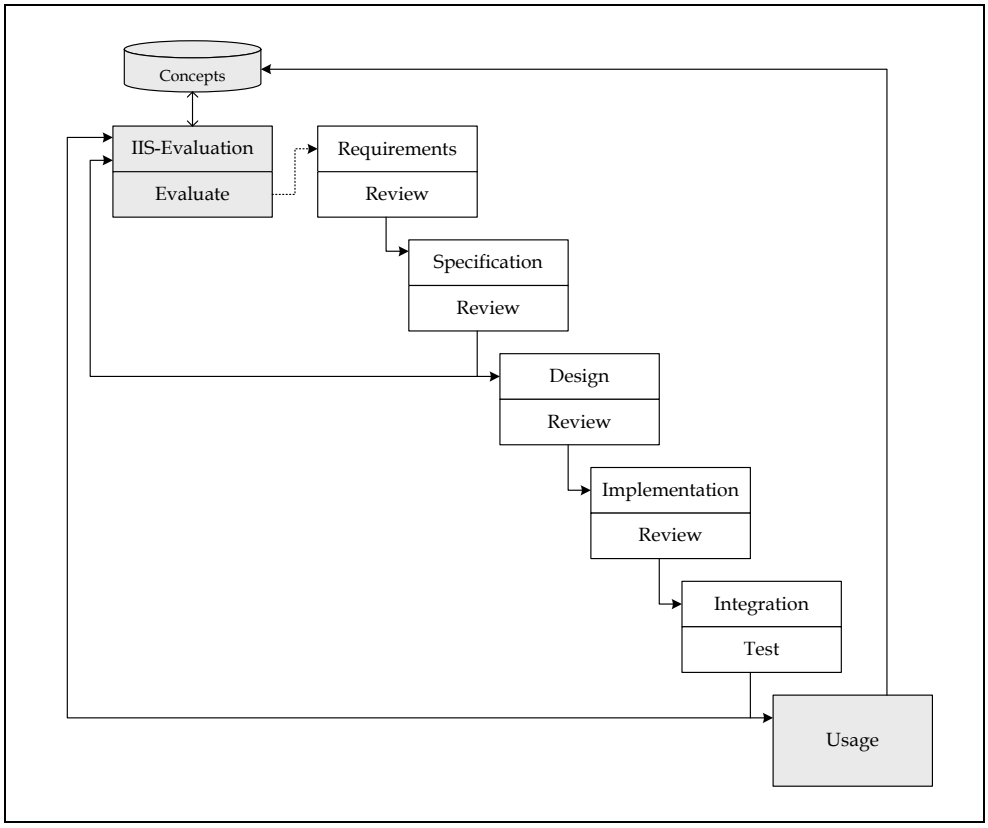


Fig. 7. Integration of concept catalog into an exemplary IIS development process

In Figure 7 the application of our concept catalog (IIS-Evaluation) is put just before the requirements phase, which is necessary for the purpose of targeted innovation. Since the insertion of the concept knowledge gathered should be carried out systematically, the method needs to incorporate a step to evaluate the IIS. During this step the IIS is assessed on the basis of results of chronologically prior process steps (i.e. documentations, specifications) in order to determine its status regarding the concepts and to disclose

potential for innovation - for instance gaps within the specification. Input of the IIS-Evaluation might be specifications – for instance a feature specification as shown in Figure 7 - as well as the released IIS. Depending on the development state the resulting output of the IIS-Evaluation can be incorporated into upcoming versions.
The external interfaces to the IIS-Evaluation, which makes use of the concept catalog, are now defined. It still bares the question how this process step looks like in the inside and especially how the concept catalog is designed in order to efficiently operationalize the concepts.

4.2 Operationalization of Concept Knowledge

In order to operationalize concept knowledge, concepts are aggregated and structured by a reference model. It features success factors of the industrial project business at the top. They are considered to be external quality characteristics IIS are measured with. "*External quality characteristics are those parts of a [system]. that face its users, where internal quality characteristics are those that do not*" (see McConnell, 1993). Some authors use the term *quality in use* and define it as "*the user's view of the quality of the software product when it is used in a specific environment and a specific context of use*" opposed to internal quality, which is measured during implementation phase and external quality measured during testing phase (see for instance ISO 9126-4, 2001). Much like in these definitions, it is the IIS-Evaluation's intention to measure the extent to which users can achieve their goals – but with a focus on a business scenario, rather than measuring generic properties of the plain IIS.
In order to measure quality in use, a multitude of approaches exists that all share the lack of consideration of business domain specificity and instead concentrate on common quality criteria - for instance reliability and usability. They do however bring with them evaluation workflows, structures for characteristics and metrics that can easily be adapted in order to operationalize concept knowledge instead of common quality characteristics. The most recent approach for measuring quality in use is described in ISO 9126, 2001 as well as the associated standard ISO 14598, 1999, which covers the development of an evaluation by means of a four step process:

- Establish evaluation requirements
- Specify the evaluation
- Design the evaluation
- Execute the evaluation

Step one mainly covers the external interfaces to integrate the IIS-Evaluation into a development process as described in chapter 4.1. Specifying structure and metrics of the quality characteristics to be used is part of step two. Fundamental aim of this structure is the applicability of quality characteristics (see also Balzert, 1998 as well as Abran & Buglione, 2000). In case of the IIS-Evaluation generic success factors as well as those of production system life cycle phases – namely engineering, commissioning, operation, service, and modernization – are used to structure the catalog of associated concepts, which are known to be realizable in IIS and simultaneously support the user in an adequate manner. When used on competing IIS as input, the IIS-Evaluation can reveal the used concepts and thereby derive the underlying philosophy of the IIS. This enables IIS-suppliers to innovate in a systematic manner for instance by implementing cutting-edge or unique concepts first and thereby to convince potentially undecided customers in favor of their IIS.

4.3 Structuring Concepts

The structure chosen to break down the universe of concepts follows the models introduced by McCall et al., 1977 and Boehm et al., 1976 which were later adopted by many models - ISO 9126 being one of them. They structure characteristics – which translate to what we identified as *Challenges* - and sub-characteristics hierarchically, having a metric on the lowest level. Every Challenge describes a success factor of either a specific life cycle phase or all life cycle phases and is subdivided further by a number of so-called Sub-Challenges. Every Sub-Challenge describes a single determinant on the efficiency of IIS regarding their addressed business – in our case the industrial project business. For every determinant described by a Sub-Challenge five different concepts are gathered, which cover the corresponding determinant within the IIS and act as a metric (Kitchenham, 1990). Attached to every Best Practice is one central question, which functions as a barrier that has to be overcome in order to reach a certain concept class. Examples are used to substantiate concepts by means of precise applications and case studies. Of course examples can be added as the concept catalog matures (i.e. after a special evaluation), refining it even further. The interrelation between Challenges, Sub-Challenges, Best Practices, Central Questions and Examples is described by a corresponding meta-model (see Figure 8).

The term Challenge in this context translates to characteristic in McCall's approach. It was chosen to reflect filling the structure with characteristics specific for the industrial project business. To discover these characteristics methods of social research like guideline based expert interviews and workshops of experts carrying business knowledge were used (see Yin, 1994). The results of our findings, which make use of the introduced structure, are described within the next section (see also Amberg et al., 2008).

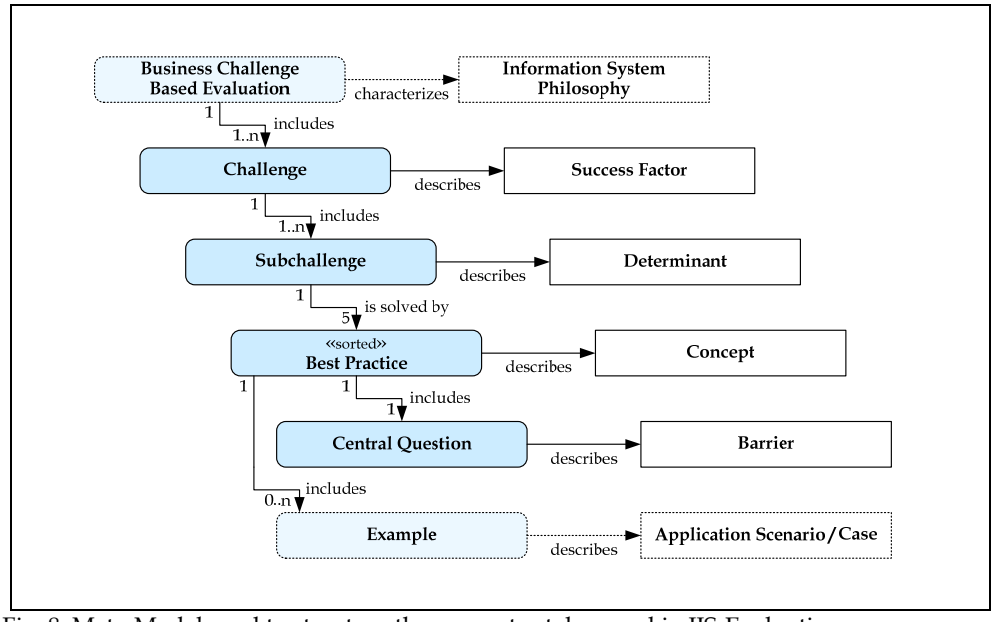


Fig. 8. Meta-Model used to structure the concept catalog used in IIS-Evaluation

4.4 Concept Catalog for the Industrial Project Business

Siemens Corporate Technology in cooperation with Universität Stuttgart and Friedrich-Alexander-University Erlangen-Nuremberg gathered a vast amount of concepts, life cycle phase-dependent as well as spanning, and aligned them based on the structure described within the previous section. The resulting pattern is called Siemens Challenge Reference Model and is depicted in Figure 9 for one particular IIS used in technical services as example.

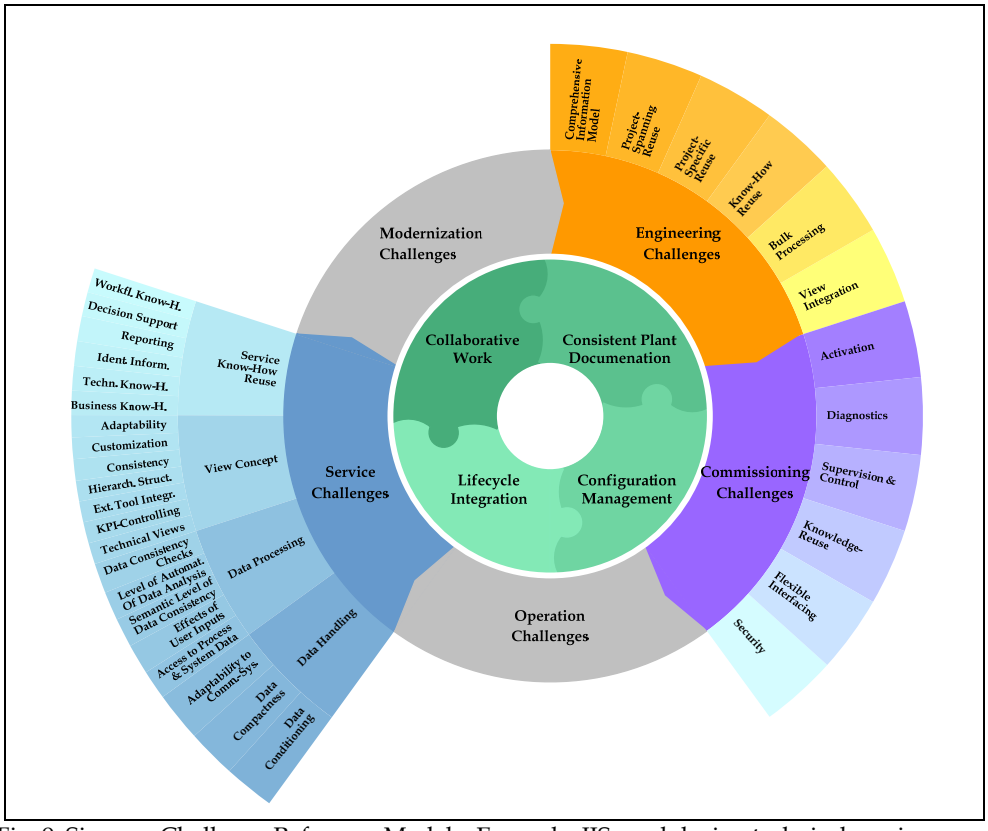


Fig. 9. Siemens Challenge Reference Model – Example: IIS used during technical service

The first type of Challenges is called Project Challenges and effects all life cycle phases of industrial project business (life cycle phase-spanning). Project Challenges are depicted within the center of the Siemens Challenge Reference Model shown in Figure 9. Yesterday's as well as today's IIS are mostly specialized in dealing with delimited tasks and therefore often support only sub-processes, for instance the execution of on-site but not remote maintenance. They often don't integrate with other IIS or information already gathered in preceding life cycle phases - for instance engineering data describing the production system's structure - which clearly offers room for improvement. Life cycle integration, collaboration, and configuration as well as continuous documentation are key success

factors aggregated in these Project Challenges that are valid for all IIS used in all production system life cycle phases.
The structure described within the previous section is reflected by making use of selected Challenges which in contrast to Project Challenges are life cycle phase-dependent. Figure 9 shows the applicable Challenges for an exemplary IIS which is setup during engineering as integral part of the production system. Naturally it is also element to the commissioning phase and is finally used continuously as part of technical services. The Challenges for the three corresponding phases engineering, commissioning, and service are expanded in Figure 9 and accordingly described below.
During engineering a lot of technical dependencies and risks have to be managed and experts from a great variety of crafts have to be coordinated, including for example electrical- and mechanical engineering. They are united by a superior task – namely the built of the production system – which was above referred to as project. Selected success factors during engineering are an comprehensive information model, intra- as well as inter-project reuse, and the reuse of engineering know-how, and bulk processing.
Commissioning the production system built during engineering usually is an strenuous task consisting of a step-by-step start-up of the production system and marking off long checklists of customer requirements. Selected and self-explanatory success factors are for instance an agile diagnostic of unexpected production system states, security as well as safety, and the re-use of commissioning knowledge.

	General Meaning of Class	**Example: Sub-Challenge External Tool Integration of Service Challenge View Concept**
Class 4	Concept for Generic Support of User	Plug-in-concept - access data on logic level (e.g. API, SOA)
Class 3	Concept for Explicit Support of User on a High Level	Standard file format - with automated im- / export
Class 2	Concept for Explicit Support of User on a Low Level	Standard file format
Class 1	Concept for Implicit Support of User	Access on DB-level
Class 0	No Concept Regarding Challenge	No support

Table 2. Concepts associated to Sub-Challenge Ext. Tool Integration of Challenge View Concept

Important activities of service execution are those summarized by the term maintenance. They cover inspection, monitoring, compliance test, function check-out, routine, overhaul, rebuilding, repair, fault diagnosis and localization, improvement, as well as modifications (see EN 13306, 2001). Maintenance for production systems is defined by (Ehrlenspiel et al, 2007) as *„all technical measures [necessary] to obtain or restore the functional state*" of an production system. Minimizing the cost accumulating during service, which may easily exceed a multiple of the initial costs of purchase, "*should be the primary goal of a cost-conscious developer*" (Ehrlenspiel et al,.2007). Challenges in service execution result mainly from the existence of an actual production system, for instance the need to respond to urgent

malfunctions and the production system itself as data source of the IIS. Figure 9 shows the four Challenges of service and additionally the subsidiary Sub-Challenges. For each of the 21 Sub-Challenges five concepts have been gathered which break down each success factor's determinants even further (example see Table 2).

5. Application of the Concept Catalog

The Siemens Challenge Reference Model serves as a basis for instruments of evaluation and improvement of Industrial Information Systems that can be used from the perspectives of both the IIS-supplier as well as the user of IIS. Therefore two complementary methods (see Figure 10) were developed by Siemens Corporate Technology (Dencovski et al., 2008) together with Friedrich-Alexander-University Erlangen-Nuremberg and Universität Stuttgart:

- IIS-Profile - General assessment and evaluation of the concepts (philosophy) of an Industrial Information System
- IIS-Usage-Profile - Analysis of the user workflow in relation to the Industrial Information Systems used, and findings derived from this (e.g. strengths and weaknesses of IIS, requirements).

The first method, which is the IIS-Profile-Analysis, serves as a general evaluation of the concepts of IIS, and enables the positioning of released Industrial Information Systems. It works also on their requirements specification, feature specification, prototypes as well as user's manuals relative to the Challenges described above. The evaluation is based on a generic usage context as a reference of evaluation, and shows principal possibilities and concepts of IIS – hence *IIS-Profile*.

The second method, which is the IIS-Usage-Profile Analysis, supports the concrete development of an IIS-Usage-Profile. It is the goal to evaluate the effectiveness of IIS in their concrete usage context, that is, the user's processes, by means of a strengths / weaknesses analysis, and to derive from this evaluation well-founded, structured requirements for the IIS used or to be used. The basis for the method is, on the one hand, a standardized list of questions, and on the other hand, a workflow model that is created interactively with the users. In this process, the roles and concrete tasks (including expenditures), the Industrial Information Systems used in the process, the processed / generated technical information, and its logical dependencies in the workflow of a production system project are considered. This also enables the method to place the focus on processes and dependencies that overlap IIS. The workflow model has multiple tasks to accomplish (see Figure 11):

- Visualization and documentation of interrelationships between tasks, information and Industrial Information Systems for users and the generation of questions in the interview
- Demonstrations object: Problems with the use of the Industrial Information Systems become evident in the workflow model, especially with regards to existing interrelationships
- Basis for the processing of concrete, common visions of Industrial Information Systems and systems landscapes; concepts can be clearly positioned, organized, demonstrated, and evaluated using the workflow model.

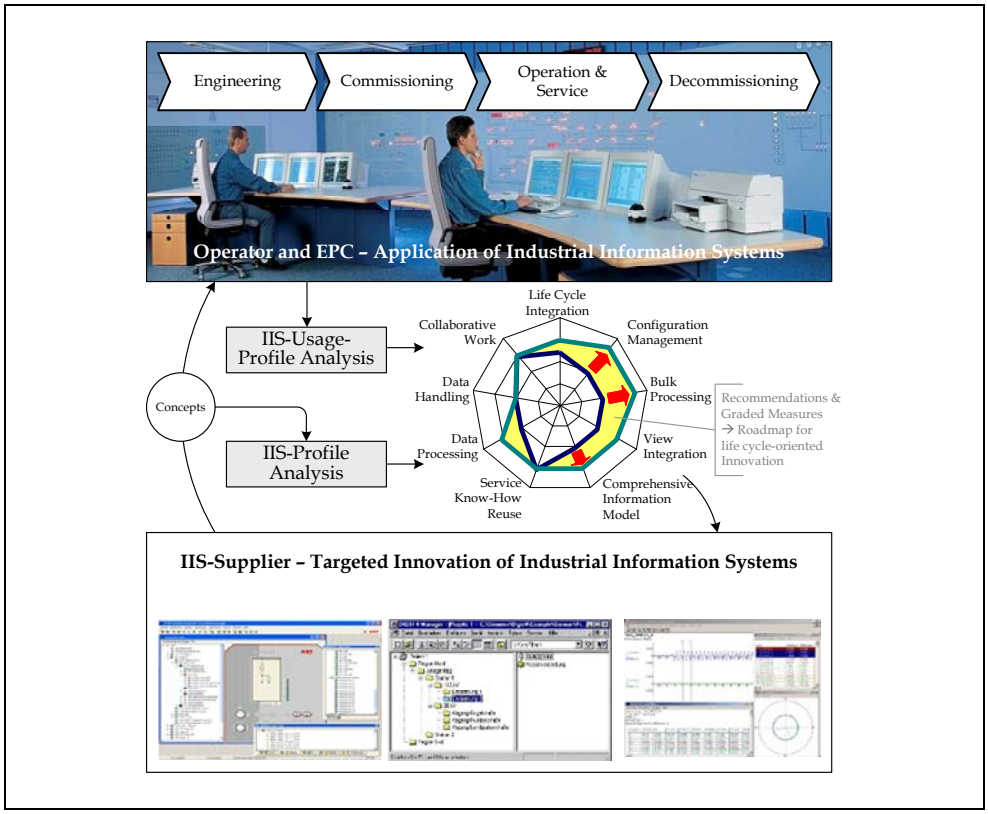


Fig. 10. Complementary methods for use of Siemens Challenge Reference Model

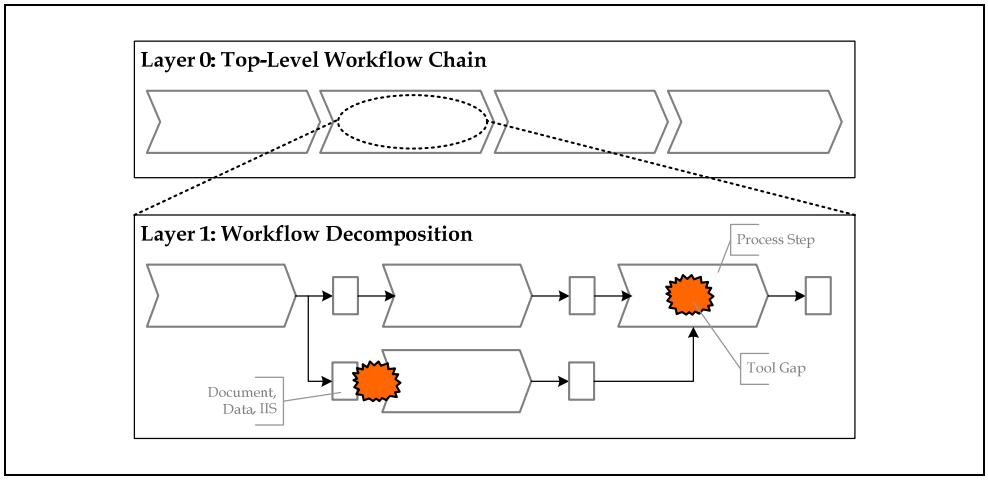


Fig. 11. Modeling of user workflows

Subsequently, using a standard list of questions and an interactively created workflow model, an IIS-Usage-Profile is created that contains existing strengths and weaknesses, as well as the expectations of the user for future concepts. The Challenges serve as an structuring perspective during workshops with users for a systematic examination of all important aspects. Existing Best Practices from the concept catalog are used, in order to put the requirements and expectations of the users into concrete terms, to illustrate them to the users; and possibly to limit them. For example, it is not meaningful to strive for the highest level of Best Practices when it is not required by the user's task, because with an increasing degree of freedom and number of possible settings, there is a simultaneous increase of time and energy needed for learning the ropes of the IIS.
The described two methods complement each other in order, on the one hand, to highlight principal (existing or planned) possibilities and concepts of IIS, and on the other hand, building on those, on the basis of developed IIS-Usage-Profiles and expectations, to develop life cycle-oriented innovation steps and roadmaps based on the idea of the Digital Factory. The results are summarized compactly at the highest level in a clearly arranged graphical representation. Individual levels represent how comprehensively the Industrial Information System supports the user. Through comparison with the current IIS-Profile, the appropriate adjustments for increasing the productivity can be identified and corresponding optimization measures can be derived. Figure 10 shows the combination of the two methods.
In recent years, more than 15 Industrial Information Systems were classified by Siemens Corporate Technology with the IIS-Profile method. Among these were commercially available tools, such as Freelance 800F from ABB, but also, tools of Siemens AG like Comos® (see for instance Maurmaier et al., 2008) or SIMATIC PCS 7. It can be seen, that IIS-suppliers follow different strategies in addressing Challenges in production system projects via concepts.
Further on, for several automation solution suppliers an IIS-Usage-Profile analysis was executed to model and visualize the actual user's workflow in relation to the IIS used, and to show strengths, weaknesses and improvement potentials of these and as well user expectations towards Industrial Information Systems.

6. Conclusion

Changed basic conditions and trends, postulated as such across various industries, demand from suppliers of automation solutions for production systems like Siemens AG to choose new paths in order to reduce the total cost of ownership and to increase the total value of production systems. The most important are

- to establish efficient integrated engineering processes and methods based on mechatronic components
- to provide standardized solutions for production systems (reusable work results) including adequate libraries
- to offer an integrated view on the production system over all life cycle phases and crafts

based on efficient and integrated Industrial Information Systems.
These new paths are combined within the idea of the Digital Factory, where for instance the production system is engineered entirely in a virtual way first, before the real production

system is realized. According to Siemens AG, integrated engineering for mechanics, electrics and automation will be available in ten to 15 years not only but especially for automotive industry.

In addition, the continuous and integrated provision of digital information over the whole production system life cycle based on mechatronic components enables to consider dependencies between various life cycle phases in advance. For instance, maintenance and condition monitoring activities for the operation and service phase of a production system can already be prepared during the engineering phase in a virtual way (Wucherer, 2006).

As an overall result it can be stated that suitable and integrated Industrial Information Systems, which are designed to realize the idea of the Digital Factory are crucial for the future success of all stakeholders of production systems. As a consequence, Industrial Information Systems have to be innovated in an adequate way. The method for the life cycle-oriented innovation of Industrial Information Systems introduced in this chapter allows for a targeted evaluation and innovation of IIS with the aim to realize the ideas of the Digital Factory in an evolutionary way.

The core of the presented method is built by the Siemens Challenge Reference Model, which operationalizes the knowledge on concepts used to cope with the challenges in industrial project business. This reference model integrates both the knowledge on life cycle phase-specific concepts and life cycle- and craft-spanning dependencies. With the help of the IIS-Profile-Analysis and IIS-Usage-Profile-Analysis which both use the Siemens Challenge Reference Model, IIS-suppliers as well as users of Industrial Information Systems have essential benefits, which are for instance

- status quo analyses of Industrial Information Systems as a basis for life cycle-oriented innovation
- competitor analyses, i.e. comparison of different Industrial Information Systems of various IIS-suppliers in order to generate unique selling points
- basis for selection and purchasing activities for users

On the road to realizing the Digital Factory, the stepwise innovation of Industrial Information Systems according to the introduced innovation method along the life cycle of a production system generates short and mid term benefits for all stakeholders of production systems. For the engineering and commissioning phase, following benefits can be described, for instance

- reduced engineering and reduced time-to-operation
- easy overview of complete project
- high quality of technical information
- flexible concurrent engineering
- reduced technical risks
- reduced commissioning time and costs
- reduced engineering and commissioning effort by standardized and compatible technical solutions
- faster start up by better diagnostic data of equipment and reduced effort for failure and malfunction fixing.

In addition, there are several benefits for the operation and service phase, too, for instance

- availability of engineering and diagnostic data for production system elements
- reduced production system down-times applying intelligent maintenance strategies

- fast failure fixing
- access to all relevant production system data for decision making.

Due to the fact that the introduced method already considers and describes life cycle- and craft-spanning dependencies of production systems within the Siemens Challenge Reference Model and the complementary described methods of a IIS-Profile-Analysis and IIS-Usage-Profile-Analysis, it can finally be stated, that a stepwise innovation of Industrial Information Systems according to this presented approach enables a stepwise realization of the vision of a Digital Factory.

7. References

Abel, D.; Allgöwer, F.; Bretthauer, G.; Bruns; Isermann, R.; Konigorski, U.; Premauer, T.; Schaudel, D.; Spohr, G.-U.; Steusloff, H.; Terwiesch, P.; Westerkamp, D.; Wucherer, K. & Zühlke, D. (2003). *AT 2010 - Thesen der Task-Force „Automatisierungstechnik 2010" der VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA)*, VDE Verlag, http://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/automatisierungstechnik2010.pdf, accessed 2009-05-06

Abran, A. & Buglione, L. (2000). QF²D: A Different Way to Measure Software Quality, *Proceedings of the 10th International Workshop on Software Measurement (IWSM2000)*, ISBN: 3-540-41727-3, pp. 205-219, Berlin, Germany, Springer Verlag, Berlin

Amberg, M.; Holm, T. & Dencovski, K. (2008). Business Challenge Based Evaluations - Using Methods of Social Research to Gather Quality Characteristics and Increase Software Quality, *Proceedings of the International Conference on Innovation in Software Engineering (ISE'2008)*, ISBN: 978-1-7408829-8-9, Vienna, Austria

Balzert, H. (1998). *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*, ISBN: 978-3827400659, Spektrum Akademischer Verlag, Heidelberg Berlin, Germany

Boehm, B. W.; Brown, J. R. & Lipow, M. (1976). Quantitative evaluation of software quality, *Proceedings of the 2nd International Conference on Software Engineering*, pp. 592-605, Catalog No. 76CH1125-4C, San Francisco, USA, IEEE Computer Society Press, Los Alamitos, USA

Dencovski, K.; Wagner, T. & Schroedel, O. (2008). Produktivitäts-Check für Engineering-Software, *Elektrotechnik & Automation (etz)*, Vol. 129, No. 1, p. 56, ISSN: 0948-7387

Ehrlenspiel, K.; Kiewert, A. & Lindemann, U (2007). *Cost Efficient Design*, ISBN: 978-3-540-34647-0, Springer Verlag, Berlin, Germany

EN 13306 (2001). European Standard: Maintenance terminology, ISBN: 0580377121, British Standards Institute

Fay, A.; Schleipen, M. & Mühlhause, M. (2009). How can we systematically improve the engineering process?, *atp – Automatisierungstechnische Praxis*, Vol. 51, No. 1-2, pp. 80-85, ISSN 0178-2320

Green, D.; DiCaterino, A. (1993). *A Survey of System Development Process Models*, Work Report Center for Technology in Government, University at Albany, http://www.ctg.albany.edu/publications/reports/survey_of_sysdev/survey_of_sysdev.pdf, accessed 2009-05-06

Hiesinger, H. (2008). *Siemens – Answers for Industry*, Hannover Messe 2008, Hannover Germany, http://www.automation.siemens.com/wwwdocs/nc_folien/f_hiesinger .pdf, accessed 2009-06-05

ISO/IEC 9126-1 (2001). *Software engineering-software product quality-part 1: Quality model*, International Organization for Standardization Geneva, Switzerland

ISO/IEC 9126-4 (2001). *Software engineering-software product quality-part 4: Quality in Use metrics,* International Organization for Standardization Geneva, Switzerland

ISO/IEC 14598-1 (1999). *Information technology – Software product evaluation - Part 1: General Overview*, International Organization for Standardization Geneva, Switzerland

Kiefer, J.; Baer, T. & Bley, H. (2006). Mechatronic-oriented Engineering of Manufacturing Systems - Taking the Example of the Body Shop (Daimler AG), *Proceedings of the 13th CIRP International Conference on Life Cycle Engineering*, pp. 681-686, ISBN: 90-5682-712-X, Leuven, May 2006

Kitchenham, B. (1990). *Software metrics*, Lecture Notes, Part I & II

Löwen, U.; Bertsch, R.; Böhm, B.; Prummer, S. & Tetzner, T. (2005) Systematization of industrial plant engineering, *atp - Automatisierungstechnische Praxis*, Vol. 47, No. 4, pp. 54-61, ISSN 0178-2320

Maurmaier, M.; Dencovski, K. & Schmitz, E. (2008) Engineering Challenges - Evaluation Concept for Engineering Tools, *atp - Automatisierungstechnische Praxis*, Vol. 50, No. 1, pp. 50-58, ISSN 0178-2320

McCall, J.; Richards, P. & Walters, G. (1977). *Factors in software quality*, Vol. I, II, and III, US. Rome Air Development Center Reports NTIS AD/A-049 014, NTIS AD/A-049 015 and NTIS AD/A-049 016, U. S. Department of Commerce

McConnell, S. (1993). *Code Complete (First ed.)*, Microsoft Press, ISBN: 1-55615-484-4, Redmond

Preiss, K.; Patterson, R. & Field, M. (2001). The Future Directions of Industrial Enterprises, In: *Maynard's Industrial Engineering Handbook*, Kjell B. Zandin (Ed.), pp. 133-161, McGraw-Hill, ISBN 978-0-0041102-9

Royce, W. (1970). Managing the Development of Large Software Systems, *Proceedings of IEEE WESCON 26*, pp. 328-339, August 1970, TRW

Schott, T. (2007). *Die Automobilindustrie – Eine Schlüsselindustrie für die Anwendung von Antriebs- und Automatisierungstechnik*, SPS/IPC/Drives 2008, Nuremberg, http://www.automation.siemens.com/_de/portal/news/speeches_detail.htm?rss ItemURL=/detail_rss.php3?template_id=158&id=8450, accessed 2009-05-06

Tayeh, M. (2009). Harnessing IT Solutions for Global Operations (Keynote Address), *daratechPLANT2009, Houston, USA*

VDI 2206 (2004). *VDI Richtlinie 2206 - Digital factory Fundamentals*, Verein Deutscher Ingenieure e. V. Düsseldorf, Germany

VDI 3695 (2009). *VDI Richtlinie 3695 - Engineering of industrial plants - Evaluation and optimization - Fundamentals and procedure*, Verein Deutscher Ingenieure e. V. Düsseldorf, Germany

VDI 4499 (2008). *VDI Richtlinie 4499 - Design methodology for mechatronic systems*, Verein Deutscher Ingenieure e. V. Düsseldorf, Germany

Wucherer, K. (2006). Innovative Automation für eine zukunftsfähige Produktion., In: *Jahrbuch Elektrotechnik 2007*, Grütz, A. (Ed.), pp. 13-18, VDE Verlag, ISBN 978-3-8007-2943-2

Yin, R (1994). *Case Study Research*, Sage Publications, ISBN: 978-0761925538, Beverly Hills, USA

20

Asynchronous Analogue-to-Digital Conversion Techniques

Nikos Petrellis, Michael Birbas, John Kikidis and Alex Birbas
Analogies S.A., Patras Science Park, Platani-Rio, 26504 Patras Greece

1. Introduction

The sensor values readout is a critical issue in Factory Automation systems and control applications in general. Analogue-to-Digital Converters (ADCs) are the circuits that accept as input the analogue indication of the sensors and provide as output the corresponding digital code representation of the analogue value that can be exploited by the digital part of the controller. The most important parameters that affect the selection of an appropriate ADC are the conversion resolution, the sampling rate, the power consumption and the die area required. Several architectures have been proposed for ADCs that target different application requirements. For example, cable TV and PAL/NTSC decoders require 8-12bits resolution with 10-20MS/s sampling rate while high speed logic analyzers require 5-7bits resolution with multi-GS/s sampling rate. The industrial systems that require ADC conversion are production lines, command-control facilities, product quality measurement, communication networks, security systems etc. Industrial systems may use ADCs for the control of simple sensors like temperature, humidity, pressure, etc as well as for the interface of input devices with increased complexity like high speed and precision imagers.
The most popular ADC architectures are the Flash, Pipeline, Successive Approximation, Sigma-Delta and Folding-Interpolation ones. The Flash ADCs achieve the highest speed but occupy large silicon area and consume high power. For this reason, their resolution is practically limited to less than 8-bits. Furthermore, special analogue front-end components or technology processes need to be employed to design Flash ADCs that achieve multi-GS/s sampling rates. The Pipeline or Subrange ADCs consist of two or more ADCs with smaller resolution that operate on successive inputs in a pipelined manner and generate different groups of output bits. The throughput of pipeline ADCs is comparable to that of the Flash ADCs, the die area and power consumption are lower but the latency of a single sample is much longer. Counting and Successive Approximation ADCs consist of a digital counter that feeds a Digital-to-Analogue Converter (DAC) with increasing values until the analogue output of the DAC gets higher than the input. These ADCs require a very small number of components but need a variable large number of clock periods to reach a decision. Folding-Interpolating ADCs use a small number of comparators that compare the input successively with different sets of reference values. Finally, Sigma Delta ADCs are based on input oversampling. Parallel ADCs consist of multiple slower ADCs that are interleaved in time.

Within this context, we have developed an ADC architecture (patent pending) that extends the concept of pipeline ADCs through the employment of a binary tree structure that leads to the implementation of fast ADCs that occupy very small die area and have configurable resolution for achieving lower power consumption. The specific ADC architecture is based on the integer division of an analogue input by an appropriate power of 2. Specifically, a novel circuit has been designed that accepts as input an analogue value and generates the quotient and the residue of the integer division. Based on this principle, a Combo 4/8/12-bit ADC has been developed that operates in current mode in order to implement the functions of addition, subtraction, multiplication/division by a constant with high speed simple circuits that need low power supply. The required die area is only 0.12mm^2, the power consumption is 72mW and the average sampling rate exceeds 140MS/s for 12-bit resolution. No clock signal is required by that ADC due to its asynchronous nature. Thus, the ADC input can be connected either to a sample and hold circuit or directly to the input analogue signal. The ADC output can be latched using an independent clock.
The linearity errors are a major problem in any Analogue-to-Digital Conversion method. Several methods like trimming, real time calibration, generation of additional bits for error correction etc, have been employed for the improvement of the real time behaviour of an ADC. Beside these popular approaches, the techniques that have been used to enhance the linearity and the transistor mismatch problems in the proposed binary tree ADC architecture are furthermore discussed.
The most important quality metrics of an ADC are briefly presented in Section 2. The popular ADC architectures that were mentioned above are described in more detail in Section 3. The architecture of the proposed asynchronous ADC with binary tree structure along with implementation details are presented in Section 4. Finally, simulation results and a comparison with other ADCs are presented in Section 5.

2. ADC quality metrics

The most important parameters of an ADC that are taken into consideration in research papers as well as commercial products are the resolution, the speed, the power consumption, the various error metrics and the area required. The die area requirements of an ADC IP component are directly determined by the designed layout or the component count. Commercial ADC chips are characterized by the chip size, pin count etc, instead of just the die area. The average and maximum power consumption can be determined either in DC or preferably in AC operation/simulation.
The resolution determines the conversion step of the analogue input i.e., the input voltage change needed for triggering a corresponding change in the Least Significant Bit (LSB) of the ADC. If the input signal ranges between 0 and V_{ref}, then an ADC with 4-bit resolution has a conversion step of $V_{ref}/16$ while this step is $V_{ref}/256$ in 8-bit resolution. It is obvious that the higher the resolution is, the less distortion is introduced by the analogue to digital conversion mechanism.
The speed of an ADC can be expressed by the sampling rate, the conversion latency and the analogue input bandwidth. The maximum sampling rate (throughput) determines how fast an input signal can be sampled in order to avoid missing transitions. Again, the higher sampling frequency allowed, the less distortion of the analogue input is achieved. The sampling rate is not always proportional to the maximum frequency allowed for the

analogue input signal. Regarding the analogue input signal bandwidth this is not always clearly defined. Some authors mention that this is associated with the input signal frequency used for their measurements but do not specify whether this is an analogue signal that swings in the full input range, or an input that is appropriate for Nyquist compatible sampling. The Signal to Noise and Distortion Ratio (SNDR) and the Spurious Free Dynamic Range (SFDR) are two parameters that can provide an indication on how close the output of an ideal DAC that is connected to the ADC output is, to the original input.

The latency is the time needed for the conversion of a single input sample. The conversion time determined by the throughput is usually shorter or in the worst case equal to the latency, since multiple samples may be processed concurrently as in the case of pipelined ADCs.

Several parameters have been defined for the description of possible errors in an ADC operation. An ideal conversion is plotted with the solid line in Fig. 1, along with a conversion characterized from offset, gain and monotonic errors (dotted line). Offset errors shift left or right the ADC output while the taps' height is erroneous on the presence of gain errors. A missing code appears in Fig. 1 due to a gain error. Differential Non-Linearity (DNL) indicates how far the current code is from its previous one and can be expressed as

$$DNL_i = \frac{D_i - LSB}{LSB} \tag{1}$$

The DNL_i is the DNL error of a specific code *i*, D_i is the actual duration of this code, while LSB is the ideal duration of this code. The Incremental Non-Linearity (INL) is defined as the integral of the DNL errors of all the output codes. The INL indicates how far the ideal ADC transfer function is from the measured one. The Signal to Noise Ratio (SNR) of an ADC is defined as the ratio of the input signal energy to the noise energy from DC to the Nyquist frequency. Another error source is the temperature dependence of an ADC. Although the specifications of an ADC guarantee that no missing codes occur within a temperature range, the linearity as well as the gain and offset errors may depend on temperature variations too.

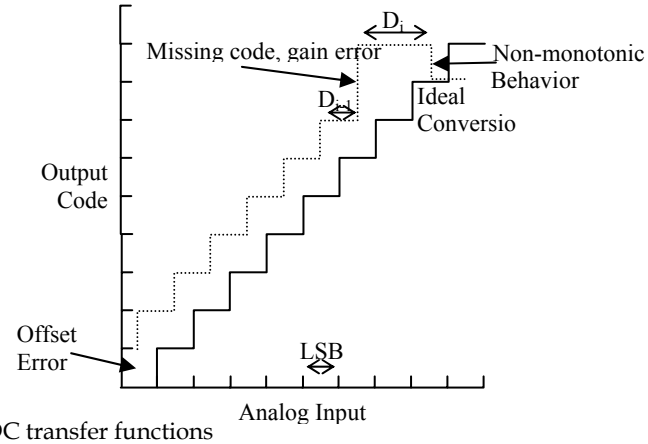


Fig. 1. Example ADC transfer functions

The ADC resolution and speed requirements for various applications are summarized in Table 1.

Application	Resolution	Speed
High Definition Data Analyzers	>16bits	>60MS/s
High Definition Imagers	12bits	80MS/s
Cable TV	8-12bits	15MS/s
High Definition TV	8-10bits	75MS/s
Magnetic Storage Read	8bits	150MS/s

Table 1. Indicative ADC resolution and speed requirements

3. Popular ADC architectures

3.1 Flash ADCs

Several Voltage-mode Flash ADCs have been proposed in the literature (Nejime et al., 1991) (Walden et al., 1990). An n-bit Flash ADC compares the input with 2^{n-1} voltage levels as shown in Fig. 2. These levels are: $V_{ref}/2^{n-1}$, 2 $V_{ref}/2^{n-1}$,…, $(2^{n-1}-1)$ $V_{ref}/2^{n-1}$ and are generated by a resistor ladder consisting of identical resistors. The input voltage (V_{in}) range is 0..V_{ref} and can be adjusted by connecting V_{ref} to an appropriate voltage. Current Mode Flash ADCs have been recently presented in the literature (Bhat et al., 2004) (Masood Ali et al., 2005). In this case, the input current is compared to 2^{n-1} current levels generated by appropriately sized current mirrors. The voltage comparators of Fig. 1 are replaced with current comparators in this case.

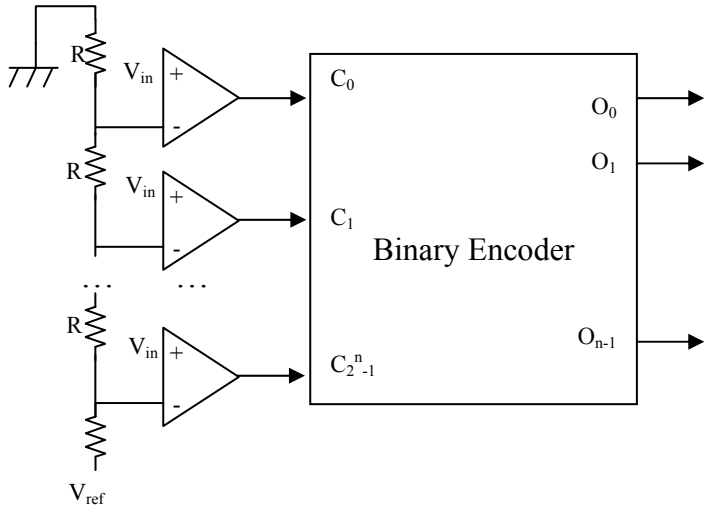


Fig. 2. Flash ADC architecture

The 2^{n-1} comparators of an n-bit Flash ADC, generate a "temperature code" that should be encoded into a binary representation. The relations that generate the binary representation $O_{n-1}O_{n-2}…O_0$ from the 2^n comparator outputs: $C_{2^n-1}C_{2^n-2}…C_0$ are the following:

$$O_n = C_{2^n-1} \tag{2}$$

$$O_{n-1} = C_{2^n-1} \oplus C_{2^{n-1}-1} \tag{3}$$

$$O_{n-2} = C_{4(2^n/4)} \oplus C_{3(2^n/4)} + C_{2(2^n/4)} \oplus C_{2^n/4} \tag{4}$$

…

$$O_0 = C_{2^n-1} \oplus C_{2^n-2} + \ldots + C_1 \oplus C_0 \tag{5}$$

The binary encoder can be implemented with domino XOR gates (Liu et al., 2003) since half of the XOR operations required by O_i are also needed in the estimation of O_{i-1}.
The source of non-linearity errors in the Flash ADC converters are the resistor mismatches in the ladder that generates the comparator voltage references. Contemporary Flash ADCs exploit special analogue techniques and process technologies to achieve multi GS/s throughput with good linearity behaviour. In (Walden et al., 1990) the authors used focused ion beam to create transistors with adjustable thresholds. In this way the resistor ladder is eliminated since the comparator thresholds are determined by the individually calibrated transistor thresholds. In (Chan et al., 2007) a submicron InP HBT technology is employed to achieve 5GS/s with 7-bit resolution. In (Park et al., 2006) inductors are used to improve speed and comparator pre-amplification bandwidth along with comparator kickback reduction techniques. The authors in (Deguchi et al., 2008) developed a 6-bit 3.5GS/s Flash ADC in 90nm CMOS technology using clamp diodes.
A Flash ADC can be implemented with asynchronous circuits. In this case, the digital outputs are settled after a period of time that depends on the comparators used and the binary encoder speed. The ADC outputs can be latched by a clock that is conformant with this settling time. The Flash architecture is the fastest one but the die area and power consumption required gets too high if the resolution is more than 7-bits due to the large number of comparators that are required.

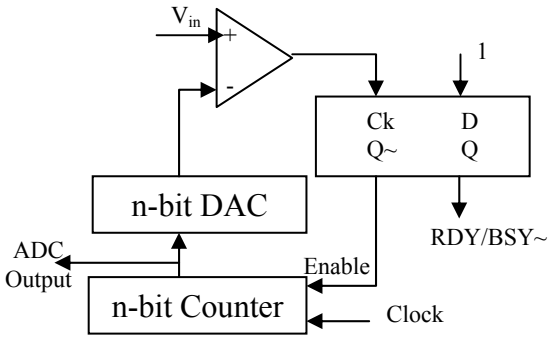


Fig. 3. Counting ADC architecture

3.2 Counting and Successive Approximation ADCs

The architecture of a Counting ADC appears in Fig. 3. A fast digital counter generates the successive digital representations of the numbers 0, 1, 2,…, 2^n-1. A Digital-to-Analogue

Converter produces an analogue value from this digital representation that is compared to the analogue input. If the digital counter output is found to be higher than the input, the ADC conversion operation is terminated. The average convergence time of this synchronous architecture is $2^{n-1}T_{ck}$, where T_{ck} is the period of the internal clock. This variable convergence time is too long but the number of components is quite small leading to slow but very low power and area ADCs. The Successive Approximation ADCs (Yuan & Svensson, 1994) (Promitzer, 2001) are an improvement to the Counting ones achieving $T_{ck}\log_2 2^n = nT_{ck}$ convergence time. Instead of a free running n-bit Counter they apply to the n-bit DAC values derived from an interpolation search. Initially the value 2^{n-1} is applied. If for example, the input value is smaller, the range $(2^{n-1}..2^n)$ is rejected and the value 2^{n-2} is applied to the DAC. Assuming that the input value is found higher, the range $[0..2^{n-2}]$ is rejected and a search in the range $(2^{n-2}..2^{n-1})$ is initiated. This procedure is repeated until the input value is approached. This algorithm is similar to a search in a binary tree. The linearity errors stem from the potential linearity errors of the DAC that is used.

3.3 Pipeline ADCs

The Pipeline or Subrange ADCs (Ahmed & Johns,2005) (Iizuka et al.,2006) achieve conversion speeds comparable to that of the Flash ADCs with significantly lower power consumption and die area. The principle of their operation is described by Fig. 4. The analogue input is used by a coarse Flash ADC to generate the most significant bits. The output of the coarse ADC is input to a DAC. The DAC's output is subtracted from the original analogue input and the difference is input to a fine Flash ADC that generates the least significant bits. Assuming that the resolution of both the coarse and the fine ADC of the n-bit Pipeline ADC are identical (n/2), then the required number of comparators is $2^{n/2}+2^{n/2}$ instead of 2^n that are required by a Flash ADC with n-bits resolution. The two constituent ADCs operate on successive samples held by the Track and Hold (T/H) circuits connected at their inputs. The Pipeline ADCs are synchronous systems since the T/H circuits of each stage require a clock signal.

The concept of the described Two-Stage Pipeline ADCs can be extended to more than two stages leading to further reduction of the required components. In the extreme case, each stage produces a single bit. In many approaches, each ADC stage generates redundant bits for error correction and calibration. For example if a 10-bit pipelined ADC has two stages, the first stage can generate 6-bits and the second 5-bits. The least significant bit of the first stage should match the most significant bit of the second stage. In the case they do not match, a calibration procedure can be initiated that modifies e.g. the references of the comparators in each ADC stage or the DAC biasing.

3.4 Other Common ADC Architectures

Folding and Interpolating ADCs (Makigawa et al, 2006) consist of a relatively small number of comparators. Different groups of reference values are applied to these comparators within a sampling period. In this way, the group that includes the closest reference to the analogue input is located using a fast interpolation algorithm.

The Algorithmic or Cyclic ADCs (Hedayati, 2004) (Chen & Wu,1998) are similar to the pipeline ADCs with 1-bit stages. Nevertheless, they consist of a single stage that is repeatedly used for the generation of each bit of the output.

Another popular architecture is the Sigma-Delta ADC (Arias et al., 2006) that is based on oversampling of the analogue input. The sampling frequency is much higher than the Nyquist limit and modulates the analogue input within the sampling period. The pulse density of the modulated input is then digitally filtered removing the noise components in the frequency domain. Sigma-Delta ADCs are implemented with integrators and 1-bit DACs. Due to the oversampling principle, the input signal frequency should be much lower (up to a few MHz) than the sampling frequency. The Sigma-Delta ADCs can easily achieve a resolution higher than 16-bits.

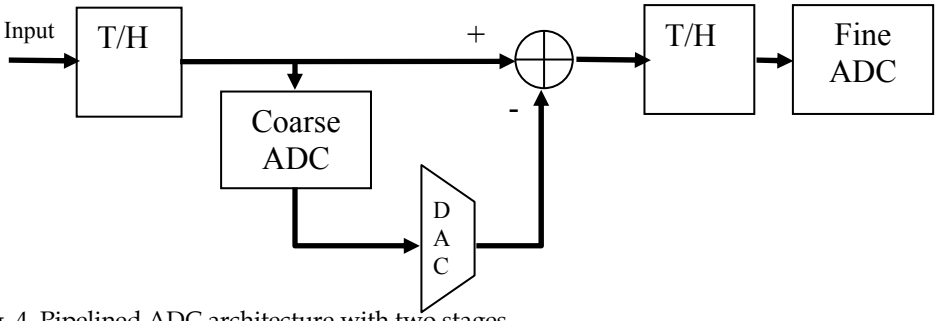


Fig. 4. Pipelined ADC architecture with two stages

Multi GS/s ADCs can be constructed with multiple parallel ADCs that are time interleaved i.e. they sample the input with a phase shift of a few psec. The individual ADCs are slower, robust circuits that dissipate low power. For example, in (Poulton et al., 2003) a parallel ADC with 80 slices is presented achieving 20GS/s sampling rate on an input analogue signal bandwidth of 6GHz. Each slice is a current mode pipelined ADC that achieves a 250MS/s throughput, occupies only 0.12mm^2 area and dissipates 57mW power. The clocking system creates 80 clocks with 250MHz frequency. The clocks used by neighbouring slices have a 50psec phase shift.

4. An Asynchronous ADC with Binary Tree Structure

4.1 General Architecture

An Asynchronous ADC architecture that requires a very small number of components and achieves high throughput is proposed in this section. The general architecture of this ADC is shown in Fig. 5. The analogue input value is divided in the root node by an appropriate power of 2. The quotient and the residue of this integer division are the inputs of two subtrees that correspond to ADCs with half resolution of the overall ADC. The integer division performed at a node of the balanced binary tree presented in Fig. 5 is 2^{2^L}, where L is the level of that specific node in the tree. The leaves are assigned to L=0. For example, the root node in an 8-bit ADC tree divides the input by 16. The residue and the quotient of this division is input to two subtrees that correspond to 4-bit ADCs. The root nodes of these 4-bit ADCs divide their input by 4. The quotient and residue of this division are input to 2-bit ADCs. Finally, these 2-bit ADCs divide their input by 2 and generate a quotient and a residue that are digitised by a comparison to a threshold.

The binary tree does not necessarily need to be balanced. For example, a 12-bit ADC has been developed with a 4-bit ADC and an 8-bit ADC that are connected to a root node. If the input of this 12-bit ADC is in the range [0..256], then the root node divides the input by 16. The quotient of this division is between 0 and 16 and is driven to the 4-bit ADC. The residue is between 0 and 15 and is driven to an 8-bit ADC with the same input range [0..256] after a multiplication by 16. Similar range adaptations may be decided during the design and simulation of the actual circuit that implements the architecture of Fig. 5, for optimisation purposes.

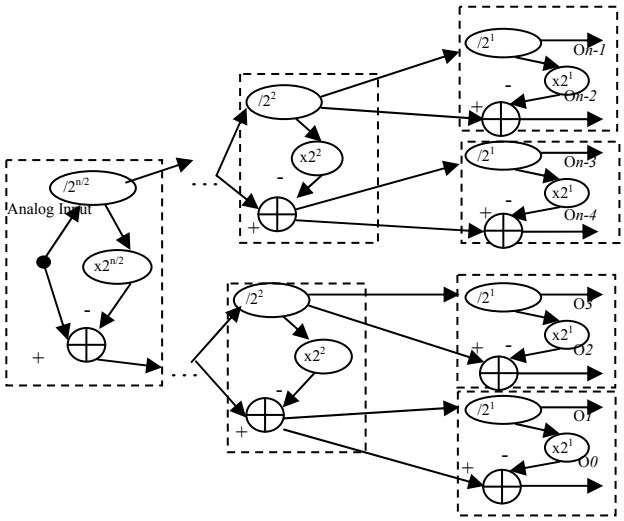


Fig. 5. Architecture of an n-bit ADC with binary tree structure

The architecture presented in Fig. 5 can be realised with current mode circuits due to the simplicity in the implementation of the various operators required, such as additions, subtractions and multiplication/division by a constant. Moreover, current mode circuits operate with low voltage supply and achieve high speed operation.
The integer division is performed by the circuit presented in Fig. 6. The input current signal I_{in} is concurrently compared against the references I_{ref}, $2I_{ref}$,…,$(N-1)I_{ref}$, where N is determined by the maximum input current allowed ($I_{max}=N \cdot I_{ref}$). If the input is between $q \cdot I_{ref}$ and $(q+1) \cdot I_{ref}$, then q comparator outputs will be high leading to the addition of q current sources at the output of the divider. The current $q \cdot I_{ref}$ represents the quotient of the division I_{in}/I_{ref}. The residue current I_r of this division is:

$$I_r = I_{in} - nI_{ref} \tag{6}$$

If N is too high then a large number of comparator and current sources are needed in the circuit of Fig. 6. In order to avoid the large area occupied by such a divider as well as its high power consumption, two simpler dividers can be connected in series as shown in Fig. 7. This is derived by the fact that a division by $N=N_2 \cdot N_1$ is equal to a division by N_1 and then, a division by N_2. The reference currents of the two dividers are selected as:

$$I_{ref1} = \frac{I_{\max}}{N_1} \tag{7}$$

$$I_{ref2} = \frac{I_{\max}}{N_1 N_2} \tag{8}$$

The area and power consumption are significantly reduced in this way, but an additional delay is introduced by the second stage of division. If

$$q = \frac{I_{in}}{N_1 N_2} \tag{9}$$

then the output of the circuit in Fig. 7 is $q \cdot I_{ref2}$.

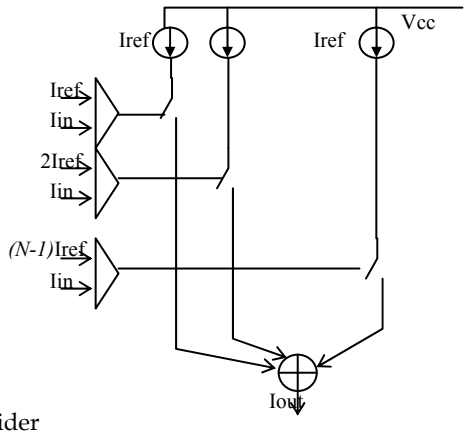


Fig. 6. The integer divider

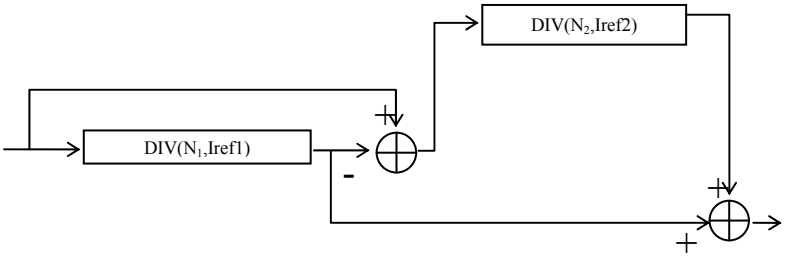


Fig. 7. Replacing a large divider by two simpler ones

4.2 Implementation Details

The integer divider of Fig. 6 can be implemented by using either synchronous or asynchronous current comparators. Synchronous comparators are assumed to be more

accurate and achieve faster convergence than the asynchronous ones. A part of the clock period is used in the synchronous comparators to reset the charged contacts of the transistors used while the comparator output convergence occurs during another part of the clock period. Nevertheless, although being the fastest and more accurate synchronous comparators are very sensitive to transistor mismatches. The mismatch immunity can be improved if cascode transistor arrangements are used, but the convergence speed is slowed down in this case. Another drawback of a synchronous comparator is the kickback effect i.e., glitches and jitter that appear to its inputs due to internal clock switching.
Asynchronous comparators offer high mismatch immunity with very low jitter. For this reason, an asynchronous comparator with positive feedback (Traff, 1992) has been employed for the implementation of the proposed divider that is presented in Fig. 6. The convergence time of this comparator ranges from 1ns to 100ns for input current difference between 10uA and 0.01uA respectively. Using asynchronous comparators, the whole ADC architecture does not need a clock signal. Consequently, if an analogue input is applied the outputs will settle in a short time period. The worst settling time observed either through simulation or experimentation can be considered as the latency period for the conversion of a single sample. Nevertheless, the outputs can be latched using a higher frequency clock if the transient output codes that are generated before the end of the ADC settling period do not break the monotonic principle. In this way, higher throughput and conversion accuracy can be achieved.

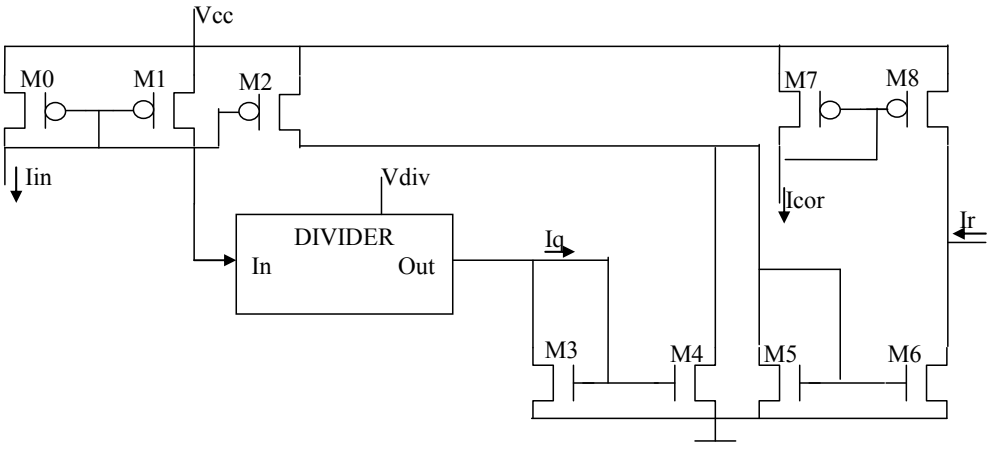
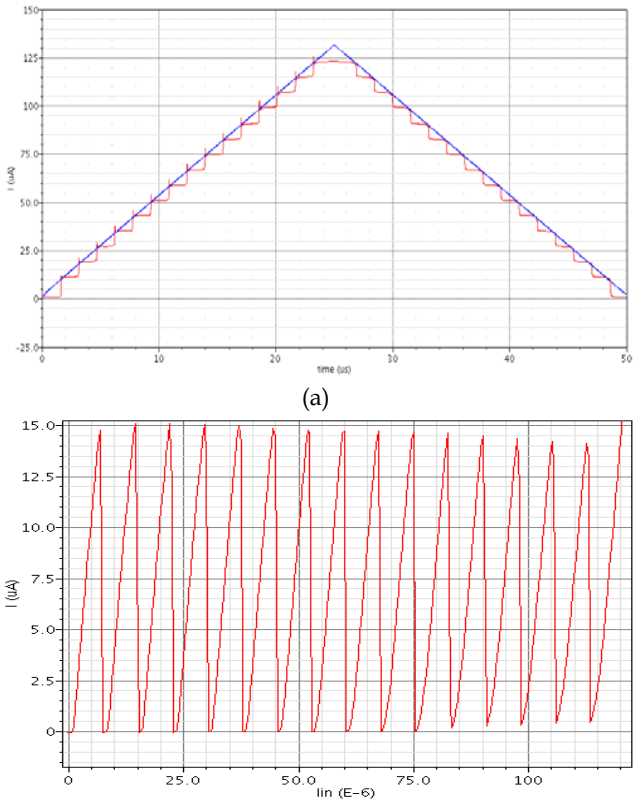


Fig. 8. Generation and offset correction of the integer division residue

The copies of the comparator input current and references as well as the current sources used in Fig. 6 are generated through cascode current mirroring. The channel length of the transistors used in mirrors that reproduce constant currents like the current sources or the comparator references is selected to have higher value (0.8um..1um) than the channel length of the transistors that are used in the mirrors of the input signal. The switches that are controlled by the comparator outputs can be implemented by NMOS or PMOS transistors or pass gates. If a switch is open, the corresponding I_{ref} current source should be preferably connected to the ground to reduce transient glitches.

The residue of the division is generated as shown in Fig. 8. In a real design environment, the simple mirrors used in this figure are replaced by cascode ones for higher mismatch immunity or gain boosted mirrors for achieving real time calibration. The PMOS mirror that consists of the transistors M0-M1-M2 generates two copies of the input current. One of them is used as input to the integer divider. The output of the integer divider (quotient) is mirrored in M4 and subtracted from the source current of M2 that generates the second copy of the input. The NMOS mirror M5-M6 accepts as input the residue of the subtraction. The potential offset I_{cor} that appears in the residue can be removed by the PMOS mirror M7-M8.

The output of the divide by 16 circuit used in the root of an 8-bit ADC is shown in Fig. 9a along with the divider input. The difference of these signals represents the residue and is shown in Fig. 9b.



(a)

(b)

Fig. 9. The input and output of the divider (a) and their difference (b)

The top level design of an 8-bit ADC in Cadence environment is presented in Fig. 10. The displayed current mirrors and the divider DIV16xIref implement the root node of the binary

tree (see Fig. 5). The 4-bit ADC blocks are driven by the quotient and the residue of the division by 16 implemented in the root node.

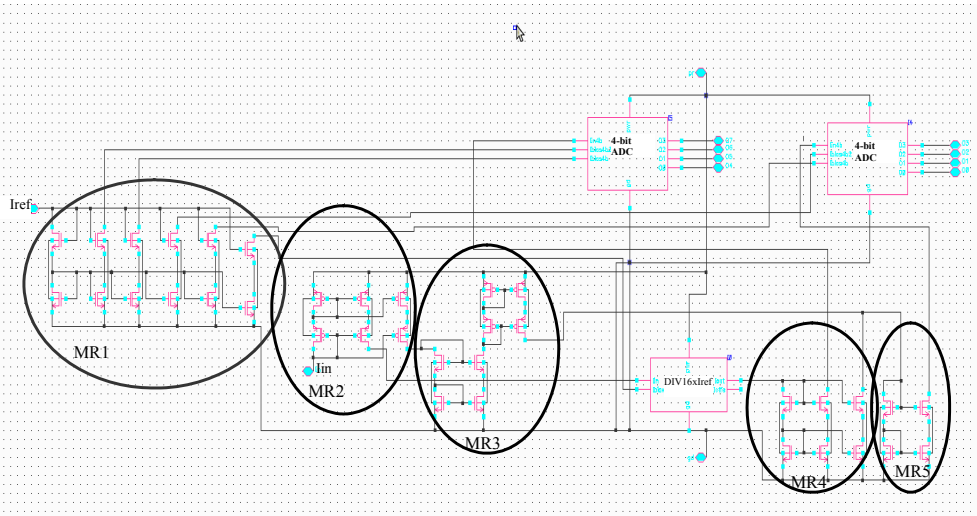


Fig. 10. Top level design of an 8-bit ADC

The mirror MR1 generates the various reference currents needed by the root divider and the 4-bit ADCs, while MR2 generates two copies of the input current. The divider output will be subtracted from one of the input copies. This specific input copy is delayed by the PMOS/NMOS mirror pair MR3 in order to compensate the delay introduced by the divider. The residue is driven into the 4-bit ADC that generates the least significant bits through MR5.

In a similar way, a 12-bit ADC has been developed by using a root node that divides the input by 16 and drives the quotient to a 4-bit ADC and the residue to an 8-bit ADC through a range adaptation circuit as already described in section 4.1. The output of such a 12-bit ADC is shown in Fig. 11.

A current mode ADC input is connected to the output of a Voltage to Current converter (V2I) since the analogue input is usually a voltage signal. A V2I circuit is characterized by its linearity. This can be expressed as the ratio of the input voltage to the output current and should remain as close to a constant as possible throughout the whole input voltage range.

A Sample and Hold (S/H) circuit may also be necessary in order to hold the ADC input stable for as long as the conversion takes place. A voltage S/H circuit should be placed between the analogue input and the V2I converter while a current S/H should be placed between the V2I and the ADC.

Half of the clock period in an S/H circuit is dedicated for the sampling of the analogue input (Sample period) while its output is kept stable during the second half clock period (Hold period). The quality of an S/H circuit depends on how stable the S/H output is during the Hold time (e.g., free of jitter, leakage etc) for the desired Hold duration. In an asynchronous ADC the sampling period is an idle time. For this reason, a pair of S/H

circuits that use complementary clocks are used. In this case, one of the two S/H circuits is always in its hold period continuously feeding the ADC with valid samples.

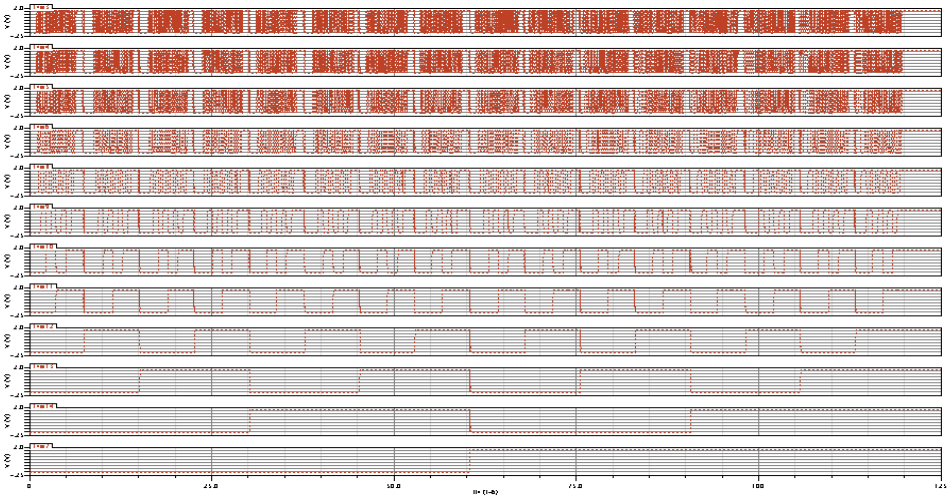


Fig. 11. The output of a 12-bit ADC

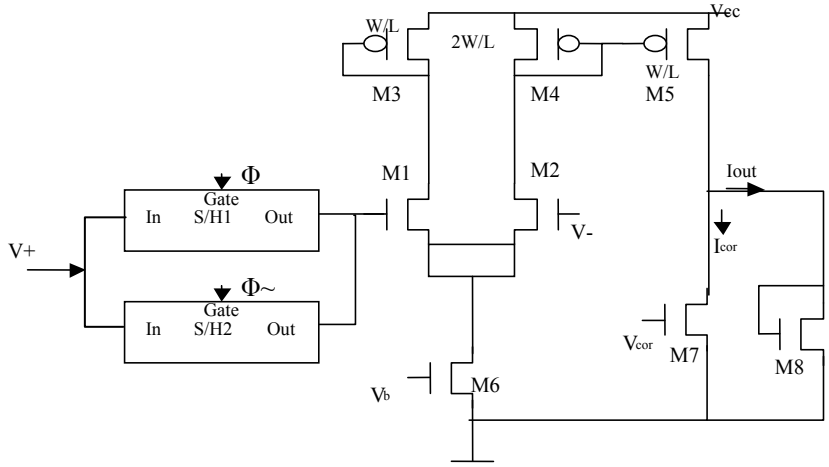


Fig. 12. S/H and V2I architecture for the developed asynchronous ADC

An appropriate V2I and S/H topology for the developed asynchronous ADC is presented in Fig. 12. The inputs of the two S/H circuits are connected to the positive pole of the analogue signal source. The outputs of the two S/H circuits are connected together to one of the differential inputs of the V2I (M1 gate). The second differential input (M2 gate) is connected to the negative pole of the analogue input source. The V2I consists of the transistors M1-M6. M1 and M2 should be identical while the size of M3-M5 should follow the designated in Fig l2 channel dimensions. The linear region of the V2I operation may not start from 0. The

undesirable offset current can be removed by biasing properly the drain current of M7 during the calibration of the ADC.

5. Simulation Results and ADC Comparison

The developed 12-bit ADC incorporates an 8-bit and a 4-bit ADC that can be isolated by powering off the rest of the circuitry in applications that favour lower power dissipation and faster operation instead of high resolution. The die area occupied by the 8-bit and the 12-bit ADC is only 0.06mm^2 and 0.12mm^2 respectively. The power dissipation is also quite low: 32mW for 8-bit and 72mW for 12-bit resolution. The average sampling rate is 140MS/s in the 12-bit case or more than 150MS/s if 8-bit resolution is required. The area, power consumption and sampling rate of the incorporated 4-bit ADC are 0.008mm^2, 11mW and more than 200MS/s respectively.
The temperature range where the developed ADC is guaranteed to operate without missing codes is -10^{o}C..+50^{o}C. This range can be extended to -30^{o}C..+90^{o}C if a calibration procedure is followed that adjusts appropriately the gain of the M5-M6 and M7-M8 mirrors of Fig. 8. This can be achieved by using gain boosted mirrors instead of the simple ones used in this figure.
The 12-bit ADC is powered by a 1.8V voltage supply if cascode mirrors are used. A 1.1V supply can be used if the ADC is designed with simple instead of cascode mirrors. For example, a 4-bit ADC developed with simple mirrors occupied only 0.0011mm^2, dissipated 3.5mW and achieved a speed of 350MS/s, with the cost of significantly higher mismatch sensitivity. Table 2 summarises the most important features of ADCs that are referenced in this chapter. In Sigma delta ADCs, the speed refers to the sampling frequency which should be significantly higher than the input signal frequency. A question mark is used if the die area occupied by an ADC is not mentioned by the authors. In the last column a composite measure is given in order to compare ADCs with different resolution developed in different technologies. The speed is multiplied by the resolution bits and then divided by the area and the power. The area used in this estimation is the actual die area divided by the technology in order to offer a fair comparison for the older ADCs although the sizes of the transistors used in mixed analogue/digital circuits do not change linearly with the technology length.
It is clear from Table 2, that the presented ADC architecture requires the lowest die area compared to other approaches with similar resolution, without sacrificing speed and with a low enough power consumption. Consequently, the proposed ADC achieves a remarkable trade off between resolution, speed, area and power consumption, that is better than most of the listed approaches.
The DNL error of the developed 8-bit ADC is plotted in Fig. 13a and the corresponding INL error in Fig. 13b. There are some significant DNL errors that appear at a few codes. These codes have a binary representation of the form xxxx0000 and xxxx1111. The DNL error at these codes can be reduced by optimising the height and offset of the teeth in Fig. 9b. More specifically, the current sources of the divider that was described in Fig. 6 can be designed with an optimised value close to I_{ref} that is decided through simulation. By modifying the value of these sources the taps' height at the output of the divider is adjusted. An appropriate scaling of the transistor sizes of the current mirrors used in Fig. 8 can also be used to adjust globally the height and offset of the teeth in Fig. 9b.

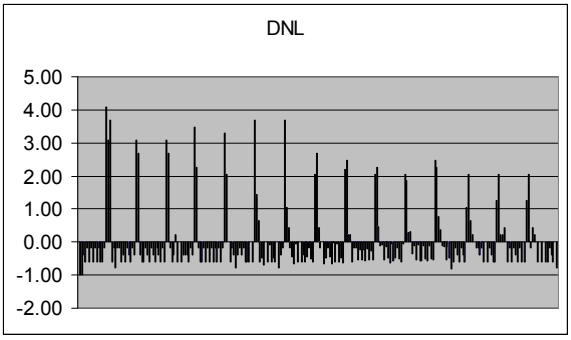
Reference	Resolution Bits	Speed	Area (mm^2) @Technology	Power mW	(Speed·Bits)/ (Area·Power)
This Work	4bits-simple mirror	350 MS/s	0.0011 @90nm	3.5	32727
This Work	4bits-cascode mirror	200 MS/s	0.008 @90nm	11	818
This Work	8	150 MS/s	0.06 @90nm	32	56
This Work	12	140 MS/s	0.12 @90nm	72	17
(Ahmed&Johns, 2005)	10	50 MS/s	1.2 @0.18um	35	2.14
(Arias et al,2006) (Sigma Delta)	8.9	320MHz	0.44 @0.25um	32	-
(Bhat et al,2004)	7	80 MS/s	?	78	-
(Chan et al,2007)	7	5GS/s	?	?	-
(Chen & Wu, 1998)	10	12kS/s	4 @0.8um	2	0.012
(Deguchi et al, 2008)	6	3.5GS/s	0.1485 @90nm	98	130
(Hedayati,2004) (Sigma Delta)	11.8	20MHz	?	1.1	-
(Iizuka et al, 2006)	14	40MS/s	1.15 @0.18um	72.8	1.2
(Liu et al, 2003)	6	450MS/s	2.4 @0.5um	190	3
(Makigawa et al, 2006)	7	800MS/s	0.32 @90nm	120	13
(Masood et al, 2005)	6	2GHz	0.025 @0.18um	19	4547
(Nejime et al, 1991)	8	300MS/s	33 @2.5um	3300	0.055
(Park et al, 2006)	5	3.5GS/s	0.658 @90nm	227	11
(Poulton et al, 2003)	8	250MS/s	0.12 @90nm	57	26
(Walden, 1990)	4	400MS/s	?	100	-

Table 2. ADC comparison

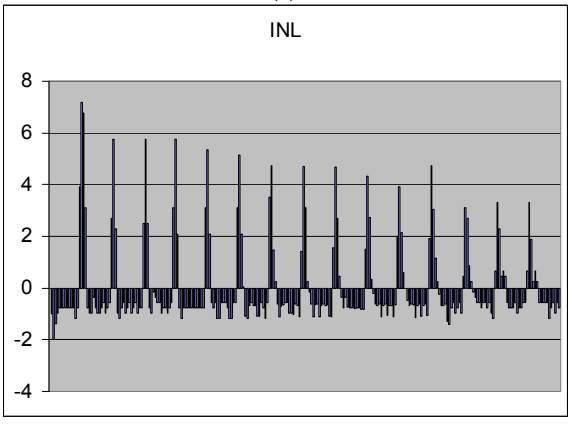
6. Conclusions

The most important Analogue to Digital Conversion techniques have been described briefly in this chapter focusing on their throughput, resolution, power consumption as well as the required area. An asynchronous Analogue to Digital Conversion technique based on a binary tree structure has been proposed. It is implemented with current mode circuits requiring a very small die area and low power consumption without sacrificing speed. More specifically, a 12-bit ADC, that has been developed based on this technique occupied only $0.12mm^2$, dissipated 72mW and had an average throughput of 140MS/s. A thorough comparison with 16 other ADCs showed that the developed 4-, 8- and 12-bit ADCs achieve a

very good trade off between resolution, speed, area and power consumption making its use attractive in various control, sensor readout, communication and other applications.
Our future work includes the investigation of the use of several different synchronous and asynchronous comparators in order to further enhance the speed of the conversion. Higher resolution ADCs will also be developed. The use of various compression algorithms will be studied in order to achieve lower power consumption by transmitting the digitised samples with lower frequency through a smaller number of transmission lines.



(a)



(b)

Fig. 13. DNL and INL error of the developed ADC

7. Acknowledgments

This work was supported by Analogies S.A. and is patent pending (Application No. 0815802.4 / UKPTO).

8. References

Ahmed, I. & Johns, D. (2005). A 50-MS/s (35 mW) to 1-kS/s (15 uW) Power Scaleable 10-bit Pipelined ADC Using RapidPower-On Opamps and Minimal Bias Current Variation. *IEEE Journal of Solid State Circuit*, 2005, Vol. 40, No. 12, pp. 2446-2455.

Arias, J., Kiss, P., Prodanov, V., Boccuzzi, V., Banu, M., Bisbal, D., Pablo, J.S., Quintanilla, L., and Barbolla, J. 2006. A 32-mW 320-MHz Continuous-Time Complex Delta-Sigma ADC for Multi-Mode Wireless-LAN Receivers. *IEEE Journal of Solid State Circuits*, 2006, Vol. 41, No. 2, pp. 339-351.

Bhat, M.S., Rekha, S. & Jamadagni, H.S. (2004). Design of Low Power Current Mode Flash ADC. *Proceedings of the IEEE TENCON Conference*. Vol. 4, pp. 241-244, Nov. 2004.

Chan, B., Oyama, B., Monier, C. & Guitierrez, A. (2007). An Ultra-Wideband 7-bit 5Gsps ADC Implemented in Submicron InP HBT Technology. *Proceedings of the IEEE Compound Semiconductor Integrated Circuit Symposium*, Oct 2007.

Chen, C.C. & Wu, C.Y. (1998). Design Techniques for 1.5V Low Power CMOS Current Mode Cyclic ADCs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. Jan 1998. Vol. 45, No. 1, pp. 28-40.

Deguchi, K., Suwa, N., Ito, M., Kumamoto, T., and Miki, T. (2008). A 6-bit 3.5GS/s 0.9-V 98-mW Flash ADC in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 2008, Vol. 43, No. 10, pp. 2303-2310.

Hedayati, H. 2004. A Low Power Low Voltage Fully Digital Compatible ADC. *Proceedings of the IEEE International Conference on Microelectronics*, pp. 227-230, 2004.

Iizuka, K., Matsui, H., Ueda, M., and Daito, M. (2006). A 14-bit Digitally Self-Calibrated Pipelined ADC With Adaptive Bias Optimization for Arbitrary Speeds Up to 40 MS/s. *IEEE Journal of Solid State Circuits*, 2006, Vol. 41, No. 4, pp. 883-890.

Liu, F., Jia, S., Lu, Z. & Ji, L. (2003) CMOS Folding and Interpolating A/D Converter with Differential Compensative T/H Circuit. *Proceedings of the IEEE Conference on Electron Devices and Solid State Circuits,* pp. 453-456, Dec. 2003.

Makigawa, K., Ono, K., Ohkawa, K., Matsuura, K. and Segami, M. 2006. A 7bit 800Msps 120mW Folding and Interpolation ADC Using a Mixed-Averaging Scheme. *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, 2006.

Masood Ali, S., Raut, R. & Sawan, M. (2005). A Power Efficient Decoder for 2.5GHz, 6-bit CMOS Flash-ADC Architecture. *Proceedings of the 9th IEEE International Database Engineering & Application Symposium.*

Nejime, Y., Hota, M. & Ueda, S. (1991). An 8-b ADC with Over-Nyquist Input at 300-MS/s Conversion Rate. *IEEE Solid State Circuits*, Vol. 26, No. 9, Sept. 1991, pp. 1302-1308.

Park, S., Palaskas, Y., Ravi, A., Bishop, R. & Flynn, M. (2006). A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS. *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 489-492. Sep. 2006.

Poulton, K., Neff, R., Setterberg, B., Wuppermann, B., Kopley, T., Jewett, R., Pernillo, J., Tan, C. & Montijo, A. 2003. A 20GS/s 8b ADC with a 1MB Memory in 0.18μm CMOS. *Proceedings of the IEEE International Solid-State Circuits Conference*. 2003.

Promitzer, G. (2001). 12-bit Low Power Fully Differential Switched Capacitor Non-Calibrating Successive Approximation ADC with 1MS/s. *IEEE Journal of Solid State Circuits,* Vol. 36, No. 7, July 2001, pp. 1138-1143.

Traff, H. (1992). Novel Approach to High Speed CMOS Current Comparators. *IEEE Electron. Letters*, Vol. 28, No. 3, 1992, pp. 310-312.

Walden, R., Schmitz, A., Kramer, A., Larson, L. & Pasiecznik, J. (1990). A Deep Submicrometer Analog-to-Digital Converter Using Focused-Ion-Beam Implants. *IEEE Solid State Circuits,* Vol. 25, No. 2, Apr. 1990, pp. 562-571.

Yuan, J. & Svensson, C. (1994). A 10-bit, 5-MS/s Successive Approximation ADC Cell Used in a 70-MS/s ADC array in 1.2um CMOS. *IEEE Journal of Solid State Circuits.* Vol. 29, No. 8, Aug. 1994, pp. 866-872.

21

Implementation of the delay compensator approach

Ana Antunes[1], Fernando Dias[2], José Vieira[3] and Alexandre Mota[4]
[1]*ESTSetúbal, Instituto Politécnico de Setúbal*
[2]*DME and CCM, Universidade da Madeira*
[3]*EST, Instituto Politécnico de Castelo Branco*
[4]DETI and IEETA, *Universidade de Aveiro*
Portugal

1. Introduction

Modern distributed embedded control systems need to be highly distributed, highly integrated and should support operational flexibility (Årzén et al., 2005). The distribution of control systems induces delays in the control loop that degrade the control performance. These delays are usually variable from iteration to iteration of the control loop and depend on several factors as the scheduling of the processes in the network nodes or the scheduling of traffic in the network.
Classic control solutions do not account for the effect of such variable delays. The delay compensator principle was recently proposed in order to deal with the variable sampling to actuation delay effect in real-time distributed control systems.
The delay compensator principle proposes the addition of a compensator to an existing distributed control system to overcome the effect of the variable sampling to actuation delay introduced in the loop, allowing the achievement of a better control performance.
The compensator action is based on the knowledge of the sampling to actuation delay that affects the control loop and it can have any other input needed to generate the correction output that will be added to the output of the existing controller.
This approach can be applied to any distributed control system provided that the sampling to actuation delay is known for each control cycle. The delay compensator principle is generic and can be implemented using different techniques.
In this chapter the delay compensator approach is presented in detail and several implementations are proposed based on non-linear modelling strategies, namely fuzzy logic, neural networks and neuro-fuzzy techniques.
Simulation results are presented for each one of the implementations proposed and the results obtained are compared.
The remainder of this chapter is organized as follows: section 2 presents the problem description, section 3 presents the delay compensator approach, section 4 proposes the fuzzy implementation of the compensator, section 5 presents the neural networks

implementation of the compensator, section 6 proposes the neuro-fuzzy implementation of the compensator, section 7 describes the test system and the tests, section 8 presents the simulation results and compares the implementations and section 9 concludes this chapter.

2. Problem Formulation

Networked control systems are widely used in embedded applications. Figure 1 presents the block diagram of a distributed real-time control system. In this kind of systems the sensor, the controller and the actuator are implemented in separated nodes that are connected through a communication network, usually a fieldbus.

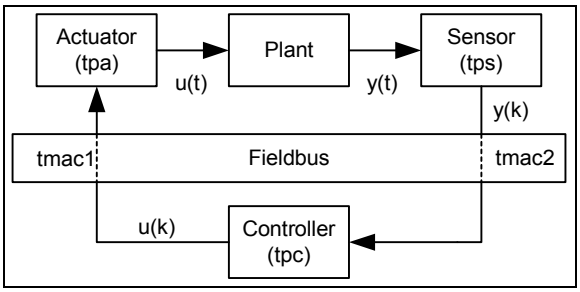


Fig. 1. Block diagram of a distributed real-time controller.

The distribution of the controller and the use of a communication network to connect the nodes of the control loop induce variable delay between the sampling instant and the actuation instant as can be seen in fig. 2.

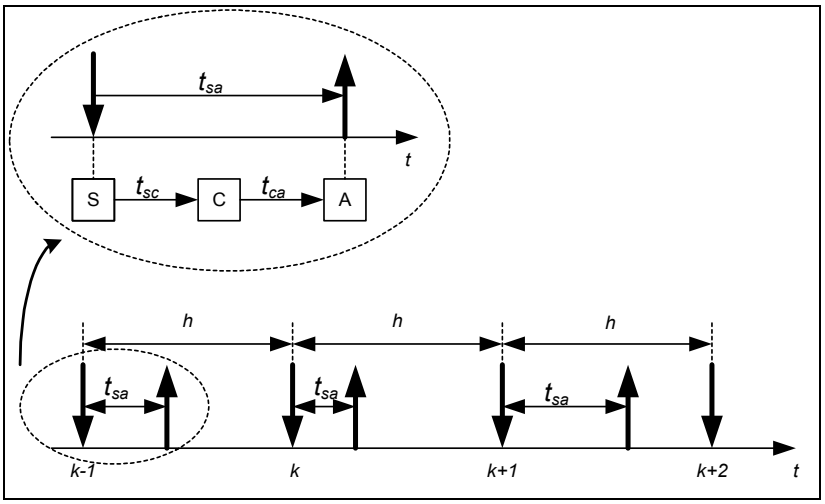


Fig. 2. Representation of the time between the sampling and the actuation instants.

The sampling to actuation delay (tsa) can be derived from fig. 2 as

$$t_{sa} = t_{sc} + t_{ca} \quad (1)$$

where, from fig. 1

$$t_{sc} = t_{ps} + t_{mac1} \quad (2)$$

and

$$t_{ca} = t_{pc} + t_{mac2} + t_{pa} \quad (3)$$

The delays are introduced due to the Medium Access Control (MAC) and the scheduling mechanism used to schedule the bus time at the network level (t_{mac}) and the processing time and scheduling at the node level (t_{ps}, t_{pc} and t_{pa}). These delays are usually variable from iteration to iteration of the control loop.
In this work the sampling to actuation delay is considered to be less than or equal to the sampling period, since the system is implemented in a real-time framework, where the transmission of the messages is bounded by a hard deadline.
The variable delays introduced in the control loop by the distribution degrade the performance of the control system and can even destabilize the system (Cervin, 2003a), (Tipsuwan and Chow, 2003), (Colom, 2002), (Sanfridson, 2000), (Antunes et al., 2004a) and (Antunes et al., 2004b). Since the destabilization of the control system is not acceptable and the degradation of the control performance is not desirable, it is necessary to search for solutions to overcome this problem.
There are several ways to deal with the variable sampling to actuation delay in distributed control systems including the use of linear and non-linear modeling techniques. In this work we only consider the use of non-linear techniques.
In the framework of non-linear techniques the use of compensators to improve the control performance of the loop is common. Different kinds of compensators have been used to improve the control performance of networked control systems.
(Almutairi et. al., 2001) proposed a fuzzy modulator to use with a PI controller. The modulator produces a multiplying factor that is applied to the control signal coming from the PI controller. The modulator's output is based on the error and the control signal, making it unclear whether the compensation is related to the delay effect or to the need to improve the PI controller output.
(Lin et al., 2008) proposed a fuzzy-PID controller to work with a neural net-based Smith predictor in order to compensate for the delay effect in a non real-time implementation. This compensation scheme is used in an Ethernet based networked control system with random delays that can be longer than the sampling period with the aim of maintaining the stability of the system.
The delay compensator approach was presented in (Antunes et al., 2006) where a fuzzy implementation of the compensator was proposed. The fuzzy compensator was able to improve the control performance but did not achieve the same performance obtained in the system in the absence of delays.
Since it is well known that the effect of the delays introduced are difficult to model due to its complexity (Cervin, 2003a), (Colom, 2002) the next step was the use of neural networks and neuro-fuzzy techniques to implement the compensator.

3. The Delay Compensator Approach

The delay compensator approach proposes to add a delay compensator to an existing controller that does not take into account the effect of the variable sampling to actuation delay in the loop.
The compensator proposes a correction to the control action in order to reduce the effect of the sampling to actuation delay.
Figure 3 shows the block diagram of the delay compensator approach.

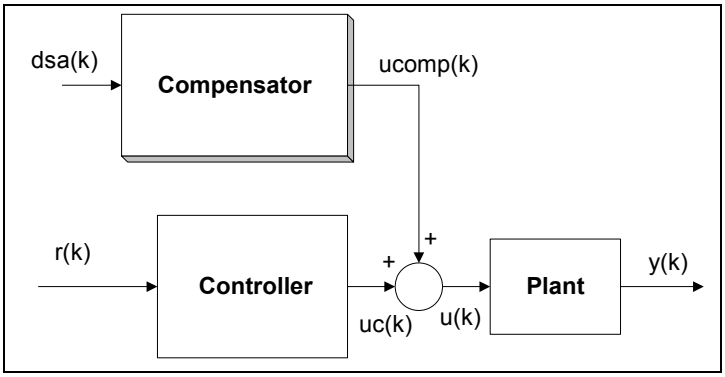


Fig. 3. Block diagram of the delay compensator approach.

The compensator action is based on the sampling to actuation delay that affects the system at each control cycle but it can have any other input.
This principle can be applied to any distributed control system provided that the sampling to actuation delay is known for each control cycle.
The determination of the sampling to actuation delay can be done either by an online measurement or by off-line computations depending on the a priori knowledge of the overall system operation conditions. The online measurement of the sampling to actuation delay provides a more generic and flexible solution that does not require the knowledge of the details of the operation conditions of the global system in which the distributed controller is inserted.
In order for the compensator to operate with the correct value of the delay affecting the network, the controller and the actuator must be implemented in the same network node.
The delay compensator principle is generic and can be implemented using different techniques.
Since the compensator output is added to the output signal from the existing controller, the compensator can easily be turned on or off and the control loop will be closed by the existing controller.

4. Fuzzy Implementation of the Delay Compensator Approach

In (Antunes et al., 2006) a fuzzy (FZ) implementation of the delay compensator was proposed based in the empirical knowledge about the delay effect on the distributed control system. A linear approximation was used to model the delay effect. The linear

approximation is based on the simple fact that when there is sampling to actuation delay there is less time for the control signal to act on the controlled system. This can be compensated by increasing or decreasing the control action depending on the previous values of the control action and on the amount of the sampling to actuation delay. It is also known that small amounts of delay do not need to be accounted because they do not affect significantly the control performance (Cervin, 2003a).
The fuzzy (FZ) compensator is based on this linear approximation of the sampling to actuation delay effect. The compensator action is achieved by increasing or decreasing the control output according to the delay amount and the previous value of the control output in order to compensate for the delays introduced in the loop.
The fuzzy compensator block is presented in figure 4.

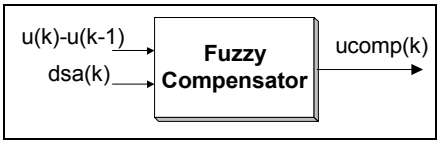


Fig. 4. Block of the fuzzy implementation of the delay compensator approach.

The fuzzy module uses a Mamdani type function (Mamdani and Assilian, 1975) with two inputs (the sampling to actuation delay and the previous value of the control signal from the existing controller), one output (the compensation value) and six rules.
The rules state that if the difference between the previous two samples of the control signal is "null" then the contribution of the compensator is "null". If the delay is "low" then the contribution is "null". Otherwise if the delay is medium or high, the contribution of the compensator will also be medium or high, with a sign given by the difference between the previous two samples of the control signal: if the control signal is decreasing then the contribution of the compensator is negative and if the control signal is increasing then the contribution is positive. This corresponds to a linear approach of the effect of the sampling to actuation delay. The membership functions for the inputs and the output are bell shaped.
As described in (Antunes et al., 2006) the fuzzy delay compensator allowed the improvement of the control performance of the loop but was not able to overcome all the performance degradation induced by the delays.

5. Neural Networks Implementation of the Delay Compensator

This section describes the neural networks (NN) implementation of the delay compensator to further improve the control performance. This implementation was first proposed in (Antunes et al., 2008a).
The model for the NN delay compensator is not a regular model. The information available to train the model is the output of a certain system with ($y_d(k)$) and without ($y(k)$) the effect of sampling to actuation delay. The objective is to produce a model that can compensate the effect of this delay (knowing the delay for the iteration) in order to correct the control signal to avoid the degradation of the performance. It is necessary to find a way to produce a model that can perform the requested task.
The authors considered two possibilities:

- calculating the error between the two outputs: $e_y(k)=y(k)-y_d(k)$ and reporting this error to the input through an inverse model or,
- calculating the equivalent input of a system without sampling to actuation delay that would have resulted in the output $y_d(k)$. This is also obtained through an inverse model. The second approach is illustrated in figure 5.

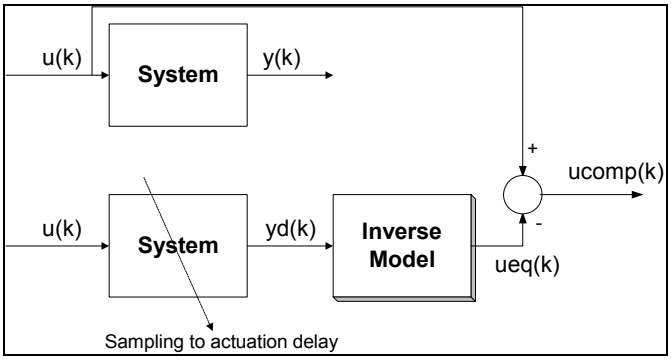


Fig. 5. Method used to calculate the output of the model.

The first alternative would only be valid in the case of a linear system, since for non-linear systems there is no guarantee that the output error could be reported to the corresponding input, since the error range ($y- y_d$) is very different from the output signal range. Using the difference between y and y_d and applying it to the inverse model could result in a distortion due to the non-linearity.
The study of the lag space and the use of the method proposed in fig. 5 resulted in the model represented in fig. 6.

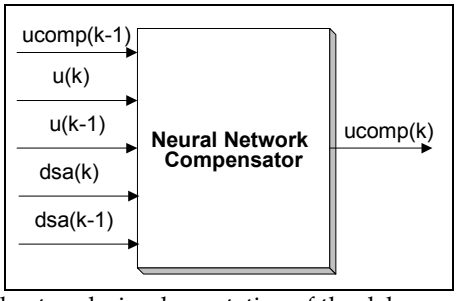


Fig. 6. Block of the neural networks implementation of the delay compensator approach.

This model has, as inputs, a past sample of the output of the compensator, two past samples of the control signal and two past samples of the delay information. The model is composed of ten neurons with hyperbolic tangents in the hidden layer and one neuron in the output layer with linear activation function and was trained for 15000 iterations with the Levenberg-Marquardt algorithm (Levenberg, 1944), (Marquardt, 1963).

6. Neuro-fuzzy Implementation of the Delay Compensator

The use of neuro-fuzzy techniques to implement the delay compensator was first proposed in (Antunes et al., 2008b).
The model needed to implement the neuro-fuzzy delay compensator approach is similar to the one describbed in the previous section. Using the method from fig. 5 and studying the lag space results in the model represented in figure 7.

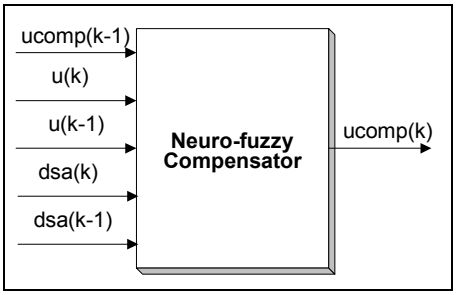


Fig. 7. Block of the neuro-fuzzy implementation of the delay compensator approach.

This model inputs' are: a past sample of the output, two past samples of the control signal and two past samples of the delay information.
The ANFIS structure used to obtain the model contains five layers and 243 rules. It has five inputs with three membership functions each (bell shaped with three non-linear parameters) and one output. The total number of fitting parameters is 774, including 45 premise parameters (3*3*5 non-linear) and 729 consequent parameters (3*243 linear). The neuro-fuzzy (NF) model was trained for 100 iterations with the ANFIS function of MATLAB toolbox (MATLAB, 1996).

7. The Test System

The architecture of the test system, the tests and the existing controller will be presented in the following subsections.

7.1 Architecture of the distributed system

The test system is composed of 2 nodes: the sensor node and the controller and actuator node connected through the Controller Area Network (CAN) bus (Bosch, 1991). The controller and the actuator have to share the same node in order to be possible to measure accurately the value of the sampling to actuation delay that affects the control loop at each control cycle. The block diagram of the distributed system is presented in figure 8.
Message M1 is used to transport the sampled value from the sensor node to the controller and actuator node.
The transfer function of the plant is presented in (4).

$$\frac{U(s)}{Y(s)} = \frac{0.5}{s + 0.5} \tag{4}$$

The system was simulated using TrueTime, a MATLAB/Simulink based simulator for real-time distributed control systems (Cervin et al., 2003b).

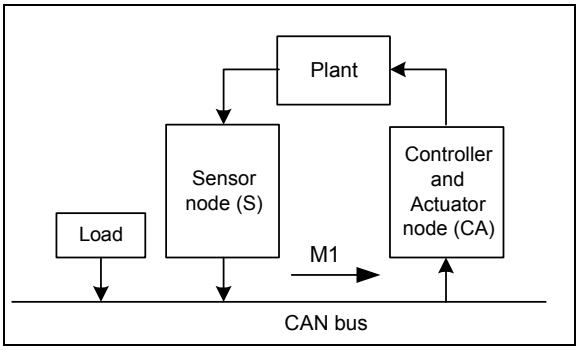


Fig. 8. Block diagram of the test system.

7.2 Description of the tests

Three different situations were simulated.

Test 1 is the reference test, where the sampling to actuation delay is constant and equal to 4 ms. It corresponds to the minimum value of the MAC and processing delays obtained when the bus is used only by message M1.

In tests 2 and 3 additional delay was introduced to simulate a loaded network. The sampling to actuation delay introduced follows a random distribution over the interval [0,h] for test 2 and a sequence based in the gamma distribution that concentrates the values in the interval [h/2, h] for test 3.

The sampling to actuation delay obtained for tests 2 and 3 is depicted in figure 9 and 10.

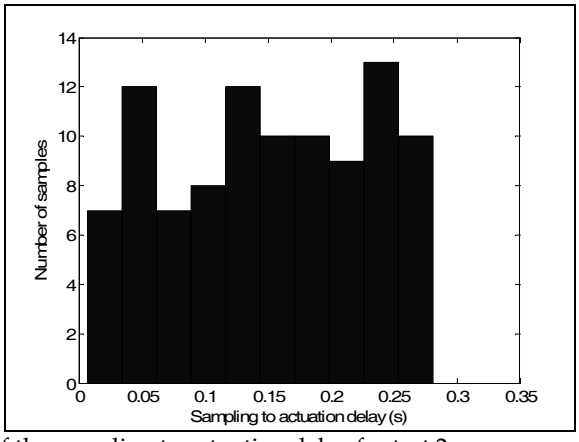


Fig. 9. Histogram of the sampling to actuation delay for test 2.

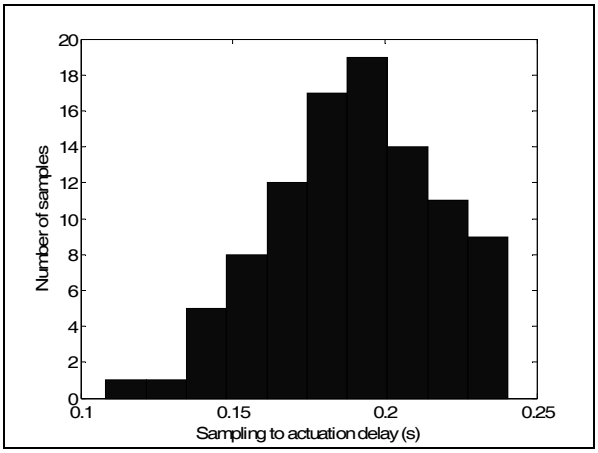


Fig. 10. Histogram of the sampling to actuation delay for test 3.

The delay compensator was implemented in the controller and actuator node according with the block diagram from fig. 11.

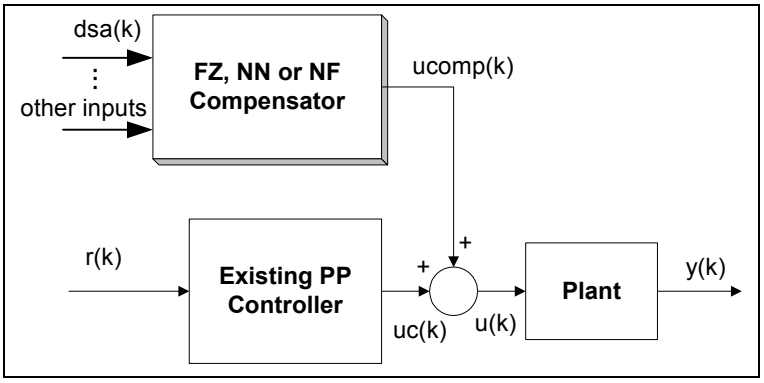


Fig. 11. Block diagram of the delay compensator implementation.

The tests were made for the system with and without the delay compensator.

7.3 Existing controller

The existing controller is a pole-placement (PP) controller (Åström and Wittenmark, 1997). It does not take into account the sampling to actuation delay. The controller parameters are constant and computed based on the discrete-time model given by equation (5).

$$G(q^{-1}) = \frac{bq^{-1}}{1 - aq^{-1}} \tag{5}$$

The pole-placement technique allows the complete specification of the closed-loop response of the system by the appropriate choice of the poles of the closed-loop transfer function. In this case the closed-loop pole is placed at α_m=2 Hz. An observer was also used with α_0=4 Hz.
The sampling period (h) is equal to 280 ms and was chosen according to the rule of thumb proposed by (Åström and Wittenmark, 1997).
The identification of the system was based in the discrete model in (5) and the parameters were computed off-line.
The parameters of the control function were obtained by directly solving the Diophantine equation for the system. The resulting control function is given by (6).

$$u_c(k) = t_0(r(k) - a_0 r(k-1)) - s_0 y(k) - s_1 y(k-1) + u_c(k-1) \tag{6}$$

where t_0=3.2832, a_o= 0.3263, s_0=7.4419 and s_1= -5.2299.

8. The Simulation Results

The results obtained for test 1 (reference test) with only the existing pole-placement (PP) controller are presented in figure 12.

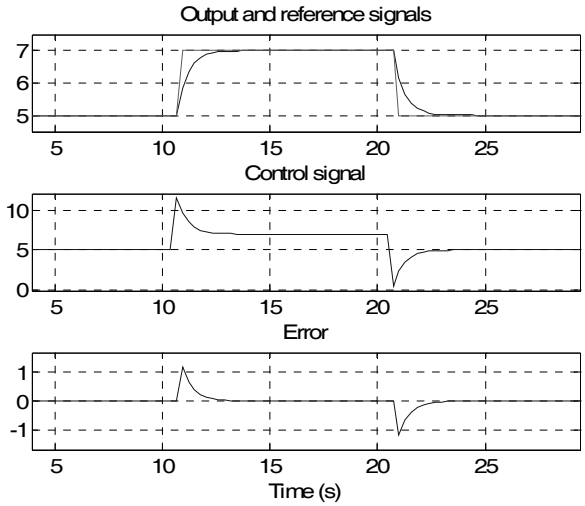


Fig. 12. Control results for test 1 (reference test) without the delay compensator.

The tests were performed for the system with only the PP controller (reference test) and for the system with the fuzzy (FZDC), the neural networks (NNDC) and neuro-fuzzy (NFDC) implementations of the delay compensator.
The control performance was assessed by the computation of the Integral of the Squared Error (ISE) between t= 5 s and t=29 s. The results obtained for ISE are presented in Table 1.

Test	PP	FZDC	NNDC	NFDC
1	3.3	3.3	3.3	3.3
2	3.8	3.6	3.4	3.6
3	4.8	4.1	3.7	4.0

Table 1. ISE report.

The percentage of improvement obtained compared to the reference test (test 1) for ISE is presented in Table 2.

Test	FZDC	NNDC	NFDC
2	40%	80%	40%
3	47%	73%	53%

Table 2. Improvement report.

The improvement is calculated as the amount of error induced by the sampling to actuation delay that the FZDC, NNDC or NFDC compensators were able to reduce. The formula used for the computation of the improvement is presented in (7).

$$Iprv(\%) = (1 - \frac{ISE_{DC} - ISE_{DC\,\mathrm{Re}\,f}}{ISE_{PP} - ISE_{PP\,\mathrm{Re}\,f}}) * 100 \tag{7}$$

where ISE_{DC} represents the ISE value obtained with the delay compensator and ISE_{PP} represents the ISE value obtained with only the PP controller.
The control results for tests 2 and 3 with and without the delay compensator are presented in figures 13 to 20.

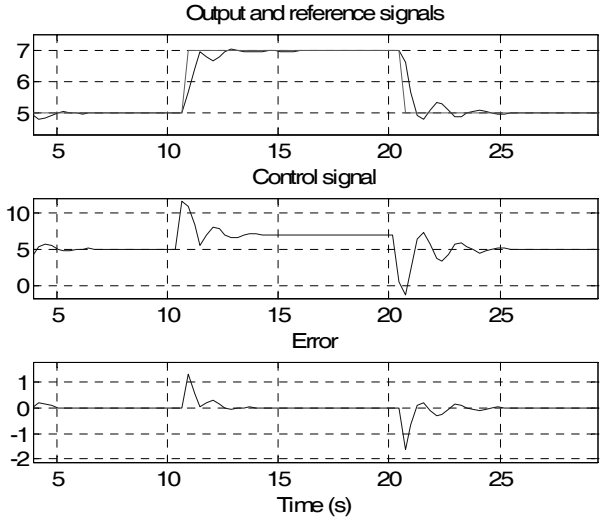


Fig. 13. Control results for test 2 without the delay compensator.

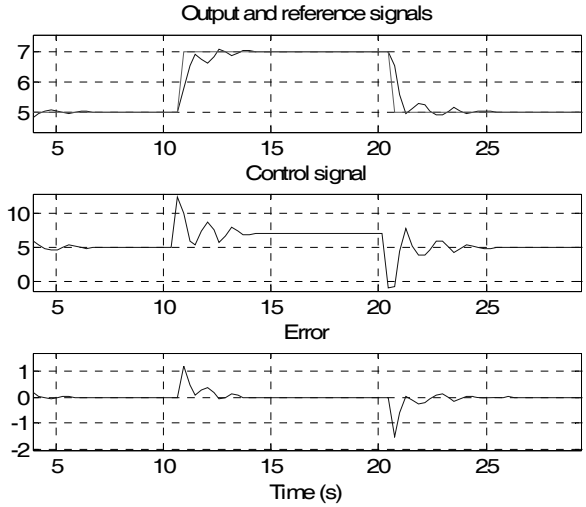


Fig. 14. Control results for test 2 with the FZ delay compensator.

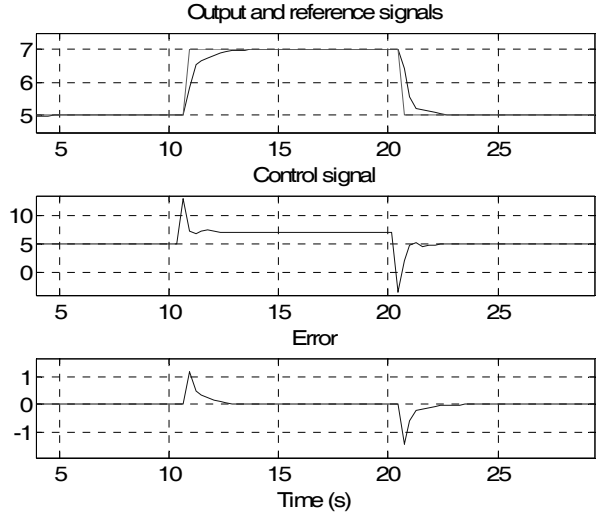


Fig. 15. Control results for test 2 with the NN delay compensator.

The results show the effectiveness of the delay compensator. For tests 2 and 3 the control performance obtained using the delay compensator is better than the ones obtained without compensation.

The NN implementation of the delay compenator achieved the best results. For test 2 the NN compensator was able to reduced by 80% the effect of the variable sampling to actuation delay and for test 3 the reduction is equal to 73%. The FZ compensator was able to improve

the control performance in 40% for test 2 and 47% for test 3. The results obtained for the NF compensator (40% for test 2, 53% for test 3) are similar to the ones obtained with the FZ compensator.
Against our expectations the NF compensator is not able to improve the control performance as much as the NN compensator.

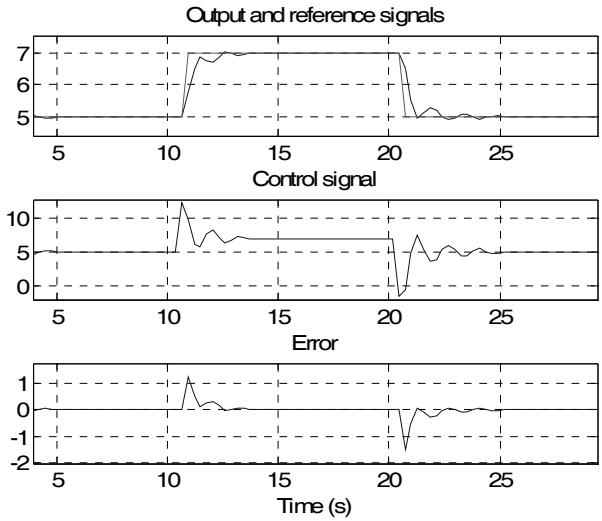


Fig. 16. Control results for test 2 with the NF delay compensator.

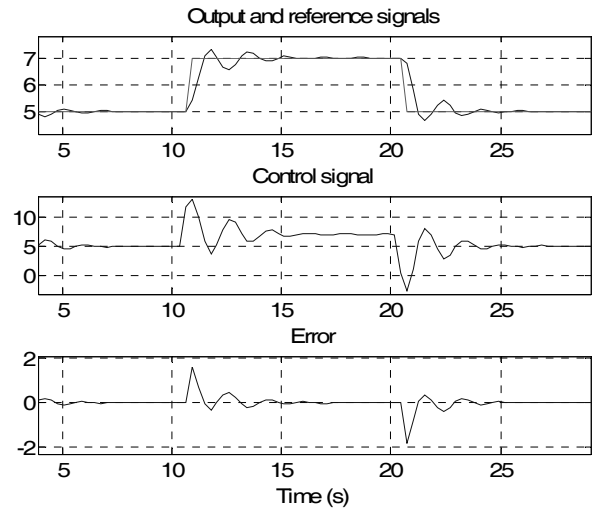


Fig. 17. Control results for test 3 without the delay compensator.

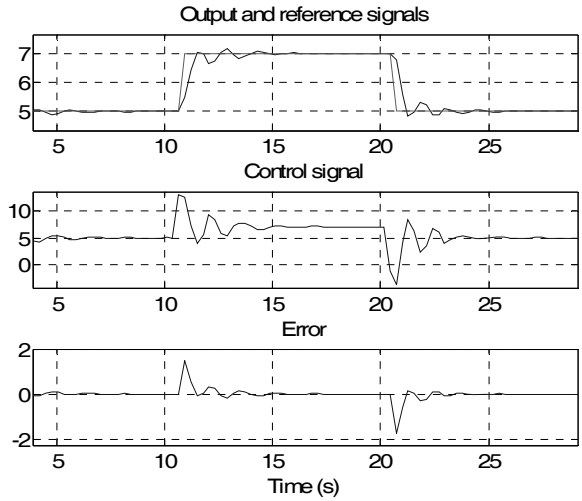


Fig. 18. Control results for test 3 with the FZ delay compensator.

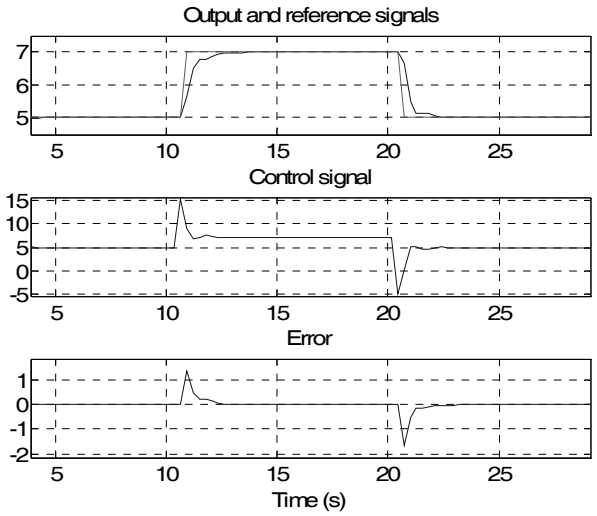


Fig. 19. Control results for test 3 with the NN delay compensator.

The control signals show that the fuzzy and the neuro-fuzzy compensators are not able to prevent the oscillations of the output of the plant, while the neural networks compensator is. Nevertheless the other techniques are also able to improve the control performance and are easier and faster to implement because they do not need the expertise required to train the neural networks model.

In conclusion, neural networks seem to be better suited for modelling the effect of the variable sampling to actuation delay in distributed control systems.

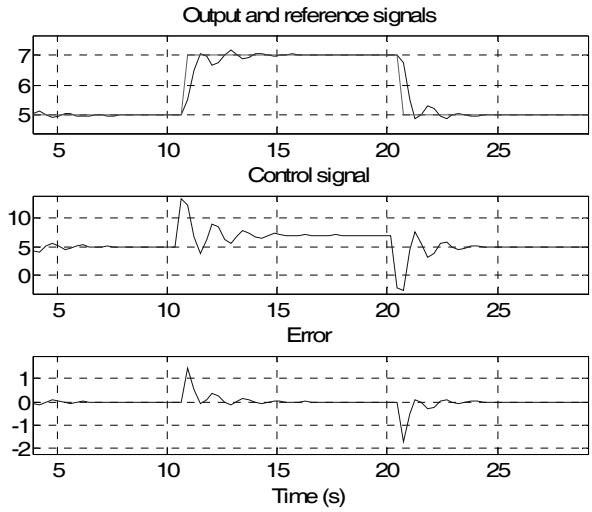


Fig. 20. Control results for test 3 with the NF delay compensator.

9. Conclusion

This chapter presents several implementations of the delay compensator approach using non-linear techniques. The delay compensator can be added to any existing distributed control system provided that the sampling to actuation delay can be determined for each control cycle. The delay compensator is implemented through a model that describes the effect of the sampling to actuation delay in terms of the control signal. This model is then used to compensate the control signal according to the delay at the actuation moment.

All the implementations proposed were able to improve the control performance in the distributed control system.

The fuzzy implementation is based in a linear approximation of the delay effect. It is simple to design and implement but is not able to prevent oscillations on the output of the plant.

The neuro-fuzzy implementation is easier to train than the neural networks approach and achieves results similar to the ones obtained with fuzzy logic.

The neural networks compensator was able to model effectively the effect of the sampling to actuation delay on the distributed control system and it achieved the best results. This approach requires the use of some expertise in produce the model needed to implement the compensator.

Future work includes the test of these implementations of the delay compensator approach using real prototypes and different plants.

10. References

Almutairi, N. B.; Chow, M.-Y. & Tipsuwan, Y. (2001). Networked-based controlled DC motor with fuzzy compensation, *Proceedings of the. 27th Annual Conf. of the IEEE Industrial Electronics Society*, pp. 1844-1849.

Antunes, A.; Dias, F.M. & Mota, A.M. (2004a). Influence of the sampling period in the performance of a real-time adaptive distributed system under jitter conditions, *WSEAS Transactions on Communications*, Vol.3, pp. 248-253.

Antunes, A. & Mota, A.M. (2004b). Control Performance of a Real-time Adaptive Distributed Control System under Jitter Conditions, *Proceedings of the Conference. Control 2004*, Bath, UK, September 2004.

Antunes, A.; Dias, F.M.; Vieira, J. & Mota, A. (2006). Delay Compensator: an approach to reduce the variable sampling to actuation delay effect in distributed real-time control systems, *Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation*, Prague, Czech Republic.

Antunes, A.; Dias, F.M. & Mota, A. (2008a) A neural networks delay compensator for networked control systems. *Proceedings of the. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Hamburg, Germany, September, 2008, pp. 1271-1276.

Antunes, A.; Dias, F.M.; Vieira, J. & Mota, A. (2008b) A neuro-fuzzy delay compensator for distributed control systems. *Proceedings of the IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Hamburg, Germany, September, 2008, pp. 1088-1091.

Årzén, K-E.; Cervin, A. & Henriksson, D. (2005). Implementation-aware embedded control systems, In: *Handbook of networked and embedded control systems*, D. Hristu-Varsakelis and W. S. Levine (Ed.), Birkhäuser.

Åström, K.J. & Wittenmark, B. (1997) *Computer Controlled Systems*, Prentice Hall.

Bosch. (1991) CAN specification version 2.0. Tech. Report, Robert Bosch GmbH, Stuttgart, Germany.

Cervin, A. (2003a). *Integrated control and real-time scheduling*, Ph.D. dissertation, Lund Institute of Technology, Sweden.

Cervin, A.; Henriksson, D.; Lincoln, B.; Eker, J. & Arzen, K.-E. (2003b) How does control timing affect performance?, *IEEE Control System Magazine*, Vol. 23, pp. 16-30.

Colom, P.M. (2002). *Analysis and design of real-time control systems with varying control timing constraints*, Ph.D. dissertation, Universitat Politecnica de Catalunya, Spain.

Levenberg, K. (1944) A method for the solution of certain problems in least squares. *Quarterly of applied mathematics*, Vol. 2, pp. 164-168.

Lin, C.-L.; Chen, C.-H. & Huang, H.-C. (2008). Stabilizing control of networks with uncertain time varying communication delays, *Control Engineering Practice*, Vol. 16, pp. 56-66.

Mamdani, E.H. & Assilian, S. (1975). An experimental in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, Vol. 7, pp. 1-13.

Marquardt, D. (1963) An algorithm for least-squares estimation of non-linear parameters, SIAM Journal on applied mathematics, Vol. 11, pp. 431-441.

MATLAB. (2006) Fuzzy Logic Toolbox for MATLAB, Tech. Report.

Sanfridson, M. (2000). Timing problems in distributed real-time computer control systems, Tech. Rep., Royal Institute of Technology, ISSN 1400-1179, Sweden.

Tipsuwan, Y. & Chow, M.-Y. (2003). Control methodologies in networked control systems, *Control Engineering Practice*, Vol.11, pp. 1099-1111.

22

Control of Robot Interaction Forces Using Evolutionary Techniques

Jose de Gea and Yohannes Kassahun
Robotics Group, University of Bremen
Germany

Frank Kirchner
Robotics Group, University of Bremen,
DFKI (Robotics Innovation Center), Bremen
Germany

1. Introduction

For a robot manipulator to interact safely and human-friendly in an unknown environment, it is necessary to include an interaction control method that reliably adapts the forces exerted on the environment in order to avoid damages both to the environment and to the manipulator itself. A force control method, or strictly speaking, a direct force control method, can be used on those applications where the maximum or the desired force to exert is known beforehand. In some industrial applications the objects to handle or work with are completely known as well as the precise moment on which these contacts are going to happen. In a more general scenario, such as one outside a well-defined robotic workcell or when an industrial robot is used in cooperation with a human, neither the objects nor the time when a contact is ocurring are known.

In such a case, indirect force control methods find their niche. These methods do not seek to control maximum or desired force, but they try to make the manipulator compliant with the object being contacted. The major role in the control loop is given to the positioning but the interaction is also being controlled so as to ensure a safe and clear contact. In case contact interaction forces have exceeded the desired levels, the positioning accuracy will be diminished to account and take care of the (at this moment) most important task: the control of the forces. Impedance control (Hogan (1985)) is one of these indirect force control methods. Its aim is to control the dynamic behaviour of a robot manipulator when contacting the environment, not by controlling the exact contact forces but the properties of the contact, namely, controlling the stiffness and the damping of the interaction. Moreover, the steady-state force can be easily set to a desired maximum value. The main idea is that the impedance control system creates a virtual new impedance for the manipulator, which is being able to interact with the environment as if new mechanical elements had been included in the real manipulator.

First industrial approaches were focused on controlling the force exerted on the environment by a direct force feedback loop. A state-of-the-art review of the 80s is provided in (Whitney (1987)) and the progress during the 90s is described in (Schutter et al. (1997)). In many in-

dustrial applications, where objects are located in a known position in space and where the nature of the object is also familiar, the approach is well-suited since it prevents the robot from damaging the goods. If a detailed model of the environment is not available, the strategy is to follow a motion/force control method obtained by closing a force control loop around a motion control loop (De Schutter & van Brussel (1988)). If controlling the contact force to a desired value is not a requirement, but rather the interest is to achieve a desired dynamic behaviour of the interaction, indirect force control methods find their application. This would be the case when the environment is unknown and the objects to manipulate have non-uniform and/or deformable features. In this strategy, the position error is related to the contact force through mechanical stiffness or with adjustable parameters. This category includes compliance (or stiffness) control (Paul & Shimano (1976)), (Salisbury (1980)) and impedance control (Hogan (1985)), (Caccavale et al. (2005)), (Chiaverini et al. (1999)) and (Lopes & Almeida (2006)).
Several schemes are proposed to regulate the robot-environment contact forces and to deal with model uncertainties. In (Matko et al. (1999)) a model reference adaptive algorithm is proposed to deal with the uncertainty of the parameters that describe the environment. In (Erol et al. (2005)) an artificial neural network-based PI gain scheduling controller is proposed that uses estimated human arm parameters to select appropriate PI gains when adapting forces in robotic rehabilitation applications. In (Jung & Hsia (1998)) a neural network approach is also used to compensate both for the uncertainties in the robot model, the environmental stiffness, and the force sensor noise. Similarly, in (Seraji & Colbaugh (1993)) and (Lu & Meng (1991)) adaptive impedance control schemes are presented to deal with uncertainty of the environmental stiffness as well as uncertainty in the parameters of the dynamical model of the robot or the force measurement. These methods adapt the desired trajectory according to the current scenario, though using cumbersome or unclear methodologies for the selection of impedance parameters. Moreover, some of them might not be applied where the environmental properties are of non-linear nature (Seraji & Colbaugh (1993)).
This chapter aims at describing the use of evolutionary techniques to control the interaction forces between a robot manipulator and the environment. More specifically, the chapter focuses on the design of optimal and robust force-tracking impedance controllers. Current state-of-the-art approaches start the analysis and design of the properties of the impedance controller from a manually-given set of impedance parameters, since no well-defined methodology has been yet presented to obtain them. Neuroevolutionary methods are showing promising results as methods to solve learning tasks, especially those which are stochastic, partially observable, and noisy. Evolution strategies can be also used to perform efficient optimization, as it is the case in CMA-ES (Covariance Matrix Adaptation - Evolution Strategy) (Schwefel (1993)).
Neuroevolution is the combination of neural networks as structure for the controller and an evolutionary strategy which in the simplest case searches for the optimal weights of this neural network. The weights of this neural network represent the policy of the agent, in control engineering terms known as the control law. Consequently, the weights of this neural network bound the space of policies that the network can follow. In more complex strategies, the evolutionary strategy evolves both the weights and the topology of the neural network. In optimal control, one tries to find a controller that provides the best performance with respect to some measure. This measure can be for example the least amount of control signal energy that is necessary to bring the controller's output to zero. Whether in classical optimal control or in neuroevolutionary methods, there is an optimization process involved and we show in

this chapter that neuroevolutionary methods can provide a good alternative to easily design optimal controllers.
In this case study, an impedance controller represented as an artificial neural network (ANN) will be described, whose optimal parameters are obtained in a simple way by means of evolutionary techniques. The controller will regulate the contact forces between a robotic manipulator (a two-link planar arm) and the environment. Furthermore, it will be generalised and provided with force tracking capabilities through an on-line parameter estimator that will dynamically compute the weights of the ANN-based impedance controller based on the current force reference.
The resulting controller presents robustness against uncertainties both on the robot and/or the environmental model. The performance of the controller has been evaluated on a range of experiments using a model of a two-link robotic arm and a non-linear model of the environment. The results evidenced a great performance on force-tracking tasks as well as particular robustness against parametric uncertainties. Finally, the controller was enhanced with a steady-state Kalman filter whose parameters were learned simultaneously with the weights of the ANN. That provided robustness against the measurement noise, especially important in the force measurements.

2. System Description

The system's control architecture (Fig. 1) used for the experiments and implemented under MATLAB is composed of the following submodules: Trajectory Generation module, Impedance Controller (neural network-based controller), Direct and Inverse Kinematics modules, Dynamical Controller module, Two-link Arm Dynamical Model, and Environment model.

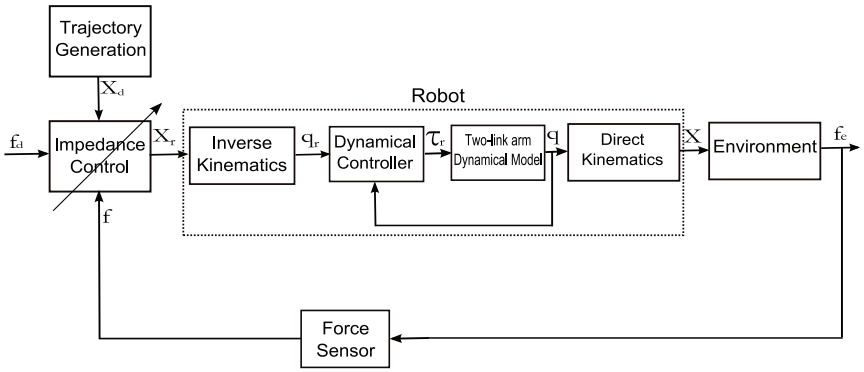


Fig. 1. System's control architecture

2.1 Evolution Strategy

Evolution Strategies (ESs) are a class of Evolutionary Algorithms (EAs) which use nature-inspired concepts like mutation, recombination, and selection applied to a population of individuals containing candidate solutions in order to evolve iteratively better and better solutions. These ESs were introduced by a (back then) unofficial workgroup on Evolution Techniques at the Technical University of Berlin in the late 1960s (Rechenberg (1973)). In contrast to

genetic algorithms, which work with discrete domains, evolution strategies were developed to be used in continuous domains, which make them suitable for continuous-space optimization problems and real-world experiments.

2.1.1 CMA-ES

CMA-ES is an advanced form of evolution strategy (Schwefel (1993)) which can perform efficient optimization even for small population sizes. The individuals are in this algorithm represented by n-dimensional real-valued solution vectors which are altered by recombination and mutation. Mutation is realized by adding a normally distributed random vector with zero mean, where the covariance matrix of the distribution is itself adapted during evolution to improve the search strategy. CMA-ES uses important concepts like *derandomization* and *cumulation*. Derandomization is a deterministic way of altering the mutation distribution such that the probability of reproducing steps in the search space that lead to better individuals is increased. A sigma value represents the standard deviation of the mutation distribution. The extent to which an evolution has converged is indicated by this sigma value (smaller values indicate greater convergence).

2.2 Impedance controller

The classical impedance controller (Fig. 2) is described by Eq. (1):

$$H(s) = \frac{E(s)}{F(s)} = \frac{1}{M_T s^2 + D_T s + K_T} \tag{1}$$

where M_T, D_T and K_T are the inertia, damping, and the stiffness coefficients, respectively, e is the trajectory error, defined as $e = x_d - x_r$ where x_d is the desired trajectory input, and x_r will be the reference trajectory for the next module (the inverse kinematics), corrected depending on the value of the contact force f. The parameters M_T, D_T and K_T will define the dynamic behaviour of the robot that could be compared to the effect of including physical springs and dampers on the robot.

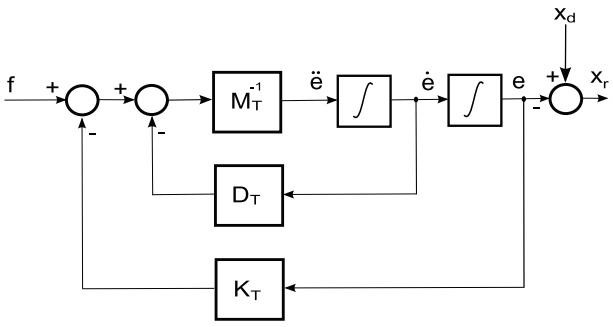


Fig. 2. Classical impedance controller

Starting from (1), the impedance controller can be discretized for its implementation in a computer. Using the bilinear transformation, $H(z) = H(s)\,|_{s=\frac{2}{T}\frac{z-1}{z+1}}$, the discrete version of the impedance controller is obtained.

$$H(z) = \frac{E(z)}{F(z)} = \frac{T^2(z+1)^2}{w_1 z^2 + w_2 z + w_3} \tag{2}$$

where

$$w1 = 4M_T + 2D_T T + K_T T^2 \tag{3}$$
$$w2 = 2K_T T^2 - 8M_T \tag{4}$$
$$w3 = K_T T^2 + 4M_T - 2D_T T \tag{5}$$

From the discretized controller we can generate the difference equation which the filter will be implemented with in the computer:

$$E(n) = \frac{1}{w_1}(F(n)T^2 + 2T^2F(n-1) + T^2F(n-2) - w_2E(n-1) - w_3E(n-2)) \tag{6}$$

Following Eq. (6), it can be clearly seen that the impedance controller can be represented as a neural network as in Figure 3. That means that each classical impedance controller can be implemented as a one-neuron neural network with 5 inputs, 1 output, and only 3 weights.

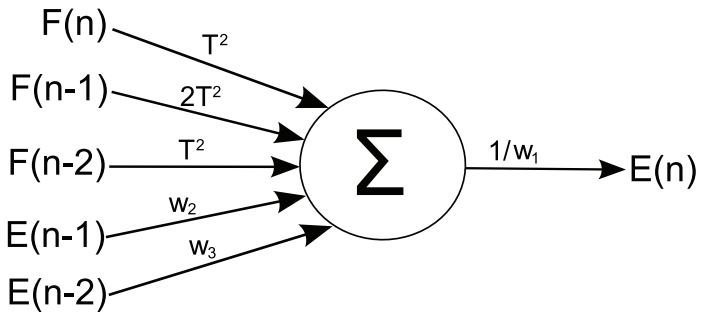


Fig. 3. Neural network representation of the impedance controller

3. Evolving the ANN-based impedance controller

3.1 Single-force reference controller

The weights of the neural network in Fig. 3 are obtained by using the CMA-ES evolutionary technique. In order to do so, the closed-loop system shown in Fig. 4 is used. The ANN-based impedance controller modifies the desired Cartesian position trajectory for the robot (x_d) and creates a new reference trajectory (x_r) based on current sensed forces. The block named *Robot* includes the blocks enclosed under the dotted-line rectangular box in Fig. 1: a dynamical model-based controller that translates the Cartesian positions into the necessary torques for the robot, and forward/inverse kinematics formulations to translate from/to a Joint reference frame to/from a Cartesian frame. The contact forces exerted by the environment onto the robot (f) are fed back to the controller in order to regulate the robot-environment interaction.

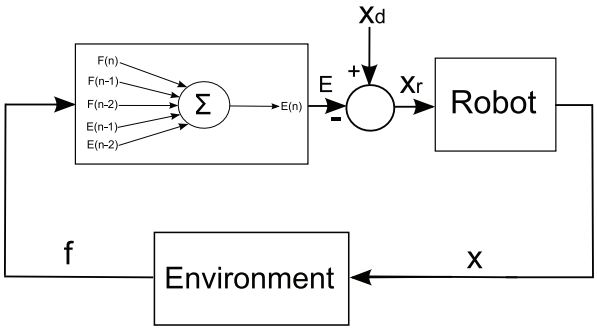


Fig. 4. Close-loop system used to evolve the parameters of the ANN-based controller

The evolutionary algorithm searches for the optimal parameters M_T, D_T, and K_T, and the weights of the neural network are then computed using Eqs. (3), (4), and (5). A fitness function needs to be defined that drives the search and in this case was defined as to minimise the following force error criterion:

$$h = \frac{\sum_{k=1}^{N} \left| f_{ref} - f_k \right|}{N} \tag{7}$$

where f_{ref} is the force reference to be tracked, f_k is the actual force at time step k, and N is the number of samples. A first set of controllers were evolved using only this criterion. By doing that, a controller with fast response is obtained. On the other hand, there are situations where stability on the contact is of outmost priority. To include this additional measure on the evolution of the controller, the following criteria was used for a second series of controllers. The contact stability criterion described in (Surdilovic (1996)) is applied on each individual in order to be selected as final solution. This criterion ensures that the contact with the environment is stable and no oscillations occur at the contact. A significantly overdamped impedance behaviour is required to ensure a stable contact with a stiff environment. If a relative damping coefficient is defined such as

$$\xi_T = \frac{D_T}{2\sqrt{M_T K_T}} \tag{8}$$

and the stiffness ratio is defined as

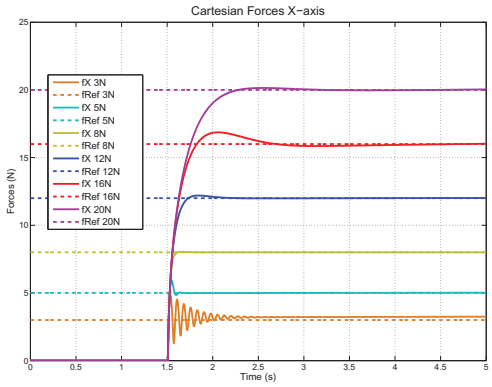
$$\kappa = \frac{K_E}{K_T} \tag{9}$$

where K_E is the stiffness of the environment, then to ensure contact stability we have to satisfy the following criterion:

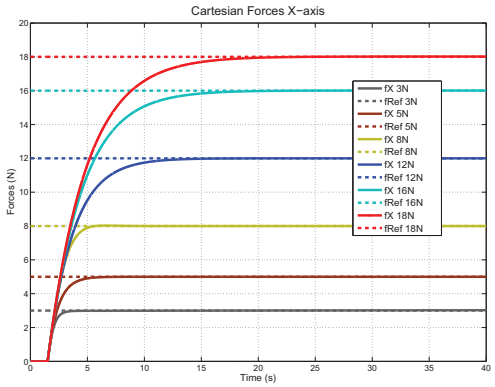
$$\xi_T > 0.5(\sqrt{1+2\kappa} - 1) \tag{10}$$

CMA-ES was initialised to start the search at [0.5,0.5,0.5], initial vector for M_T, D_T, and K_T, respectively. The initial global-step size for CMA-ES was set to $\sigma_{(0)} = 0.5$ and the system was evaluated 1000 times. The population size was chosen according to $\lambda = 4 + \lfloor 3\ln(n) \rfloor$, where n is the number of parameters to optimize and the parent number was chosen to be $\mu = \lfloor \lambda/4 \rfloor$.

A series of single-force reference controllers were evolved under this setup. Each of these controllers obtained as a result of the evolutionary process the optimal weight values for a given force reference. Figure 5(a) shows the results of the controllers evolved without being strict on the contact stability, whereas Figure 5(b) shows the results where the controller has to obey the condition given by Eq. (10). Clearly, the latter offers a safer response at the price of making the system slower.
To summarise, each single-force controller possesses three weights and their optimal values are found for a particular reference force. In a given scenario, the evolved controller is able to control the interaction forces to the desired value and with the desired dynamical characteristics. Provided the current state-of-the-art on selecting the impedance parameters, this solution is a novelty in terms of providing a simple methodology to obtain the optimal impedance parameters for a given task.



(a)



(b)

Fig. 5. Responses of the single-force controllers evolved with CMA-ES for different force reference inputs: (a) without contact stability criterion, (b) with contact stability criterion

3.2 Generalised force tracking controller

In this section, a more general force-tracking controller is designed that is able to adapt to different force references. To attain this goal, an additional block is added to the control scheme: the *Paramater Estimator*, a module that will generate estimations for $\widehat{M}_T$, $\widehat{D}_T$, and $\widehat{K}_T$ based on the current reference force. The complete control scheme can be seen in Fig. 6. The force-to-weights data sets obtained in the previous section (Fig. 5(a)) were used to generate a function that estimates the weights for the controller for any given force reference. By doing that, the input space of the force controller is generalised. Using a 6-th order polynomial as in Eq. (11) for each parameter (M_T, D_T, K_T), a function is generated that estimates the particular parameter for a given input force reference.

$$\widehat{y} = \sum_{i=0}^{n} a_i (f_{ref})^i \tag{11}$$

where $\widehat{y} = \{\widehat{y}_M, \widehat{y}_D, \widehat{y}_K\}$, are the estimation functions for each of the three parameters (M_T, D_T, K_T), respectively, and $n = 6$. The optimal coefficients a_i are again obtained using the CMA-ES evolutionary strategy.

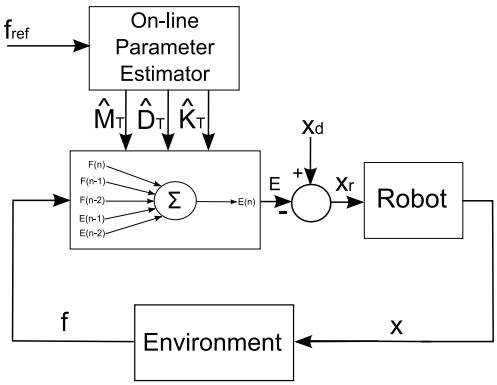


Fig. 6. Structure of the complete control scheme

The procedure is the following: CMA-ES is given the polynomial structure as in Eq. (11) and a set of force-weights training points. These points are the ones depicted in Fig. 7 for each of the parameters (inertia, damping, and stiffness) and relate an input force k with an output parameter. The vector k of input forces was $k = \{3,5,8,12,16,20\}(N)$. Note that for the sake of clarity, damping and stiffness curves have been appropriately scaled in order to be shown on the same graph. The task for the CMA-ES algorithm is to find the parameters of the polynomial that best fit through the corresponding training points. The result is a function that estimates the inertia, damping, and stiffness coefficients for any given reference force. Thus the controller will adapt its weights dynamically as the force reference requirements change. As shown in Fig. 7, the estimated curves precisely pass through the training points (the *measured* force-to-weights relationships). CMA-ES was set to stop the search for the optimal a_i coefficients when the error between the training points and the values of the curves at force k was below $1 \cdot 10^{-10}$.

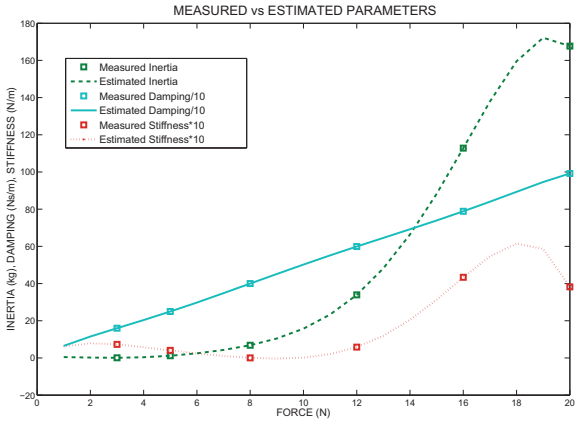


Fig. 7. Estimation functions for each of the parameters of the impedance controller

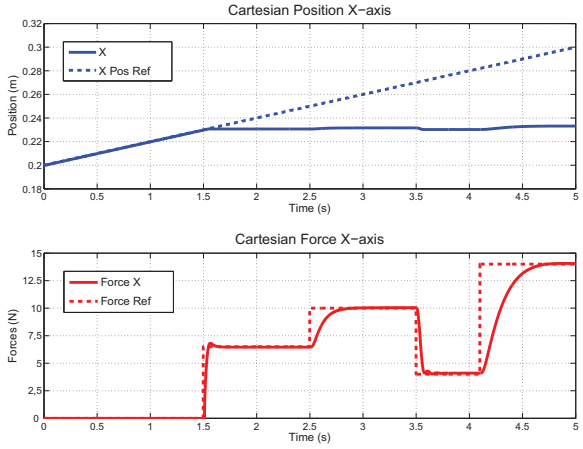
4. Experiments and Results

A series of experiments were conducted using a simulated two-link planar robotic arm (*Two-link Arm Dynamical Model* in Fig. 1) to test the performance of the ANN-based impedance controller. The robot's mechanical model is composed of two revolute joints and two bodies. The module receives torques as inputs and outputs joint angles. The masses are considered to be concentrated at the end of each link to simplify the modelling tasks. The lengths of the body links were set to $a_1 = a_2 = 0.2m$, and their masses to $m_1 = m_2 = 10kg$. A dynamic model-based controller (*Dynamical Controller* in Fig. 1) is used to cancel-out the non-linearities present on the dynamic model of the robot and to decouple the system. After this linearisation and decoupling process, a simple linear PD controller can be used to control the joint positions. The parameters K_p and K_v of the PD controller were set to $K_P = 10000N/m, K_V = 100Nms/rad$. The environment (*Environment* in Fig. 1) is modelled following a non-linear Hunt-Crossley relation (Diolaiti et al. (2005)) instead of the classical linear Kelvin-Voigt model (or spring-like model) since it achieves a better physical consistency and allows to describe the behaviour of both stiff and soft objects. Moreover, it is computationally simple to be computed on-line. The model obeys the following relation:

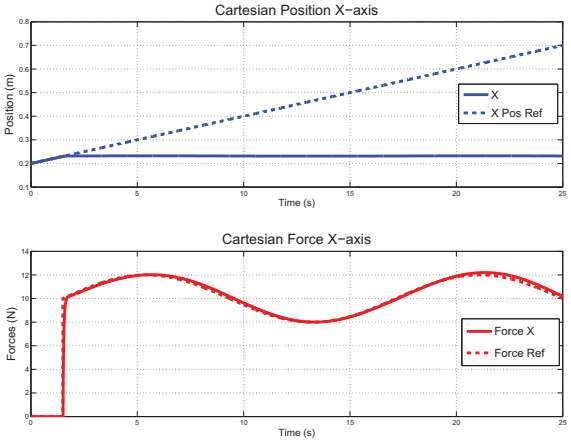
$$F(t) = kx^n(t) + \lambda x^n(t)\dot{x}(t), x \geq 0 \tag{12}$$

where n is a real number that takes into account the geometry of the contact surfaces. For these experiments, the environmental parameters were set to $k = 250N/m$, $n = 0.5$, and $\lambda = 0.0072$. For all the experiments, the robot is commanded to follow a desired position trajectory in the Cartesian space: $x(t) = 0.02t + 0.2$. This desired trajectory will be eventually modified by the impedance controller to create a new reference trajectory that complies with the current force requirements.

4.1 Response to changes on force reference



(a)

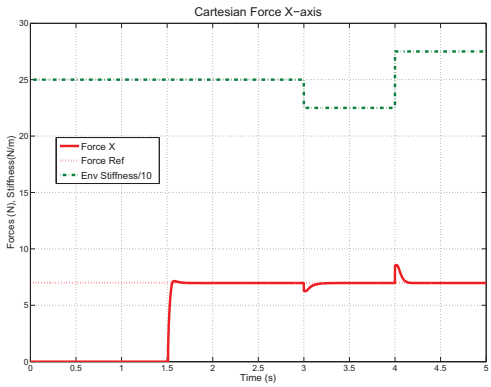


(b)

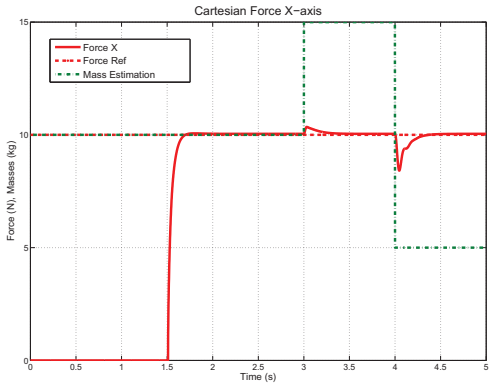
Fig. 8. Robot's response to changes on the reference force: (a) step response, (b) sinusoidal reference

To test the performance of the controller when dealing with force reference changes, two experiments were conducted. A first experiment presents multiple step changes on the reference force for the controller (Fig. 8(a)). The upper part of the figure shows the Cartesian position on the X-axis for the tip of the robot. The robot moves along that axis until it contacts a wall, placed at $x_e = 0.23m$. The bottom part of the figure shows the robot's force re-

sponses. The reference force after contacting the environment is modified and set sequentially to $\{6.5,10,4,14\}(N)$. Note that none of these values were used in the designing phase of the controller (Fig. 7). The robot is able to switch accurately between force references while keeping both a nearly-zero steady-state force error and a stable contact with the surface.
A second experiment was performed where the force reference is a sinusoidal signal (Fig. 8(b)). In this case, a sinusoidal waveform of amplitude 2N is superimposed to the reference of 10N, i.e., the reference force to be tracked is $f_{ref} = 10 + 2sin(\pi t)\ (N)$. As it can be seen on the bottom part of the figure, the robot tracks the sinusoidal force reference accurately.



(a)



(b)

Fig. 9. Robustness of the controller: (a) change on stiffness of the environment, (b) change on mass of the robot's links

4.2 Robustness against uncertainties

The following experiments aimed at testing the robustness of the controller for changes on both the environment and the robot's model. A robust controller has to be able to cope with uncertainties, especially those related to uncertainties in the parameters of the models.

4.2.1 Variations on the environmental stiffness

The first experiment modifies the stiffness of the environment during a stable contact. As previously stated, the stiffness of the environment in the Hunt-Crossley model was set to $k = 250N/m$. For this experiment, the stiffness is modified as $k_m = 250 \pm 10\%k$ (N/m). Figure 9(a) shows the behaviour of the controller in consequence of the changes on the environmental stiffness. The robot is able to recover and set back to the original force reference of 7N in a short time, despite of the fact that the stiffness is kept constant to a value below or over the nomimal.

4.2.2 Variations on the robot's model

A second experiment was conducted where the masses of the links of the robot were modified during a contact situation. As previously stated, the masses of the robot's links were set to $m = m_1 = m_2 = 10kg$. For this experiment, the estimated masses used on the dynamical model of the robot are modified as $m_m = 10 \pm 50\%m$ (kg). Figure 9(b) shows the behaviour of the controller to the changes on links' masses. The robot is again able to recover and set back to the original force reference in a short time, despite of the fact that the masses are kept constant to a value below or over the nomimal.

4.3 Robustness against noise

A final series of experiments aimed at testing the controller against the inherently-present measurement noise, especially important in the force measurement. The purpose of these experiments is twofold: on the one hand, to test whether the algorithm is able to find a solution using real-world noisy signals and, on the other hand, to enhance the evolved controller with a zero-delay noise filter using a Kalman filter. The filter is included on the evolution process in order to generate a one-step solution that takes into account noisy signals. In other words, the optimal parameters of the Kalman filter will be searched using the CMA-ES evolution strategy while simultaneously the controller's parameters are learned.

The Kalman filter (Kalman (1960)) estimates the state of a linear dynamical system that is perturbed by a gaussian noise. Formally, the filter addresses a general problem of estimating the true state $x \in \mathcal{R}^n$ of a discrete linear time system governed by

$$x_k = A_k x_{k-1} + B_k u_{k-1} + w_{k-1}, \tag{13}$$

where A_k is an $n \times n$ state transition matrix, B_k is an $n \times m$ control input model matrix, $u_k \in \mathcal{R}^m$ is the control vector, and w_k is the process noise which is assumed to be drawn from a zero-mean multivariate normal distribution with covariance matrix Q_k of size $n \times n$. The measurment (observation) $z_k \in \mathcal{R}^l$ of the true state is modelled by

$$z_k = C_k x_k + v_k, \tag{14}$$

where C_k is an $l \times n$ matrix representing the measurement model and v_k is the measurment noise which is again assumed to be drawn from a zero mean multivariate normal distribution with covariance matrix R_k of size $l \times l$.

The Kalman filter recursively estimates the current state based on the current measurement and the estimate from the previous state. The filter has basically two distinct phases: *predict* and *update*. Let $P_{k|k-1}$ and $\hat{x}_{k|k-1}$ be the *a priori* estimate of the error covariance matrix and the true state at timestep k, respectively, and $P_{k|k}$ and $\hat{x}_{k|k}$ be the *a posteriori* estimate of the error covariance matrix and the true state at timestep k, respectively. The filter starts with initial estimates for the true state $\hat{x}_{k-1|k-1}$ and the error covariance matrix $P_{k-1|k-1}$, and then

repeatedly executes its *predict* and *update* phase routines. Refer to (Welch & Bishop (1995)) for a more detailed introduction to the Kalman filter.

4.3.1 The Steady-state Kalman Filter with Constant Velocity Model

The Kalman filter used in our implementation is a particular type of the general Kalman filter in which a constant velocity model is assumed. The constant velocity model is usually used in tracking applications (Kalata (1992); Bar-Shalom et al. (2001); Perez-Vidal et al. (2009)) and is also known as an $\alpha\beta$ filter. Since we assume that the system's velocity does not change dramatically, we are able to assume a constant velocity model. The steady-state version of the Kalman filter is used in cases where the time required to compute the algorithm is an important constraint. For a given system, one can let the Kalman filter run for several cycles and record the Kalman gains K in steady state. These will be constant, so the computation can easily be sped up by always using these constants instead of updating K each cycle (which requires a matrix inversion computation). The equations that describe the steady-state Kalman filter are:

$$\hat{x}_{k|k-1} = A \cdot \hat{x}_{k-1|k-1} \quad (15)$$

$$\tilde{y}_k = z_k - C \cdot \hat{x}_{k|k-1} \quad (16)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K \cdot \tilde{y}_k \quad (17)$$

where $\hat{x}_{k|k-1}$ represents the estimate of x at time k given observations up to and including time $k-1$. z_k is the measurement at time k, A is the state transition matrix, C is the output array and K is the steady-state Kalman gain. Expression (16) computes the innovation factor that allows the predictions to be updated after new measurements have been obtained. Given the assumption of a constant velocity model, the filter will choose two weighting coefficients (α and β) that will weight the differences between predictions and new measurements when updating the current prediction to find a new estimate. To better illuminate this, consider the classical tracking equations for the $\alpha\beta$ filter:

$$x_p(k) = x_s(k-1) + v_s(k-1)T \quad (18)$$

$$v_p(k) = v_s(k-1) \quad (19)$$

$$x_s(k) = x_p(k) + \alpha(z_k - x_p(k)) \quad (20)$$

$$v_s(k) = v_s(k-1) + (\beta/T)(z_k - x_p(k)) \quad (21)$$

where $x_p(k)$ and $v_p(k)$ are the predicted position and velocity at time k, $x_s(k)$ and $v_s(k)$ are the smoothed position and velocity at time k, T is the sampling time, and α and β are the weighting coefficients. After calculating $x_p(k)$ and $v_p(k)$ (Eqs. 18 and 19), the calculation of the smoothed parameters only requires the proper selection of values for α and β. The optimal values for α and β have been derived by (Kalata (1992)), and depend on the assumed variance of both measurement and process noises (σ_v and σ_w):

$$\gamma = \frac{T^2 \cdot \sigma_v}{\sigma_w} \tag{22}$$

$$r = \frac{4 + \gamma - \sqrt{8 \cdot \gamma + \gamma^2}}{4} \tag{23}$$

$$\alpha = 1 - r^2 \tag{24}$$

$$\beta = 2 \cdot (1 - r)^2 \tag{25}$$

$$K = \begin{bmatrix} \alpha \\ \beta / T \end{bmatrix} \tag{26}$$

The state transition matrix A is initialized the with a constant velocity model:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{27}$$

and the output array C is

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{28}$$

where the '1' in the first column indicates that we have measurements from the force, and the '0' in the second column indicates that we have no information about the force change with respect to time (derivative of the force).

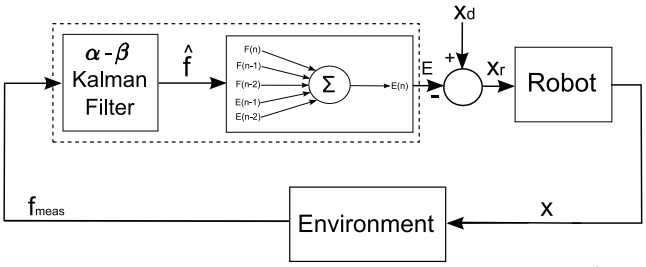
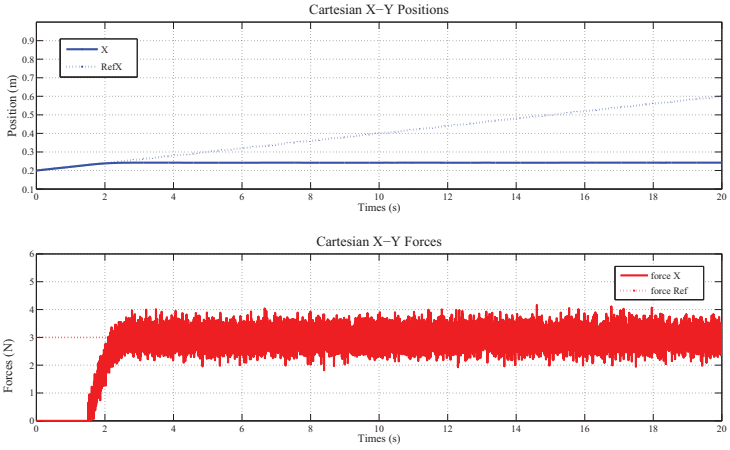


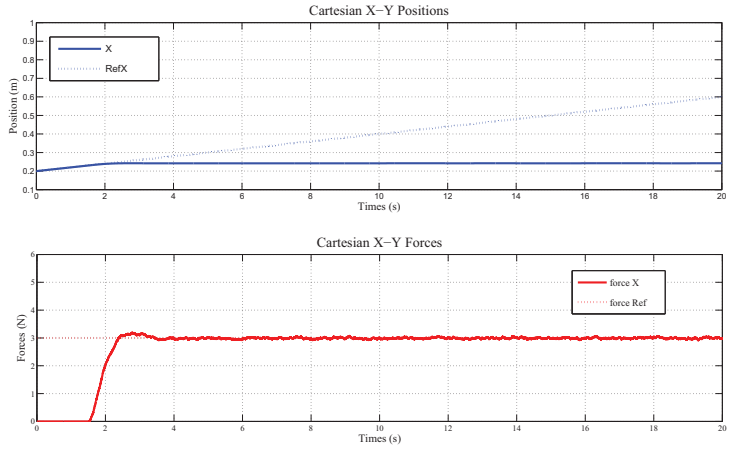
Fig. 10. The $\alpha - \beta$ Kalman filter is used to estimate the sensor value $\hat{f}$ from the measured (noisy) value f_{meas}. The parameters of the blocks enclosed under the dotted lines are obtained using evolution strategies

In our experiment, we use one Kalman filter for the force measurement. The measurement noise that is introduced to the system is a Gaussian signal with zero mean and standard deviation $\sigma_v = 10^{-1}$. This noise is added to the input f_{meas} (the force measurement) of the Kalman filter depicted in Figure 10. In this experiment, the same neural network structure was used as in the previous experiments, i.e. the one-neuron feedforward neural network with 5 inputs, 1 output, and 3 weights. Additionally, the optimization of the Kalman filter was incorporated into the evolutionary process, where optimal values for the parameters σ_v and σ_w were searched for using CMA-ES (along with the weights for the neural network). Because the problem is simulated, the standard deviation of the measurement noise we are introducing is known and thus the initial value for σ_v in the Kalman filter can be set to this value. In the case of a real system, however, a set of real measurements could have been collected, the mean and standard deviation of the data set calculated, and the standard deviation used as the value

for σ_v. In the case of process noise, manual tuning is typically used due to the complexity of determining the value of the noise. The Kalman filter, however, usually performs well with only a rough estimate of σ_w.



(a)



(b)

Fig. 11. Robustness against noise: (a) evolving the controller without a Kalman filter, (b) weights of the ANN-based controller and the Kalman filter evolved simultaneously

Figure 11 shows the results of the experiments with noisy signals. Figure 11(a) depicts the case of learning to track a specific force reference with a highly-noisy force measurement. As it can be seen, the algorithm is able to, despite the noise, learn a proper solution in order to

achieve the reference force. However, the controller would be useless in a practical scenario since the robot would oscillate at high frequency around the contact point. In order to provide a compact solution, the Kalman filter presented previously is included in the evolution process. By doing that, we obtain a solution in one step: both the weights of the neural network and the parameters of the Kalman filter are obtained simultaneously, without requiring of a pre-processing of the measurement data. Figure 11(b) shows the response obtained with the system depicted on Figure 10. The robot is able to reach the targetted reference force and, at the same time, imperceptible noise remains on the force response of the robot, i.e. no oscillations occur on the contact.

5. Conclusions

The work presented describes the design of an ANN-based impedance controller by using evolutionary techniques. The impedance controller is first discretized and represented as a neural network. The use of evolutionary techniques provides a simple methodology to evolve the controller requiring only the definition of a proper performance criteria to be optimised. Currently, unclear or cumbersome methodologies are found to select impedance parameters. The proposed approach obtains optimal parameters given a task to perform. Besides, it is shown how the classical impedance controller can be described as a single-neuron neural network with 5 inputs, 3 weights, and 1 output. Since the weights of the neural network bound the policy space of the controller, and in this case they are only three, the space of the possible inputs is unique. To generalise the controller for any given force reference input, an on-line estimator has been designed that estimates the weights for the current force reference. Using the values of a series of single-force controllers, the parameters of a polynomial are obtained that estimate the proper neural network weights for the current scenario. The resulting controller is able to track a great range of force reference inputs, a quality that is not intrinsically present on a classical impedance controller. Moreover, the robustness of the controller is demonstrated by modifying both the robot and the environmental model parameters. The controller is able to set back to the current reference force after abrupt changes on the environmental stiffness, even when it is constantly kept to values 10% below or over the nominal one. Similarly, abrupt changes on the estimated masses of the robot links of up to 50% of the nominal value are absorbed by the controller, which is able to keep track of the current reference force. Finally, the controller is enhanced with a Kalman filter to improve the controller's robustness against the measurement noise. Both the controller and the parameters of the Kalman filter are evolved simultaneously, thus providing a one-step solution which does not require a pre-processing of the measurement data used to learn the solution.

6. References

Bar-Shalom, Y., Li, X. & Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, USA.

Caccavale, F., Natale, C., Siciliano, B. & Villani, L. (2005). Integration for the next generation - embedding force control into industrial robots, *IEEE Robotics and Automation Magazine*, pp. 53–64.

Chiaverini, S., Siciliano, B. & Villani, L. (1999). A survey of robot interaction control schemes with experimental comparison, *Mechatronics, IEEE/ASME Transactions on* **4**(3): 273–285.

De Schutter, J. & van Brussel, H. (1988). Compliant robot motion II: A control approach based on external control loops, *Int. J. Rob. Res.* **7**(4): 18–33.

Diolaiti, N., Melchiorri, C. & Stramigioli, S. (2005). Contact impedance estimation for robotic systemss, *IEEE Transactions on Robotics*, pp. 925–935.

Erol, D., Mallapragada, V. & Sarkar, N. (2005). Adaptable force control in robotic rehabilitation, *IEEE Int. Workshop on Robots and Human Interactive Communication*, pp. 649–654.

Hogan, N. (1985). Impedance control-an approach to manipulation, *Journal of Dynamics Systems, Measurement, and Control-Transactions of the ASME* **107**: 8–16.

Jung, S. & Hsia, T. (1998). Neural network impedance force control of robot manipulator, *IEEE Transactions on Industrial Electronics*, pp. 451–461.

Kalata, P. R. (1992). Alpha-beta target tracking systems: A survey, *American Control Conference*, pp. 832–836.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Transactions of the ASME-Journal of Basic Engineering* **Series D**: 35–45.

Lopes, A. M. & Almeida, F. (2006). Acceleration based force-impedance control, *MIC'06: Proceedings of the 25th IASTED international conference on Modeling, indentification, and control*, ACTA Press, Anaheim, CA, USA, pp. 73–81.

Lu, W.-S. & Meng, Q.-H. (1991). Impedance control with adaptation for robotic manipulations, *IEEE Transactions on Robotics and Automation*, pp. 408 – 415.

Matko, D., Kamnik, R. & Badj, T. (1999). Adaptive impedance force control of an industrial manipulator, *Proc. IEEE Int. Symposium on Industrial Electronics*, pp. 129–133.

Paul, R. & Shimano, B. (1976). Compliance and control, *Proceedings of the 1976 Joint Automatic Control Conference*, pp. 694–699.

Perez-Vidal, C., Gracia, L., Garcia, N. & Cervera, E. (2009). Visual control of robots with delayed images, *Advanced Robotics* **23**: 725–745.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart.

Salisbury, J. K. (1980). Active stiffness control of a manipulator in cartesian coordinates, Vol. 19, pp. 95–100.

Schutter, J. D., Bruyninckx, H., Zhu, W.-H. & Spong, M. W. (1997). Force control: A bird's eye view, *In B. Siciliano (Ed.), Control Problems in Robotics and Automation: Future Directions*, Springer Verlag, pp. 1–17.

Schwefel, H.-P. P. (1993). *Evolution and Optimum Seeking: The Sixth Generation*, John Wiley & Sons, Inc., New York, NY, USA.

Seraji, H. & Colbaugh, R. (1993). Adaptive force-based impedance control, *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1537 – 1544.

Surdilovic, D. (1996). Contact stability issues in position based impedance control: Theory and experiments, *Proc. IEEE ICRA96* pp. 1675–1680.

Welch, G. & Bishop, G. (1995). An introduction to the Kalman filter, *Technical Report TR95-041*, University of North Carolina at Chapel Hill, Department of Computer Science, USA.

Whitney, D. (1987). Historical perspective and the state-of-the art in robot force control, *International Journal of Robotics Research* **6**: 3–14.

Formal Methods in Factory Automation

Corina Popescu and Jose L. Martinez Lastra
Tampere University of Technology
Finland

1. Introduction

The world market share for European industries targeting capital intensive products and equipment for manufacturing is only 22% (Manufuture, 2004). This position need not only be secured, but improved – to have the world standards of manufacturing made and approved in Europe.
In EU, each job in manufacturing is linked to two jobs in services. To support competitiveness of its industries in the global economy, Europe must be a leader in manufacturing technologies, at both process control and coordination control level. Having the highest-tech equipment in factories is necessary but not sufficient to achieve high production effectiveness. Research is needed to assist the devices cooperate (optimally) reducing waste caused by loss of energy/material and inefficient processes, while ensuring correct design and execution of standalone processes. Formal methods have an important potential to assist the development of feasible solutions in this sense.
This chapter is an introduction to formal methods in factory automation. Far from being an extensive review of the state of the art, this work provides a structured start-point for the newcomers to the field, stressing pointers to some of the most relevant works in the area. The chapter is organized as follows: Section 2 presents and compares significant features of three formalisms, leaving out specific usages of these formalisms in particular scenarios. Section 3 describes the use of formal methods in factory automation for two main purposes: verification/validation/synthesis of software control, and coordination of manufacturing activities. Section 4 presents a summary of the discussed topics and conclusions.

2. Formalisms: Overview and comparison

This section is focused on discussing relevant features and main strengths/weaknesses of three formalisms: Timed Automata, Process Algebras and Petri Nets. The intention is to leave out the specific usages of these formalisms in particular scenarios.
A timed automaton (Alur & Dill, 1994) is a finite automaton with a finite set of real-valued clocks. The clocks can be reset to zero independently of each other. The role of each clock is to keep track of the time elapsed since the last reset. The choice of the next state transition depends on the input symbol and its time relative to the times of the previously read symbols. The complexity of describing concurrent systems (especially their interactions)

with automata is high. In UPPAAL (UPPAAL), for instance, interactions can be represented through synchronization channels or guards.

Fig. 1. Conveyor-robot transfer

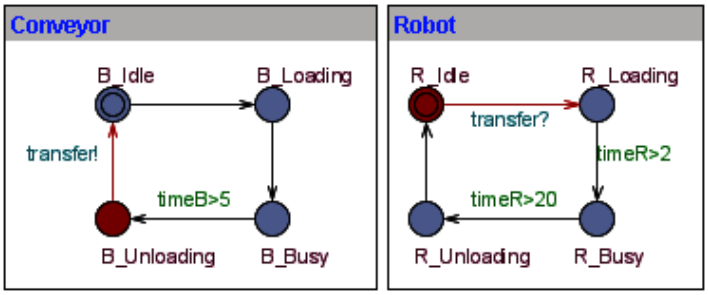


Fig. 2. Automata example: Conveyor-robot transfer

Figure 2 illustrates a simple automata representation of the transfer of a part from a buffer/conveyor of one location and a robot (Figure 1). State transitions depend on elapsed times (modelled through the clocks timeB and timeR). The interactions (e.g. the conveyor-robot transfer) are expressed through the synchronization channel 'transfer'.

The verification problem is an inclusion problem of the languages accepted by the implementation and the specification automata. The automata-theoretic approaches to verification have drawbacks related to the needed computational space and time. Even if there is enough space to store the specification and implementation automata separately, after computing the synchronized product the size of the representation can become too large. The size of the representation influences proportionally the execution time as well.

A process algebra is the study of the behaviour of a system by algebraic/axiomatic means. It is similar to the notion of a group, in the sense that both are mathematical structures having operators that satisfy a set of axioms (the equational theory) of the structure. The main derivatives of process algebra are CCS (Calculus of Communicating Systems) (Milner, 1980), CSP (Calculus of Sequential Processes) (Hoare, 1978) and ACP (Algebra of Communicating

Processes) (Bergstra & Klop, 1984). An important extension of CCS is Pi-calculus (Milner et al., 1989), which was developed to address mobility and dynamic link configuration between processes.
The details on some of the operators within the specification stand at the core of the distinctions between the various derivatives of process algebras (Philippou & Sokolsky, 2008). For instance, CCS, CSP and ACP all incorporate a different view of the synchronization-related data of the parallel composition operator. Hiding an action in ACP prevents the action from taking place altogether. Applying the same operator on a set of actions in CCS prevents the actions from taking place on the interface with the environment, but not within the system.
The view on the concurrency relation is another aspect that distinguishes certain process algebras from other formalisms. For instance, CCS approximates parallel behavior by interleaving executions. It is assumed that a system is fully described from the point of view of an external observer. Observation is made possible through the communication that takes place between the observer and the observed system. Since communication can only take place in a sequential order between the participants, the external observer can make only one observation at a time. This implies that when composing two agents in CCS, their actions are treated as occurring in arbitrary order but not simultaneously. This idea is reflected mathematically in the expression of the expansion law for CCS.
As opposed to model checking, the verification technique supporting process algebras is equational reasoning. The axioms of the algebra are used to determine the equivalence of two processes.
A Petri Net (PN) (Murata, 1989) consists of places, transitions, and flowarcs that connect places with transitions. Figure 3 illustrates a simple PN representation of the transfer of a part from a buffer/conveyor of one location and a robot (Figure 1). The elements of this net are: places (P={p1,p2,p3,p4,p5}), transitions ({t1,t2,t3,t4}) and flowarcs ({(p1;t1), (p2,t2), (p3,t2), (p4,t3), (p5,t4), (t2, p1), (t2, p4), (t1, p2), (t3, p5), (t4, p3)}). The idle statuses of the buffer and of the robot are modeled through places p1 (B_idle) and p3 (R_idle). Places p4 and p5 model the start and stop of robot processing. Place p2 (B_busy) represents the situation in which a part is available in the buffer/on the conveyor.

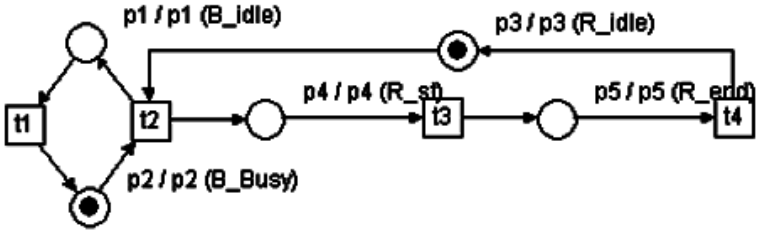


Fig. 3. PN Example: Conveyor-robot transfer

Each place may contain tokens. In the shown example, tokens are available in p2 and p3 (i.e. the marking of each of these places is 1: m(p2)=1 and m(p3)=1). A transition is enabled if each of its input places contains at least one token (e.g. in Figure 3, transition t2 is enabled as all its input places - p2 and p3 - hold one token).

If enabled, transitions act on input tokens by a process known as firing. The firing of a transition results in consumption of the tokens from its input places and the processing of some task. At the same time, a specified amount of tokens is added into each of its output places. In the shown example, the firing of t2 corresponds to the transfer of pallet between the buffer/conveyor B and the robot R. After firing, m(p2)=m(p3)=0 (i.e. a part is no longer available in buffer, and the robot is no longer free to receive part) and m(p4)=1 (i.e. the robot starts processing). State (marking) evolution is reflected in the firing of transitions.
The flow of tokens within a PN can be fully described algebraically. Table 1 illustrates the incidence matrix W for the PN example of Figure 3.

t1	t2	t3	t4	W
-1	+1	0	0	p1
+1	-1	0	0	p2
0	-1	0	+1	p3
0	+1	-1	0	p4
0	0	+1	-1	p5

Table 1. Incidence Matrixof the PN in Figure 3

This is a marking-independent description of the structure of the net. Columns correspond to places and rows correspond to transitions. Negative matrix elements are associated with place-transition flowarcs (e.g. W[p1][t1]=-1: there exists a flowarc from p1 to t1 in the described PN). Positive matrix elements are associated with transition-place flowarcs (e.g. W[p2][t1]=-1: there exists a flowarc from t1 to p2 in the described PN).
The marking M obtained by firing of a pre-specified sequence of transitions Stransitions from an initial marking M_{start} is mathematically describable through the fundamental equation:

$$M=M_{start}+W\cdot S \quad (1)$$

where S is the characteristic vector (of size equal to the number of transitions in the PN) associated with the sequence $S_{transitions}$. A firing sequence $S_{transitions}$={t1,t3,t2,t3} in a net with four transitions corresponds to a characteristic vector S=[1,1,2,0]. The first element of the vector corresponds to t1, which appears once in $S_{transitions}$. The third element of S corresponds to t3, which appears twice in $S_{transitions}$. t4 $\notin S_{transitions}$, therefore S[4]=0. Finally, S[2]=1 as transition t2 does appear once in $S_{transitions}$.
For the PN example of Figure 3, the firing effect of t2 can be calculated based on the fundamental equation:

$$\begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\\0\end{bmatrix} + \begin{bmatrix}-1&+1&0&0\\+1&-1&0&0\\0&-1&0&+1\\0&+1&-1&0\\0&0&+1&-1\end{bmatrix} \cdot \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

All states of the system can be derived algebraically. Figure 4 illustrates the reachability graph (state space) of the conveyor-robot system shown in Figure 1.

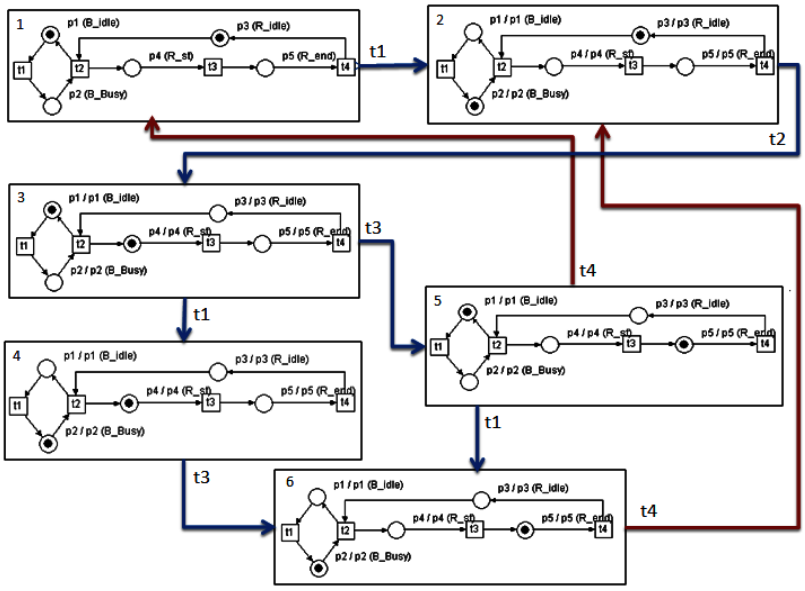


Fig. 4. Reachability graph example, shows the dynamic behaviour expressible through a PN

Several qualitative properties can be checked with PNs: liveness, boundedness, safeness, reversibility, etc. (Murata, 1989). If satisfied, the liveness property expresses potential fireability in all future markings (i.e. for every reachable state, the model can evolve in such a way that every transition can always fire in the future). A system described by a live PN is therefore a system in which every activity can ultimately be carried out.

The firm mathematical foundation confers the PN formalism a powerful set of analysis tools. Analysis methods of Petri Nets are either enumeration-based or net-driven:

Enumeration techniques rely on computation of the reachability graph. State of the art model checkers can handle state spaces up to 10^9 states with explicit state enumeration (Baier & Katoen, 2008). However, the size of the state space grows exponentially in the number of represented objects because of concurrency and interleaving semantics used to represent any sequence of possible actions. This key problem is addressed through clever algorithms and data structures (for some specific problems, state spaces of sizes 10^{20} up to 10^{476} have been handled successfully (Baier & Katoen, 2008)). Several types of methods have been researched to make enumeration-based analysis applicable in an industrial context. State-based techniques (BDDs and on-the-fly verification [Clarke et al., 2001]) aim to efficiently manage the construction of the reachability graph. Partial-order methods (sleep sets, stubborn sets and unfoldings (Girault & Valk, 2003)) make use of the dependency relations between system events to compact the state space.

Net-driven techniques aim to obtain useful information about system behaviour, reasoning from the structure of the net and the initial marking. Generative families of flows characterizing PNs are assessable based on graph theory and linear algebra techniques. Linear invariants (computable only for underlying PN models) enable certain properties of the reachable markings and firable transitions to be characterized irrespective of evolution.

Petri Nets have formal semantics, a graphical nature and an explicit representation of states (Aalst, 1998). Performance measures such as response/waiting times and occupation rates can be easily computed for PN-based models. Unlike other formalisms (e.g. CCS), PNs are capable of expressing simultaneous execution and non-determinism easily. Finally, although the verification stage does not have to be PN-based, it can benefit from the PN-nature of the model (Girault & Valk, 2003).
Table 2 summarizes the main outlined points of this section.

		Formalism		
		Timed Automata	**Petri Nets**	**Process Algebras**
Modelling issues	Describing concurrency	High complexity	**++**	**Interleaving approximation (CCS)**
	Graphical nature	+	**++**	**-**
	Explicit representation of states	-	**+**	**-**
Verification issues	Verification methods	Crossproduct; checking language inclusion	Model checking; verification does not need to be PN-based	**Equational reasoning**
	Verifiable properties, specific to the formalism		**Invariance**	

Table 2. Distinctive features of formalisms, from the modelling and verification viewpoints

3. Using formal methods in factory automation

In factory automation, formal system representations are used for two main purposes: to verify/validate/synthesize software control, and to coordinate manufacturing activities.

3.1 Verification/Validation/Synthesis of software control

The utilization of formal methods for the synthesis and verification of process logic control has arisen as an alternative to the testing of direct implementations of control realizations against informal specifications. The formalized descriptions of the control objectives, the synthesized/reinterpreted control algorithm and (sometimes) the formal model of the uncontrolled plant are input to verification and validation procedures (Figure 5).
Formal verification aims at investigating whether the design satisfies the identified standard requirements ("Are we building the product right?"). Unlike testing, verification can prove that a system has a certain property. Formal validation investigates whether the formal model is consistent with the informal conception of the design ("Are we building the right product?"). Unlike verification, validation cannot be fully automated, because it implies investigation of informal specification.

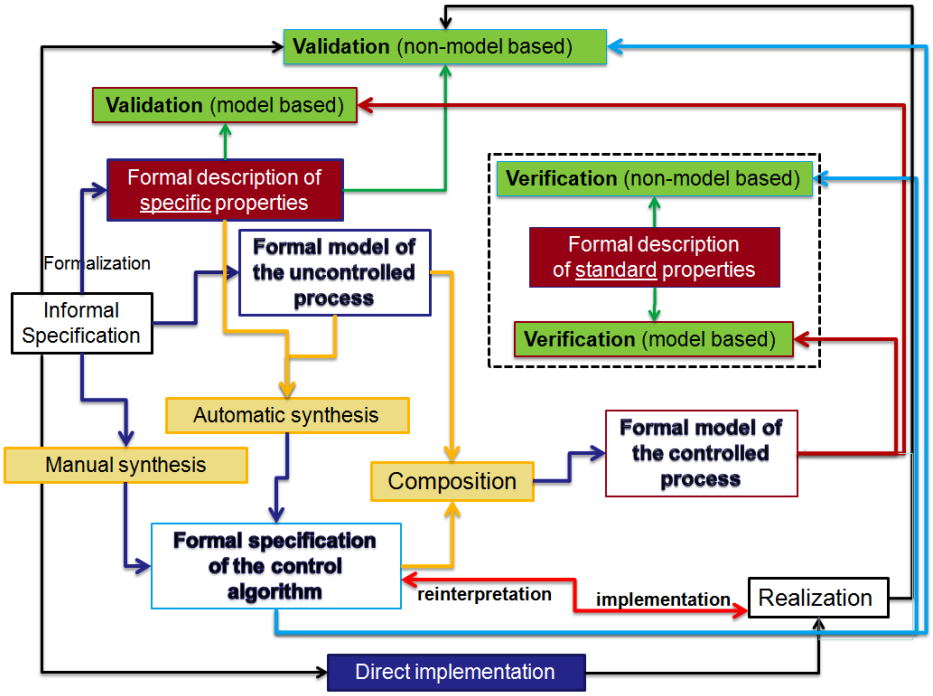


Fig. 5. Formal methods in PLC programming (adapted from (Frey & Litz, 2000))

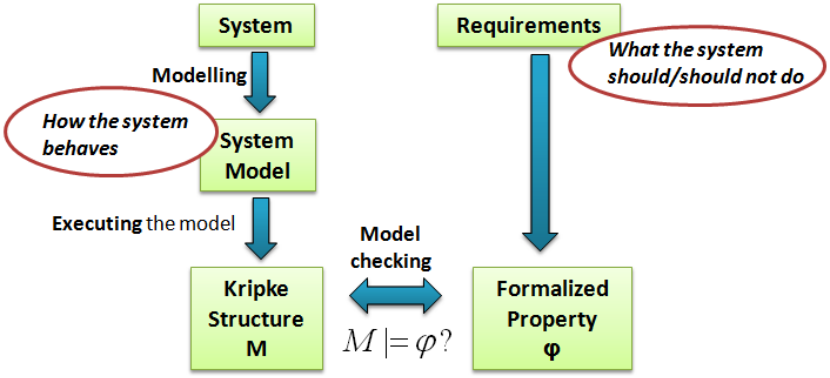


Fig. 6. Model checking

There are two main formal verification techniques: model checking (Clarke et al., 2001) and theorem proving (Duffy, 1991). In model checking (Figure 8) specifications of the system behavior (typically formulated in a temporal logic) are checked automatically on a finite model of the system (based on Petri Nets, automata, UML, etc.). The properties are investigated for given states or successions of states corresponding to the system model.

Theorem proving assumes that both the system and its expected properties are formalized in a mathematical logic. Inference rules are then applied to prove the properties from the axioms of the system description.

3.2. Coordination Control: (Re)scheduling and deadlock handling

Coordination refers to obtaining a system-level functionality based on functionalities provided by each individual component of the system. The inputs to the Coordination Control level are given by the activities to be achieved in the system (e.g. process flows, activity charts, etc.).

Planning and scheduling

Planning is deciding what actions to use to achieve some set of objectives.

A production schedule is a specification, for each resource required for production, of the planned start time and end time of each job assigned to that resource.

Scheduling is the process of creating a production schedule for a given set of jobs and resources, while optimizing some performance measure (increase of productivity, minimization of operation costs, etc.). Based on production schedules, the release of jobs to the shop can be controlled, for a better overall coordination of the activities in the manufacturing line.

Rescheduling is the process of updating an existing production schedule in response to disruptions such as machine failures and repairs, urgent job arrival, job cancelation, due date change or change in job priority.

Three main types of rescheduling strategies have been identified in the literature: completely reactive scheduling, predictive-reactive scheduling and robust pro-active scheduling:

Completely reactive rescheduling methods do not generate firm schedules in advance, but use heuristic dispatching rules to assist real time execution. Such rules are defined based on experience and are assessed through simulation, with respect to various performance criteria (e.g. tardiness, flow time, etc.). The choice of policies is problem specific, and no rule performs well for all performance criteria (Abumaizar & Svetska, 1997). Dispatching rules are used extensively in multi-agent architectures (Lee & DiCesare, 2007; Wang et al., 2008), where overall system behavior is influenced by concurrent local decisions taken by networks of individual problem solvers that cooperate. Here, heuristic guidelines come in response to the traditional drawbacks of central and hierarchical scheduling (e.g. high system complexity and cost, low fault tolerance and flexibility). Comprehensive reviews and comparative studies of such regulations in dynamic job shops and flow shops have been provided in (Rajendran & Holthaus, 1999) and (Panwalkar & Iskander, 1977).

Predictive/Reactive scheduling is an iterative process of repairing previously-created schedules (Abumaizar & Svetska, 1997; Jain & ElMaraghy, 1997) or completely regenerating schedules (Church & Uszoy, 1992). Depending on the implemented rescheduling policy, the revisions may be triggered in response to unexpected events altering the system status (event-driven), periodically, or in a hybrid manner.

Robust pro-active scheduling refers to the construction of predictive schedules which satisfy performance requirements predictably in a dynamic environment.

A wide variety of dynamic scheduling techniques have been discussed in the literature (Shukla & Chen, 1996; Stoop & Weirs, 1996; Zhou, 1995; Zhou, 1999).

Heuristics are schedule repair methods that target the finding of reasonably good solutions in short time. The main problem associated with these techniques is the difficulty to predict system performance because decisions are taken locally.
Mathematical programming techniques ignore practical constraints such as material handling capacity and complex resource sharing/routing and therefore have only a few real applications in industry (Zhou, 1995, 1999).
Meta-heuristics seek to avoid entrapment in poor local optimums obtained through local neighborhood search methods. The most popular meta-heuristic techniques include tabu search, simulated annealing and genetic algorithms (Ouelhadj & Petrovic, 2008).
Knowledge based systems, genetic algorithms (e.g. Jain & ElMaraghy,1997), fuzzy logic, case-based reasoning and neural networks have also been regarded as potential solutions to the scheduling problem.
Petri Nets can finely describe shared resources, synchronization, lot sizes and routing flexibility (Lee & DiCesare, 1994; Zhou & Jeng, 1998). PN-based scheduling implies a search for a sequence of feasible transition firings that can bring the system from an initial state to a goal state. The found schedule is deadlock free (one of the main advantage of Petri Nets over the other discussed dynamic scheduling techniques). Additionally, it is event-driven, which makes this type of scheduling perfectly suitable for real time implementation.

Deadlock handling

Deadlocks (Figure 7) are situations in which a (part of a) system remains indefinitely blocked and cannot terminate its task (Fanti & Zhou, 2004). These phenomena are caused by the inappropriate allocation of resources to concurrent executing processes.

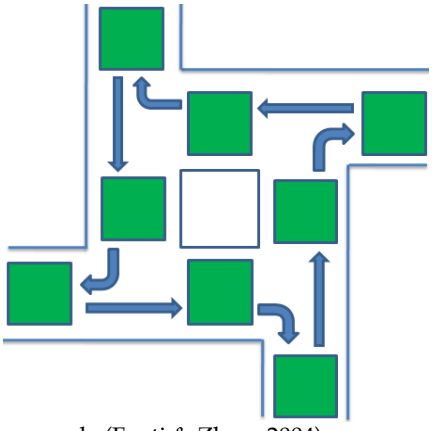


Fig. 7. Deadlock condition example (Fanti & Zhou, 2004)

Deadlocks were extensively studied first in the field of computer science and several deadlock handling techniques were originally developed for this domain. However, direct application of these methods to manufacturing systems is not possible. Computer applications only require that the bounds on the total number of resources needed by each process are known. In factory automation the information concerning the order of resource (de)allocation is also requisite (Wysk et al., 1991).

Four conditions are identified in the literature for a deadlock to occur. First, tasks claiming exclusive control of the resource they acquire may lead to deadlock (the **mutual exclusion condition**). Second, deadlock may occur when resources cannot be forcibly removed from the tasks holding them until the resources are used to completion (the **no-preemption condition**). Third, processes holding resources allocated to them while waiting for additional ones may prevent proper termination of all tasks (the **wait-for condition**). Fourth, circular claims of tasks, such that each task holds one or more resources that are being requested by the next task(s) in the claim (the **circular wait condition**), will cause indefinite blockage of a system.

In Automated Manufacturing Systems, the first three conditions always hold true. Orchestrators do claim exclusive control of the resources (machines/robots/conveyors) they acquire. Once acquired, a resource must complete the processing it was originally contracted for: a device cannot be forcibly stopped while processing in order to start machining for a different requestor. Last but not least, orchestrators hold resources allocated to them until (some of the) needed future (transportation) devices become available. Therefore, deadlocks can be excluded only if the circular wait condition is falsified.

Three main strategies have been identified for resolving deadlock problems: prevention, detection & recovery, and avoidance. **Deadlock prevention** is an offline technique involving static resource allocation policies for eliminating deadlocks. Knowledge of the system state is not required to realize the control. However with this method the utilization of resources is low and production flexibility is limited. The **detection and recovery** technique aims at resolving blockages after they have occurred. The recovery process is assisted by special buffers reserved for breaking deadlocks. This solution enables higher resource utilization, however it should be used only when deadlock is rare and detection & recovery cost is low. **Deadlock avoidance** is an online method that uses look-ahead strategies and operational control of part flow to falsify the circular wait condition. Track of the current system state and possible future states is needed. This technique is considered to yield better performance from the viewpoint of resource utilization than the first two.

Deadlock analysis and handling approaches seek the circular waits within models of process-resource interactions (job mix). The interactions between jobs and resources are traditionally represented through graphs (Wysk et al., 1991; Cho et al., 1995; Kim & Kim, 1997; Zhang & Judd, 2007) or Petri Nets (Banaszak & Krogh, 1990; Viswanadham et al., 1990; Wu et al., 2008):

Wysk, Yang and Joshi (Wysk et al., 1991) consider the deadlock problem for direct address Flexible Manufacturing Systems (FMS) during design phase. They use a graph representation of all wait relations between the input job mix and resources. All circuits within, together with their interactions , are investigated. A circuit is considered to be a deadlock if the number of jobs occupying the nodes of the cycle is equal to the numbers of nodes and edges of the cycle. The circuits are identified through a string multiplication procedure that uses one distinct character to encode each machine/node in the graph. Circuit detection is computed only upon the introducing of a new part into the system.

Cho and colleagues (Cho et al., 1995) develop graph theoretic deadlock handling procedures that are suitable for the real time control of manufacturing systems. The complete part routings of all the parts in the circuit are needed to detect impending part flow deadlocks. A system status graph is virtually updated for every part movement

before the parts move physically to the next destination. The deadlock detection and resolution procedures are based on the defined notion of 'bounded circuit' and its derivatives for this graph. A circuit becomes a sufficient condition for part flow deadlock if the number of edges in the circuit is equal to the number of parts and machines. The circuit type and its degree of node occupation characterize both part flow deadlocks and impending part flow deadlocks.

Kim (Kim & Kim, 1997) approach the deadlock avoidance problem from the graph theoretic viewpoint. Deadlock avoidance is rephrased as the problem of inserting´/deleting edges to/from the resource allocation graph while keeping it acyclic. Cycle detection on this graph is employed via a method originally developed by Belik (Belik, 1990). This technique is enriched with a resource allocation policy, effective in Automated Manufacturing Systems, to ensure superior resource utilization and productivity.

Banaszak and Krogh (1990) model concurrent job flow and dynamic resource allocation in an FMS with Petri Nets. A policy to restrict transition enabling in this model is used to avoid possible deadlocks.

Viswanadham, Narahari and Johnson (Viswanadham et al., 1990) describe a set of deadlock prevention policies that utilize look-ahead procedures on the reachability graph of the system. All behavioral characteristics of an FMS (including deadlocks) are captured offline, at modeling phase. The feasibility of the method for large systems is questionable as the entire state space of the system must be computed in its initial phase. Another important drawback concerns adaptability: if any change is made in the system the corresponding modifications have to be translated into the formal model.

Zhang and Judd (Zhang & Judd, 2008) propose a deadlock avoidance algorithm (DAA) for FMS which allows free choices in part routing. They calculate the effective free space of circuits in the digraph model of all wait relations between the resources involved in all process plans. The presented DAA runs in polynomial time once the set of necessary circuits of the digraph is computed offline.

4. Summary

This chapter provides a short introduction to the topic of formal methods in factory automation. The discussion covers the differences between two formalisms widely used in the considered application domain (Petri Nets and timed automata), and process algebras (commonly used in the field of computer science). Details are given on how formal methods are used in factory automation for verification and synthesis of process logic control and for coordination control. Pointers to relevant studies in the field are given, to provide the newcomers to the field with initial guidelines for further investigations.

5. References

Aalst, Van der, W.M.P., 'The Application of Petri nets to workflow management' Journal of Circuits, Systems and Computers, vol.8, pp. 21-66.

Abumaizar, R.J. and Svetska, J.A., 'Rescheduling job shops under random disruptions', International Journal of Production research, 1997, vol. 35(7), pp. 2065-2082

Alur, R. and Dill, D.L.(1994), A Theory of Timed Automata, Theoretical Computer Science , vol. 126, pp.183-236.

Baier, C., Katoen, J.-P., 'Principles of Model Checking', MIT Press, ISBN 978-0-262-02649-9, 2008.

Banaszak, Z.A., Krogh, B.H.(1990) Deadlock avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows, IEEE Transactions on Robotics and Automation, vol. 6, no.6, 724-734.

Belik, F. (1990), 'An efficient deadlock avoidance technique' IEEE Transactions on Computers, vol. 39, no.7, 882-888.

Bergstra, J.A., and Klop, J.W.(1984),'The algebra of recursively defined processes and the algebra of regular processes', Lecture Notes in Computer Science 172, pp.82-95.

Cho, H., Kumaran, T.K., Wysk, R.A.(1995).Graph-Theoretic Deadlock Detection and Resolution for Flexible Manufacturing Systems. IEEE Transactions on Robotics and Automation, vol. 11, no.3, 413-421.

Church, L.K., Uszoy, R., 'Analysis of periodic and event-driven rescheduling policies in dynamic shops', International Journal of Computer Integrated Manufacturing , vol. 5(3), 1992, pp. 153-163.

Clarke, E.M., Grumberg, O., Peled, D.A., "Model Checking", MIT Press, 2001, ISBN 0262032708, 9780262032704

Duffy, D. A., "Principles of Automated Theorem Proving", John Wiley & Sons, 1991.

Fanti, M.P., Zhou, M.C. (2004). Deadlock control methods in automated manufacturing systems. IEEE Transactions on Systems, Man, and Cybernetics –Part A: Systems and Humans, vol. 34, 5-22.

Frey, G., Litz L., 'Formal Methods in PLC Programming', Proceedings of SMC, pp. 2431-2436, October 2000,.

Girault, C., and Valk, R. (2003), 'Petri Nets for Systems Engineering,' Springer, ISBN 3-540-41217-4.

Hoare, C.A.R, 'Calculus of Sequential Processes' Communications of the ACM , vol. 21, pp. 666-677.

Jain, A.K, ElMaraghy, H.A., 'Production scheduling/rescheduling in flexible manufacturing', International Journal of Production Research, vol.35, no.1, pp.281-309

Kim, C.O., Kim, S.S. (1997). An efficient real-time deadlock-free control algorithm for automated manufacturing systems. Int.J. Prod. Res., vol. 35, no.6, 1545-1560.

Lee, D.Y., DiCesare, F., 'Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search', IEEE Transactions on Robotics and Automation, vol. 10, no.2, April 1994, pp.123 -132

'Manufuture: A vision for 2020', Report of the High Level group, November 2004, Directorate-General for Research, European Commission, Brussels, Belgium

Milner, R. (1980), 'A Calculus of Communicating Sequences', Lecture Notes in Computer Science, vol. 92.

Milner, R., Parrow, J. and Walker, D.(1989), 'A Calculus of Mobile Processes - Part I,' LFCS Report 89-85. University of Edinburgh.

Murata, T., 'Petri nets: Properties, analysis and applications', Proceedings of the IEEE, vol.77, no. 4, pp. 541-580, April 1989.

D. Ouelhadj and S. Petrovic (2008), 'A survey of dynamic scheduling in manufacturing systems', Journal of Scheduling

Panwalkar, S.S., Iskander, W., 'A survey of scheduling rules', Operations Research , vol. 25, no.1, Jan.-Feb. 1977, pp. 45-61

Philippou, A. and Sokolsky O. (2008), 'Process-Algebraic Analysis of Timing and Schedulability Properties', Handbook of Real-Time and Embedded Systems

Rajendran, C. and Holthaus, O., 'A comparative study of dispatching rules in dynamic flow shops and job shops', European Journal of Operational Research, 116 (1), 156-170

Shukla, C.S., Chen, F.F., 'The state of the art in intelligent real-time FMS control: a comprehensive survey', Journal of Intelligent Manufacturing , 1996, vol. 7, pp. 441-455

Stoop, P.P.M., Weirs, V.C.S., The complexity of scheduling in practice, International Journal of Operations and Production Management, vol. 16(10), pp.37-53.

UPPAAL, www.uppaal.com

Viswanadham, N., Narahari, Y., Johnson, T.L. (1990). Deadlock prevention and deadlock avoidance in Flexible Manufacturing Systems using Petri Net models. IEEE Transactions on Robotics and Automation, vol. 6, no.6, 713-723.

Wang, C., Ghenniwa, H., Shen, W., 'Real time distributed shop floor scheduling using an agent-based service-oriented architecture', International Journal of Production Research, vol. 46(9), pp. 2433-2452

Wu, N., Zhou M.C., Li, Z.W. (2008). Resource-oriented Petri Net for deadlock avoidance in Flexible Assembly Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 38, no.1, 56-68.

Wysk, R.A., Yang, N.S., Joshi, S. (1991). Detection of Deadlocks in Flexible Manufacturing Cells. IEEE Transactions on Robotics and Automation, vol. 7, no.6, 853-859.

Zhang, W., Judd, R.P. (2007). Deadlock avoidance algorithm for flexible manufacturing systems by calculating effective free space of circuits. Int.J. Prod. Res., vol. 46, no.13, 3441-3457.

Zhou, M.,(1995) Petri Nets in Flexible and Agile Automation, Kluwer Academic Publishers

Zhou, M., Jeng, M.D., 'Modeling, Analysis, Simulation, Scheduling and Control of Semiconductor Manufacturing Systems: A Petri Net Approach', IEEE Transactions on Semiconductor Manufacturing , vol.11, no. 3, August 1998, pp.333-357.

Zhou, M., Venkatesh, K.: Modeling , Simulation and Control of Flexible Manufacturing Systems – A Petri Net Approach, World Scientific Publishing , 1999.

24

Adaptive Implementation of Discrete Event Control Systems based on Sequential Function Charts

Ramón Piedrafita and José Luis Villarroel
Department of Computer Science and Systems Engineering,
University of Zaragoza
Spain

1. Introduction

The discrete-event system (DES) is a class of dynamic systems whose behaviour is governed by *discrete events* and they state occupy a discrete symbolic-valued state at each time instant. These discrete events occur asynchronously and instantaneously at discrete instants of time and lead to a change of the state. Between event occurrences, the state of DES is unaffected. The DES behaviour is described by the sequence of events that occur and the sequence of states. Examples of DES abound in the industrial world as automated manufacturing systems, monitoring and control systems, supervisory systems; in building automation; in control of aircraft systems, railway systems…(Cassandras 1993).
An example of a discrete event system is the classic programmable logic controller (PLC) controlling a sequential machine. The PLC acts as a discrete event control system (DECS). The DECS acts through the outputs over the actuators of the machines, and receives information of the state of the machines or events that happen in them through sensors. In the design of a DECS is neccesary to specify its dynamic behaviour, that is, the form of generating its outputs in response to the inputs. This specification can be carried out in different forms and will be a model of the desired behaviour of the system. There may be various desired behaviours for the same machine if the actions to be performed are different. The specification for the desired behaviour can be performed using the formalism of Petri nets. The technology translation can be done in a PLC in Sequential Function Chart language (SFC).
Programmable Logic Controllers are extensively used in the control of production systems and their use is, at the present, widespread in most industrial sectors. The combination of the PLCs intelligence with the development of sensors and actuators, ever more specialized, allows a greater number of processes to be automated. These devices offer a series of advantages that meet some of the most important manufacturing industry requirements in recent years, such as low cost, capacity to control complex systems, flexibility (they can be quickly and easily re-programmed), reduced downtime and easier programming, and reliable and robust components ensuring their operation for a long time.

The reaction time of a PLC is a fundamental matter in discrete event control systems. The PLC reads the inputs, executes the SFC and writes the output in a cyclic or periodic manner. In this chapter, we are interested in the execution time of algorithms that make the SFC of a control application evolve. We will show that the reaction time of a PLC depends greatly on the SFC structure, on the events sequence and also on the algorithm that executes the SFC. With the objective of minimizing the reaction time, we decided to design a Supervisor controller, which we have called the Execution Time Controller (ETC). The aim of the ETC is to determine in real time which algorithm executes the SFC the fastest and to change the execution algorithm when necessary.
We propose to adapt the classical implementation techniques of Petri nets to execute SFCs. Thus, we have developed execution algorithms derived, on the one hand, from the Deferred Transit and the Immediate Transit SFC evolution models and, on the other hand, from Petri net implementation techniques (Brute Force, Enabled Transitions and Representing Places).
The organization of this chapter is as follows. Section 2 is devoted to Discrete Event Systems, and Section 3 to Sequential Function Charts. Section 4 shows several implementation techniques of the SFC whose execution time is analyzed in Section 5. In Section 6 we present the Execution Time Controller. In Section 7 the technique is evaluated. The section describes the tests run to evaluate the estimation techniques and the working of the ETC in real time. Finally, in Section 8, we present the main conclusions.

2. Discrete Event Control Systems

An example of a discrete events system is the classic PLC controlling a sequential machine. The PLC acts as a discrete event control system (DECS) (see Fig. 1). The DECS acts on the machines by sending output signals to the actuators and receives information about the state of the machines or events occurring in them through sensors. The DECS receives input signals not only from the machine sensors, but also from the commands of the control panel, from supervision systems and even from other DECS. An output signal can be a signal sent to an actuator to act on a physical process, to increase a variable or to send a message.
The main function of discrete event control systems is to govern the workings of a machine in such a way that the desired behaviour is achieved. This is based on the coordination between the information received and the actions ordered to be carried out. A machine carries out the action ordered by the control system until the system decides that the action has been completed at which point it orders the machine to cease the action. In order that the control system can decide to end the action, it needs to obtain information indicating that the action should finish. This information can come from the sensors placed in the machine. With this information, the control system knows that it must execute an evolution. It has to pass from the state in which it performs the action to the subsequent state which could be one of many (perform another action, await material, etc.).
An approach to the design of a DECS involves specifying its dynamic behaviour, in other words the way it generates its outputs in response to the inputs. This specification can be carried out in various ways and will be a model of the desired functioning of the system. The same machine may have different ways of functioning if the actions to be performed are different. The specification of the desired behaviour can be carried out using formalisms such as Petri nets. The technology translation can be done in a PLC using the Sequential Function Chart language (see Fig. 2).

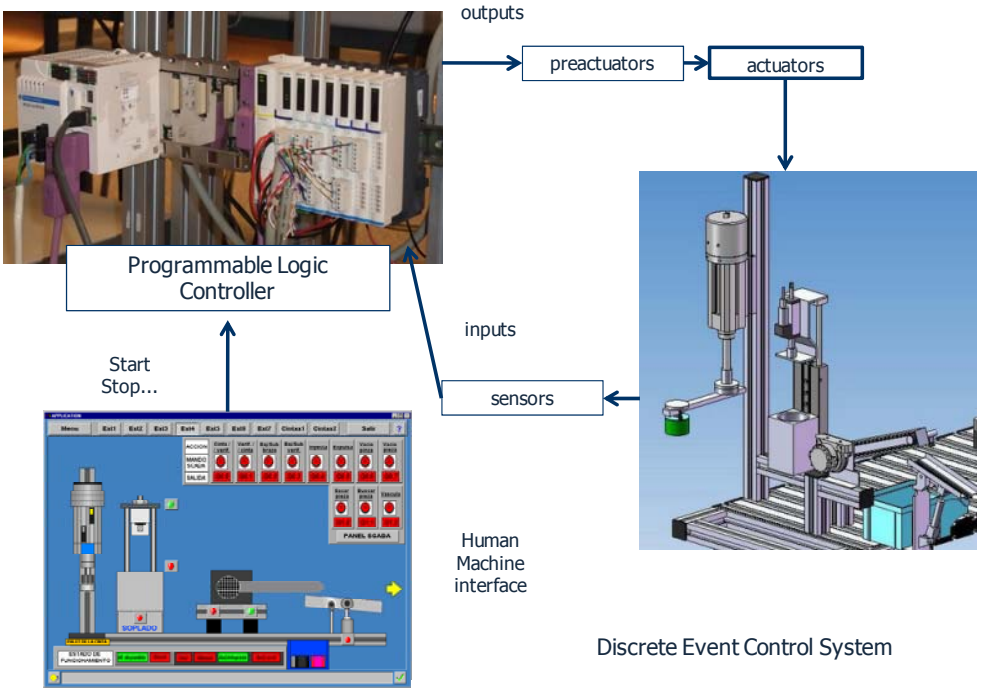


Fig. 1. Discrete Event Control System

3. Sequential Function Charts

In 1975, one of the working groups of the now defunct AFCET (Asociation Francaise pour la Cibernétique Economique et Technique), the Logic Systems group, decided to establish a commission for the standardization of the representation of logic controller specifications. In August 1977 a commission comprising 12 academics and researchers and 12 representatives of companies such as EDF, CEA, Merlin-Gerín, and Telemecanique signed the final report.
In brief, the group was looking for a model for the representation and specification of the functioning of systems controlled by logic controllers, through automatisms. The specification model only describes the desired behaviour, without detailing the technology with which the real implementation is effected. The model was named Grafcet (David 1995) and is recognised by standard IEC-848 (IEC 1988).
Similar to Grafcet, the Sequential Function Chart (SFC) are standardized in IEC 61131 (ISO/IEC 2001) where is defined as one of the main PLC programming languages. A SFC program is organized into a set of steps and transitions connected by direct links. Associated with each step is a set of actions, and with each transition a transition predicate.

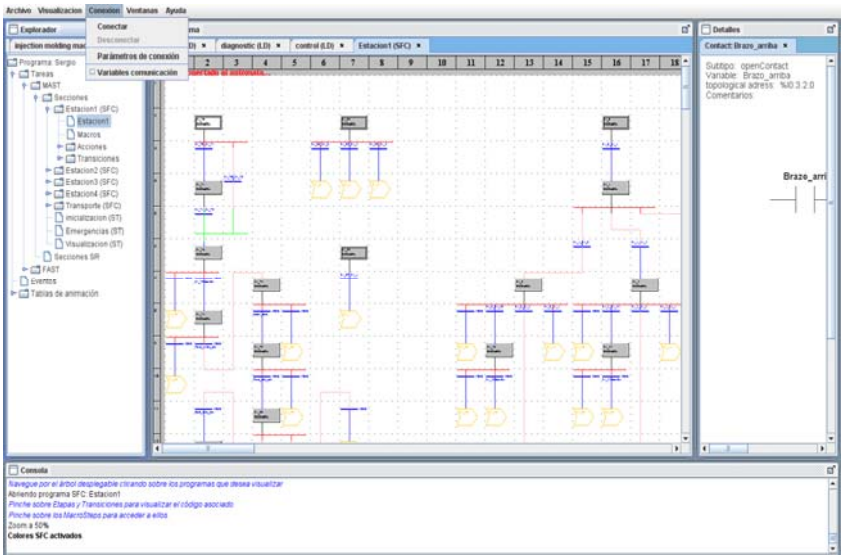


Fig. 2. PLC programming in Sequential Function Chart

The SFCs are binary Petri nets with an interpretation for the control of industrial systems (Silva 1985)

- Immediate actions are associated with the deactivation and activation of the steps (e.g., control signal changes, code execution).
- Level control signals are associated with active steps.
- Predicates are associated with transitions, as are additional preconditions for the firing of enabled transitions. Predicates are functions of system inputs or internal variables.

We take as an example the SFC shown in Fig. 3. The initial step (*Automatic_star*) is drawn with a double rectangle. The two output transitions of the initial step (*move_piece* and *NOT move_piece*) are in conflict. The default priority rule for solving a conflict is a left to right precedence. The standard does not require a priority relation between transitions or that the transitions predicates are in mutual exclusion.

When all the input steps of a transition are active and the transition predicate or condition is true, the transition is fired, the input steps are deactivated and the output steps are activated. In the example of the Fig. 3: if the step named *handgotoup* is active and the transition *hand_up* is true, the step *handgotoup* is deactivated and the step named *handgotopiece* is activated.

Actions can be programmed in a step. The type of programmed action is defined by the action qualifier. For example, a type N action is executed in all the cycles in which the step is active. The S, SD, SL, and SD actions are activated when the step in which they are programmed is activated, stored in an action buffer and from this point on are independent of the state of the step. They can only be deactivated by a type R action. Time limited actions can be programmed with type L or D qualifiers. There are also impulse type actions such as type P that are executed only when the step is activated.

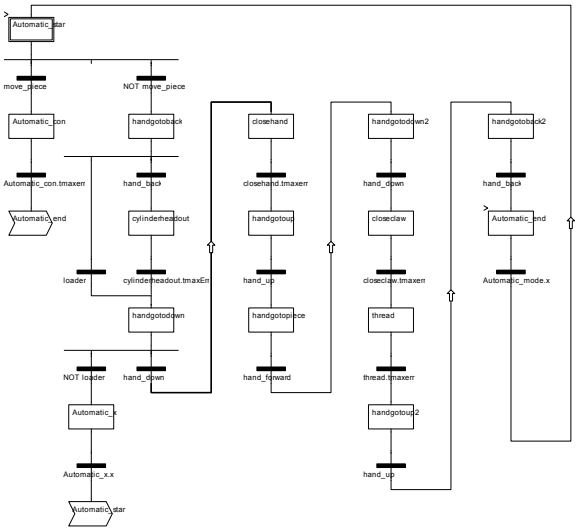


Fig. 3. Sequential Function Chart example

Table 1 shows the actions that can be programmed in a SCF. In a PLC cycle, the following must be executed:

- Actions which depend on the state of a step: action qualifiers N, L, D, P, P0, P1.
- The step in which is programmed the storage of the stored actions (S, SL, SD, DS) and their cancellation (R).
- the stored actions (S, SL, SD, DS)

The action types and qualifiers are the standard ones of the IEC 61131 (ISO/IEC 2001).

Qualifier	Description
N	Non-stored, executes while step is active.
L	Limited, executes only a limited time while step is active.
D	Delayed, starts executing after the step has been active.
S	Stored, starts executing when the step is activated until reset.
R	Reset stored action.
SL	Stored and limited
SD	Stored and delayed
DS	Delayed and stored
P	Pulse, executes when the step is activated.
P1	Pulse, positive flank, executes once when the step is activated.
P0	Pulse, negative flank, executes once when the step is deactivated.

Table 1. SFC actions.

4. Implementation of Sequential Function Charts

In the last 25 years, researchers have devoted considerable attention to the software implementation of Petri Nets (PN); see for example (Colom, Silva et al. 1986) (Briz and Colom 1994) (Taubner 1988) (Bruno & Marchetto 1986) (Garcia & Villarroel 1999) (Piedrafita & Villarroel 2006a). A PN implementation can be hardware or software. However, we are interested in the second approach, the software implementation. A software implementation is a program which fires the PN transitions, observing marking evolution rules, i.e., it plays the "token game". An implementation is composed of a control part and an operational part. The control part corresponds to the structure, marking and evolution rules of the PN. On the other hand, the operational part is the set of actions and/or codes of the application, associated with the PN elements.

According to different criteria, a PN implementation can be mainly classified as compiled or interpreted, as sequential or concurrent and as centralized or decentralized.

An implementation is interpreted if the SFC PN and the marking are codified as data structures. These data structures are used by one or more tasks called interpreters to make the PN evolve. The interpreters do not depend on the implemented PN. A compiled implementation is based on the generation of one or more tasks whose control flow corresponds to PN evolutions.

A sequential implementation is composed of only one task, even in PN with concurrency. This kind of implementation is common in applications whose operational part is composed by impulse actions without significant execution time. A concurrent implementation is composed of a set of tasks whose number is equal to or greater than the actual concurrency of the PN. Examples of concurrent implementations can be seen in (Colom, Silva et al. 1986) or in (Taubner 1988).

In a centralized implementation the full control part is executed by just one task, commonly called the token player or coordinator. The operational part of the implementation can be distributed in a set of tasks to guarantee the concurrence expressed by the PN (see for example (Colom, Silva et al. 1986)).

The problem of implementing a SFC is very similar to implementing a PN. Currently most industrial PLCs run their programs in an interpreted and centralized manner. The PLC reads the inputs, runs the SFC interpreter (also called coordinator in this paper) and writes the outputs. In the execution of the SFC it is necessary to determine which transitions can fire, and fire them making the state of the SFC evolve. It will also make the actions programmed in the steps.

The algorithm to determine which transitions are enabled and can fire is important because it introduces some overhead in the controller execution and the reaction time is affected. In the present work we have implemented and study several algorithms in which different enabled transition search techniques are developed:

- Brute Force (BF). PN implementation technique.
- Deferred transit evolution model (DTEVM). SFC implementation technique.
- Immediate transit evolution model (ITEVM). SFC implementation technique.
- Static Representing Places (SRP). PN implementation technique.
- Enabled Transitions (ET). PN implementation technique.

With the objective of carrying out a comparative study, all of these techniques have been uniformly implemented.

In the Brute force algorithm all the transitions are tested for firing. Brute Force algorithms do not try to improve the search of enabled transitions. Works such as (Peng & Zhou 2004) (Uzam & Jones 1996) (Klein, Frey et al. 2003) belong to this implementation class.
The IEC-61131 standard is not very precise in the definition of the SFC execution rules. Different execution models have been proposed to interpret the standard. As with BF, in the Immediate Transit Evolution Model (ITEVM) algorithm all the SFC transitions are tested for firing (Hellgren, Fabian et al. 2005). However, the Deferred Transit Evolution Model (DTEVM) (Hellgren, Fabian et al. 2005) only performs the testing of the transitions descending from the active steps, improving in this way the Brute Force operation.
In (Lewis 1998) the IEC-61131 standard is interpreted and the following tasks are proposed to run an SFC:

1. Determine the set of active steps
2. Evaluate all transitions associated with the active steps
3. Execute actions with falling edge action flag one last time
4. Execute active actions
5. Deactivate active steps that precede transition conditions that are true and activate the corresponding succeeding steps
6. Update the activity conditions of the actions
7. Return to step 1

These tasks are implemented in the DTEVM algorithm. In DTEVM, the transition conditions of all transitions leading from active steps (marked places in Petri net terminology) are evaluated first. Then, the transitions that were found to be fireable are executed one by one. In ITEVM, the transition conditions of all transitions of SFC are evaluated one by one. In the case of a transition condition being true, i.e., the corresponding transition is fireable, this transition is fired immediately.
In the Static Representing Places (SRP) algorithm, only the output transitions of some representative marked steps are tested (Colom, Silva et al. 1986). Each transition is represented by one of its input steps, the Representing Place. The remaining input steps are called synchronization steps. Only transitions whose Representing step is marked are considered as candidates for firing.
In the Enabled Transitions algorithm, only totally enabled transitions are tested. A characterization of the enabling of transitions, other than marking, is supplied, and only fully enabled transitions are considered. This kind of technique is studied in works such as (Silva & Velilla 1982) and (Briz. 1995).

4.1 Algorithm execution cycle

All implementation techniques are based on a treatment cycle which processes steps or transitions commonly stored in lists. The implementation of treatment the cycle is based on two kinds of lists that make an SFC evolve: treatment lists to be processed in the present treatment cycle and formation lists to be processed in the next cycle. The fundamental difference between each of the implementation techniques lies in the way in which the formation lists are built, and hence in the transitions which are considered in each treatment cycle.
One of the most expensive operations in execution time is the search and insertion in lists. The time cost of such operations depends directly on the size of the lists. Therefore, it is stated in the algorithms where it carries out such operations.

The basic treatment cycle of a SFC interpreter consists of three phases: (1) Enabling Test, (2) Transition firings (with two sub-phases: start and end), and (3) Lists update.
The Enabling Test phase verifies the enabling of the transitions belonging to the treatment list. A transition is enabled if all of the input steps are active. An enabled transition will be fired in the next phase if the associated condition is true.
All algorithms present two separate phases in the firing of transitions:

1- Start of transitions firing: deactivation of input steps of each fired transition.
2- End of transitions firing: activation of output steps of fired transitions.

The TransitionsFired list links both phases. In this way, the SFCs are executed step by step and avalanche effects are avoided. At the end of firing, the formation list is built with places or transitions being candidates for treatment in the next cycle.
Finally, at the end of the cycle, the elements of the formation list are analyzed and can become part of the treatment list for the next cycle.
In the following paragraphs we show the ET (Silva & Velilla 1982) (Briz. 1995), SRP (Colom, Silva et al. 1986) and the DTEVM (Hellgren, Fabian et al. 2005) algorithms in more detail to illustrate how all the techniques have been coded. The ITEVM algorithm can be consulted in (Hellgren, Fabian et al. 2005). The procedures for the execution of the actions programmed in the SFC have been included, with the update of the activity conditions of the actions (ISO/IEC 2001).

4.2 Enabled Transitions Technique

Program 1 presents the basic treatment cycle of the coordinator for the ET technique. This treatment cycle is also illustrated in Fig. 4. The following data structures will be available (see Fig. 4):

- Enabled Transitions List (ETL). Treatment list made up of the transitions with all active input steps.
- Almost Enabled Transitions List (AETL). Formation list which is built with the output transitions of the steps activated in the firing of the transitions, i.e., the transitions that can become enabled in the next cycle.

```
loop forever
  Executeactionswithfallingedge();
  Executeactiveactions();
// enabling test
  while elements in ETL do
    T = next_element (ETL);
    if enabled (T) and predicate(T) then
// start of transitions firing
      Demark_input_steps(T, ETL);
             // ETL updating
      Add(Transitionsfired, T);
    end if ;
  end while ;
// end of transitions firing
  while elements in Transitionsfired do
    T = next_element (Transitionsfired);
```

```
    Mark_output_steps(T, AETL);
      // update AETL
  end while ;
  Clear(Transitionsfired);
//list update
  Update(ETL, AETL); //operations of search and insertions in list
  Clear(AETL);
  Updateactivityconditions();
end loop ;
```

Program 1. ET Coordinator Treatment Loop

For each transition of the SFC a data structure is necessary that stores:

- List of input steps
- List of output steps

At the start of transitions firing Demark_input_steps (T, ETL) encapsulates the deactivation of the input steps of the transition fired, and the update of the ETL list. In this technique, the ETL (the treatment list) contains all transitions enabled at the beginning of the cycle. From this list each fired transition must be extracted and also the disabled transitions belonging to effective conflicts.

In the function Update(ETL, AETL) the treatment list is prepared for the next cycle. The transitions in AETL are verified for enabling and, if positively verified, are added to the ETL (if they do not already belong). At this point, the algorithm performs search and insertion in list operations.

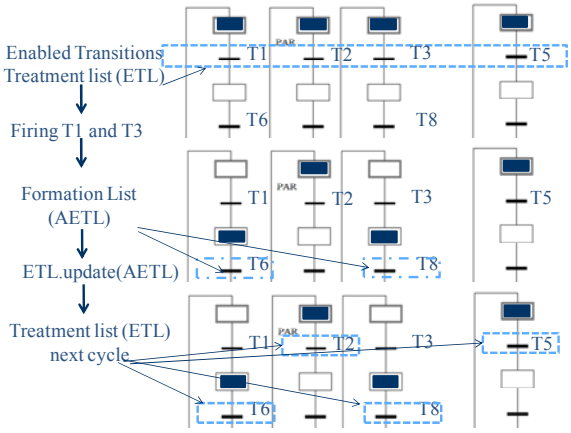


Fig. 4. Treatment List and Formation List of the Enabled Transitions Technique

4.3 Static Representing Places Technique

Program 2 presents the basic treatment cycle of the coordinator for the SRP technique. This treatment cycle is also illustrated in Fig. 5.

```
loop forever
  Executeactionswithfallingedge();
```

```
  Executeactiveactions();
  while elements in ARSL do
    Rstep = next_element (ARSL);
    Transitionsrepr = Rstep.transitionsrep;
// enabling test
    while T in Transitionsrepr do
      if enabled (T) and predicate(T) then
// start of transitions firing
       Demark_input_steps (T, ARSL, ASSL); // ARSL and ASSL updating
        Add(Transitionsfired , T);
        Break ();
      end if;
    end while ;
  end while ;
// end transitions firing
  while T in Transitionsfired do
    Mark_output_steps(T, ARSLnext, ASSLnext);
              // ARSLnext and ASSLnext updating
              // involves search and insertion in list operations
  end while ;
  Clear(Transitionsfired);
//list update
  Update(ARSL, ARSLnext);
       // involves search and insertion in list operations
  Update(ASSL, ASSLnext);
       // involves search and insertion in list operations
  Clear(ARSLnext); Clear(ASSLnext);
  Updateactivityconditions();
end loop ;
```

Program 2. SRP Treatment Loop

The following data structures will be available (see Fig. 5):

- Active Representing Steps list (ARSL) and Active Synchronization Steps list (ASSL). Treatment lists containing the active Representing and Synchronization Steps.
- Active Representing Steps list (ARSLnext) and Active Synchronization Steps list (ASSLnext). Formation lists with the Steps that will be active in the next cycle by the firing of the transitions.

For each representing step a data structure is necessary that contains:

- List of transitions represented by the Step

In all the transitions of the SFC a data structure will be necessary that stores:

- Representing step
- List of synchronization steps
- List of transitions in conflict

- List of active representing steps after firing
- List of active synchronization steps after firing

In each cycle only the output transitions of an active representing step are verified for enabling. If a represented transition fires, the verification process for the representing step ends because the rest of the represented transitions become disabled (they are in effective conflict).

At the start of the transitions firing phase the function `Demark_input_steps (T, ARSL, ASSL)` encapsulates the deactivation of the input steps of the transition fired, and the updating of the ARSL and ASSL lists. The deactivated steps should be removed from the ARSL (if it is the representing step of the transition) or from ASSL (if it is a synchronization step of the transition). These fired transitions are added to the list `Transitionsfired`.

At the end of the transitions firing phase the function `Mark_output_steps (T, ARSLnext, ASSLnext)` encapsulates the activation of the output steps of the transition fired and the building of the lists ARSLnext and ASSLnext. The output steps of the transitions in the `Transitionsfired` list are activated. The activated steps should be added to the list ARSLnext (if it is the representing step) or to ASSLnext (if it is a synchronization step). At this point, the algorithm performs search and insertion in list operations.

At the end of the cycle, the ARSL list is updated in `Update (ARSL, ARSLnext)`. The ARSLnext elements are added to the ARSL (if they do not already belong). The ASSL list is also updated in `Update(ASSL, ASSLnext)`. The ASSLnext elements are added to the ASSL (if they do not already belong). At this point, the algorithm also performs search and insertion in list operations.

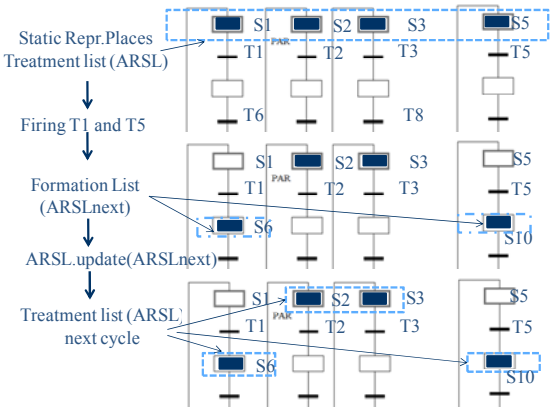


Fig. 5. Treatment List and Formation List of the Representing Places Technique

4.4 Deferred Transit Evolution Model Technique

Program 3 presents the basic treatment cycle of the coordinator for the DTEVM technique. The following data structures will be available (see Fig. 6):

- Active Steps list (ASL). Treatment lists containing all the active steps.
- Enabled Transitions List (ETL). Treatment lists containing the transitions with their input steps active and with their predicate condition true.

This treatment cycle is also illustrated in Fig. 6. The search of the Active Steps is carried out in DTEVM at the start of each cycle, in the function computeactivesteps. The execution time of this search function is proportional to the number of steps of the SFC.

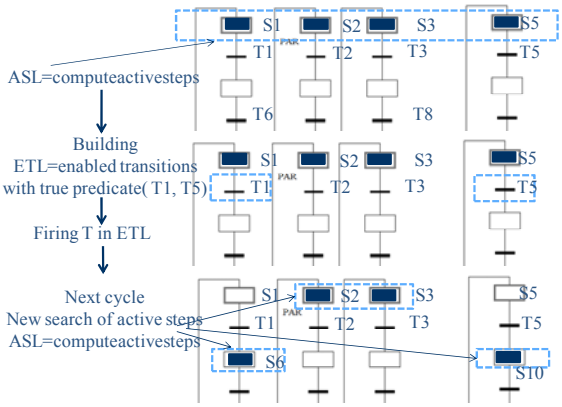


Fig. 6. Treatment List and Formation List of the DTEVM Technique

The enabling test of the transitions is carried out in two phases. First, it finds the enabled transitions with true predicates that are output of the steps in the ASL list, drawing up the ETL list. It then goes through this list and fires the transitions. The enabling must be re-evaluated to prevent the firing of transitions in conflict. This algorithm does not perform any search and insertion in list operations.

```
loop forever
  ASL=computeactivesteps();
// enabling test
  while elements in ASL do
    Activestep = next_element (ASL);
    Transoutput= Activestep.Transoutput;
    while T in Transoutput do
      if enabled (T) and predicate(T)then Add(ETL, T); end if;
    end while ;
  end while ;
  Executeactionswithfallingedge();Executeactiveactions();
// start transitions firing
  while T in ETL do
    if enabled(T) then
      Demark_input_steps(T);Add(Transitionsfired, T);
    end if;
  end while ;
// end transitions firing
  while T in Transitionsfired do
    Mark_output_steps(T);
```

```
  end while ;
  Clear(Transitionsfired);Updateactivityconditions();
end loop;
```

Program 3. DTEVM Treatment Loop

5. Estimation of the execution time of the algorithms.

An analysis of SFC implementation algorithms was carried out in (Piedrafita & Villarroel 2008 a). Brute Force (BF), Enabled Transitions (ET), Static Representing Places (SRP) Inmediate Transit Evolution Model (ITEVM) and Deferred Transit Evolution Model (DTEVM) were analyzed. The main ideas obtained in (Piedrafita & Villarroel 2008 a) are:

- The implementation of the Enabled Transitions and Static Representing Places algorithms can lead to enormous savings in execution time compared to the Brute Force algorithm.
- The choice of the most suitable type of algorithm to execute a SFC depends on the SFC behavior (effective concurrency vs. effective conflicts).

The presented tests show that the relative performance of implementation algorithms depends on the model concurrency structure but also on the dynamics imposed by the controlled system. In most of the cases, the SRP and the ET algorithms coming from PN field have good behaviors. The PN implementation techniques provide an improvement in the development of industrial controllers based on SFC language.

The execution of SFCs without a suitable algorithm can suppose an increasing of the computing time, and a worse and slower answer in control applications. It is very difficult to estimate what algorithm will run faster an SFC. In real-time control only one algorithm can run the SFC, thus it must be possible to estimate what would be the execution time of the other alternative non executed algorithms

The execution time, given its ease of measuring, is the physical parameter that most easily allows the performance of an algorithm to be evaluated. However, the execution time must be considered as an explicit measure of the performance of an algorithm, where it directly reflects the influence of the other parameters.

The execution time of the algorithms described in the previous section will depend on the number of transitions tested for enabling in each cycle, and on the number of search and insertion in list operations. The computation time of the test for enabling operations does not depend on the size of the SFC. However, the computation time of the search and insertion in list operations does depend directly on the size of the algorithm lists.

The number of transitions tested for enabling in the ET technique is the sum of the sizes of ETL and AETL. For the SRP technique, the number of transitions tested for enabling start from a minimum, being the number of Active Representing Steps (if firing even the first transition represented) to a maximum, being the total of the transitions represented by the Active Representing Steps (if firing even the last transition represented or if there is no firing transition). For the DTEVM technique, the number of transitions tested start from a minimum, the total of the output transitions of the Active Steps, to a maximum, twice the total of the output transitions of the Active Steps (if all predicates are true).

One of the most expensive operations in execution time is the *search and insertion in lists*. The presented techniques frequently use this type of operation, especially in the real time building of formation lists and in the final phase of updating lists. The execution time of

such operations depends directly on the size of the lists. There are techniques that abound in the use of *search and insertion list* operations, such as Representing Places. In other techniques such as Brute Force this type of operation is not performed since the lists are not updated.

The search and insertion in list operations are performed in techniques such as ET or SRP because of the managing in real time of the treatment and formation lists. In the algorithms such operations are performed at the end of the firing of transitions and in the final update of the lists. Hence, if no transitions fire, the number of such operations is null. In the ET technique, the number of this kind of operation is the number of transitions of AETL that are enabled and become part of ETL. In SRP, it is twice the number of Steps that become active in the transitions firing, because SRP manages four lists. The computation time of the search and insertion in list operations depends directly on the size of the lists.

The SFC implementation techniques are based on a cyclic treatment (see Program 1 to 3). The main loop goes through the treatment and formation lists using an algorithm that depends on the executed technique. The algorithm cycle execution time depends on the size of the treatment and formation lists. The size of the treatment lists in the case of ET and SRP depends on the current SFC state. This determines the number of enabled transitions and the number of active representing steps. The size of the formation lists depends on the number of transitions that fire in the cycle. Thus, the execution time depends on the evolution of the SFC state, the SFC structure and the sequence of events.

As algorithms use different lists, their execution times will be different. The estimation of the algorithm execution time is based on the measurement of the mean time taken by these loops and on the estimation in real time of the size of the treatment and formation lists.

First, we study the SRP algorithm. The cycle execution time (CET) can be estimated by the following expression:

$$\begin{aligned} CET(SRP) = {} & Tenabl * SIZE(ARSL) * TR_{TESTED} + Tfiring * FT_{NUMBER} + \\ & TinsertStep * (SIZE(ARSL_{NEXT})/2) * (SIZE(ARSL_{NEXT})) + TinsertStep \\ & * (SIZE(ASSL_{NEXT})/2) * SIZE(ASSL_{NEXT})) + TinsertStep * \\ & (SIZE(ARSL_{NEXT}) * (SIZE(ARSL)) + TinsertStep * (SIZE(ASSL_{NEXT})) * SIZE(ASSL)) \end{aligned} \quad (1)$$

Where FTnumber is the number of fired transitions; Trtested is the mean represented transitions tested of an active representing step; Tenabl is the time for the enabling test operation of one transition; Tfiring is the mean time for firing one transition; TinsertStep is the mean time necessary for the search and insertion in list operation of one Step in a List of size one (performed in the final phase of updating list).

The ET algorithm is also analyzed. The cycle execution time can be estimated by the following expression:

$$\begin{aligned} CET(ET) = {} & Tenabl * (SIZE(ETL) + SIZE(AETL)) + Tfiring * FT_{NUMBER} + \\ & Tinserttran * SIZE(AETL) * SIZE(ETL) \end{aligned} \quad (2)$$

Where TinsertStep is the mean time necessary for the search and insertion in list operation of one transition in a List of size one (performed in the final phase of updating list).

Establishing expressions for other implementation techniques is not complicated. Let us consider, for example, the brute force technique. The cycle execution time expression of the BF algorithm is:

$$CET(BF) = Tenabl * size(TL) + Tfiring * FTnumber \quad (3)$$

TL is the list with all transitions of the SFC.

6. The SFC Execution Time Controller

With the objective of minimizing SFC execution time, we decided to design a Supervisor controller which we have called the Execution Time Controller (ETC). The first version of the ETC is presented in (Piedrafita & Villarroel 2008 b).
The main function of the ETC is to determine in real time which algorithm executes a SFC fastest. The ETC executes the algorithm chosen and estimates the execution time of the other non-executed algorithms, choosing the best algorithm in line with the controlled system. If necessary, the ETC changes the algorithm. In the next section we present in detail how the execution time (ExT) of the running and the alternative algorithms are estimated. To avoid the overload of continuous algorithm changes, an integral cost function is used:

$$\varepsilon = ExT_{calculated}(running\_alg) - ExT_{estimated}(alternative\_alg) \quad (4)$$

$$I(k) = \begin{cases} I(k-1) + \varepsilon(k) & if \quad I(k-1) + \varepsilon(k) > 0 \\ 0 & if \quad I(k-1) + \varepsilon(k) \le 0 \end{cases}$$

The change is made when I(k) is greater than half of the execution time of the executed algorithm. When a change happens, I (k-1) = 0.

```
// Offline Control
Load SFC
Measuring Times
First Choice of the best algorithm
Return to initial steps
// Online Control
loop forever
  Read Inputs
  SFC execution with the best algorithm
  Write Outputs
  Compute execution time of running_alg
  Estimate execution time of alternate_alg
  Compute I(k)
  If I(k)>(ExT_calculated(running_alg)/2) then
    Change algorithm
    Initialize data structures
    I(k-1)=0
  End if
  Wait for next period();
end loop
```

Program 4. Execution Time Controller

6.1 Times measuring.

The Tenabl, Tfiring, TinsertStep and Tinserttran times are measured in an offline execution test. For this purpose, the required measurement instrumentation is incorporated into the program. This instrumentation comprises the instructions required for reading the real time system clock and the necessary time difference calculations.

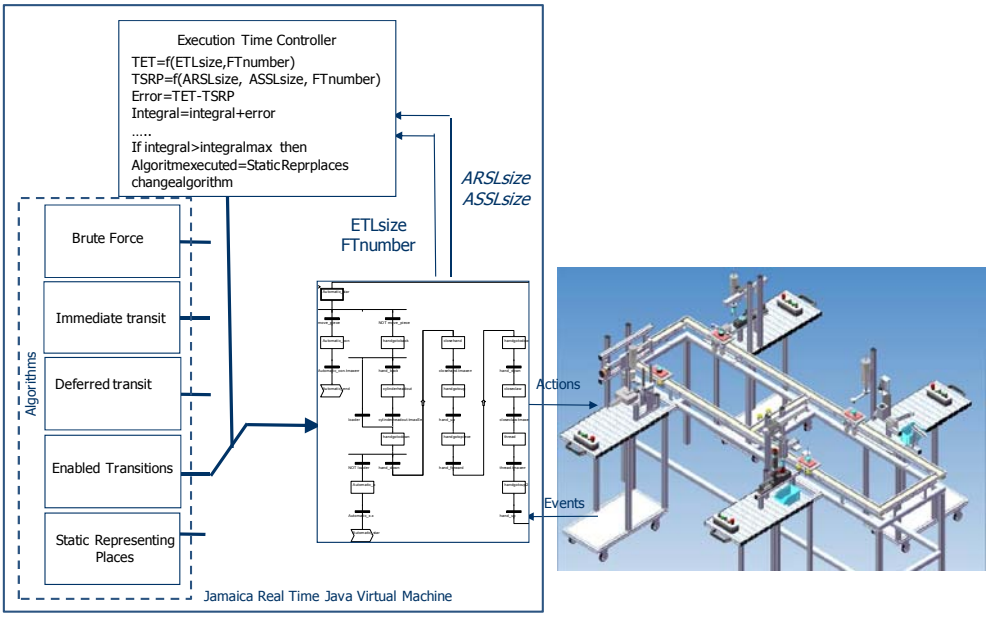


Fig. 7. Execution Time Controller

The Time measuring test consists of the execution of the SFC with one of the algorithms carrying out the firing of 2000 transitions without executing the programmed actions. The condition associated with the transitions is considered true so that the firing is immediate. If there are conflicts, the transition that fires is chosen at random.

For example, if the execution is done with the SRP algorithm during the test, the measurement of the Tenabl, Tfiring and TinsertStep times will proceed. The test is then repeated with the ET algorithm, and the Tinserttran time measurement is carried out.

The cycle times of each algorithm are also measured in these tests and the algorithm with the shortest cycle time is chosen for the first execution of the SFC with control actions.

6.2 Estimation of Data structure size.

In the real time execution of the ETC (see Program 4), the execution time of the executed algorithm can be measured by reading the system clock. To avoid an overload of the control actions, the execution time of the executed algorithm (runnig_alg) is then calculated with equations (1) to (3) (this depends on the algorithm being executed). In this case FTNUMBER and the sizes of the lists (ASRL, ASRLNEXT, …) are known by the ETC.

The ETC should obtain enough information to determine what would be the computation time of the other algorithms not executed at that moment. There is no problem with the algorithm being executed. For the other algorithms, should be obtained the size in real time of the treatment and formation lists if they were being executed. From this data it can be estimated what would be their computation time. This estimation should be carried out in real time and with an small overload in the execution time of the algorithm.
The execution times of the alternative algorithms should also be estimated with equations (1) to (3). The number of transitions fired is known given that it is the same as for the algorithm that is being executed. The value of the times measured in the test is also known. However, the size of the other lists must be estimated.
For example, in the execution of the ET algorithm, the size of the lists of the SRP algorithm must be estimated. The mean number of active representing steps and active synchronization steps is more or less constant in most SFCs; therefore, the size (ARSL) and the size (ASSL) will be the mean value estimated in the offline time measurement test.
Consequently, it can be stated that, on average, the firing of a transition involves the unmarking of its representing Step and the marking of a new one. The size (ARSL$_{NEXT}$) can be approximated by the number of transitions fired.

$$\text{size}\,(\text{ARSL}_{NEXT}) \approx \text{FT}_{NUMBER} \tag{5}$$

The size(ASSL$_{NEXT}$) can be approximated by the expression:

$$\text{size}(\text{ASSL}_{NEXT}) \approx \text{FT}_{NUMBER} * (f_{p}\_1) \tag{6}$$

fp is the average parallelism factor (number of output places of a transition) of the fired transitions.

$$\begin{aligned}\text{CET(SRP)} = {} & \text{Tenabl} * \text{size(ARSL)} * \text{TRtested} + \text{Tfiring} * \text{FT}_{NUMBER} + \text{TinsertStep} * \\ & (\text{FTnumber}/2) * \text{FT}_{NUMBER} + \text{TinsertStep} * \text{FT}_{NUMBER} * (\text{fp} - \\ & 1)/2) * \text{size(ASSLnext)}) + \text{TinsertStep} * \text{FT}_{NUMBER} * (\text{size(ARSL)}) + \text{TinsertStep} * \\ & \text{FT}_{NUMBER} * (\text{fp} - 1) * \text{size(ASSL)}) = \\ & \text{F1}\,(\text{size(ARSL)}, \text{size(ASSL)}, \text{FT}_{NUMBER})\end{aligned} \tag{7}$$

To estimate the size of the lists of the SRP algorithm when the algorithm executed is FB, the same technique is used given that in real time it is only necessary to know the number of transitions fired and the mean parallelism factor of these transitions.
If the SFC is executed with the SRP algorithm, it should be estimated what would be the computation time of the execution of the SFC with the ET algorithm. Therefore the sizes of the ETL and AETL lists should be estimated. The FT$_{NUMBER}$ is known because the two algorithms make the SFC evolve in the same way and therefore the number of fired transitions will be the same for both.
The size of the ETL list is estimated in the SRP execution when the sensibilization of the transitions represented by the active representing steps is tested. When the SRP algorithm finds an enabled transition it fires it and continues with the next active representing step. If, therefore, it is necessary to know how many transitions there are enabled among those represented by the representing place, two possible solutions are adopted:

- A first option is that the algorithm runs over all the transitions represented by a marked representing place, estimating their enabling.

- When the SRP algorithm finds an enabled transition it fires it, and the rest of represented transitions are not verified for enabling. An approximation is carried out considering enabled half of the rest of transitions.

The second solution was chosen for the tests given that the computation time is shorter. The size of the AETL list is estimated at the firing of the transitions, when the output steps are activated, as is the size of the set of output transitions of the output steps of the fired transitions.

$$\text{SIZE(AETL)} = \text{FTNUMBER} * f_p * f_d \tag{8}$$

fp is the parallelism factor (number of output steps of a transition) of the transitions fired and fd is the descendants factor (average number of output transitions of a step) of the steps activated in the transitions firing.

$$\begin{aligned} \text{CET(ET)} &= \text{Tenabl} * (\text{size(ETL)} + \text{size (AETL)}) + \text{Tfiring} * \text{FTNUMBER} + \\ \text{Tinserttran} * \text{size(AETL)} &* \text{size(ETL)} = \text{Tenabl} * (\text{size(ETL)} + \text{FTNUMBER} * f_p * f_d + \\ \text{Tfiring} * \text{FTNUMBER} &+ \text{Tinserttran} * \text{FTNUMBER} * f_p * f_d * \text{size(ETL)} = \\ & F_2(\text{size(ETL)}, \text{FTNUMBER}) \end{aligned} \tag{9}$$

A different technique is used to estimate the size of the ET algorithm lists when the FB algorithm is executed. Because with FB all the transitions in the SFC are covered in the enabling test, the size of the ETL list can be accurately known. In the first version of the ETC (Piedrafita & Villarroel 2008 b) , it was necessary to measure the size of the treatment and formation lists. In this second version of the ETC is only necessary to measure the size of the treatment lists, since the size of the formation lists is calculated from the number of transitions fired FTNUMBER.

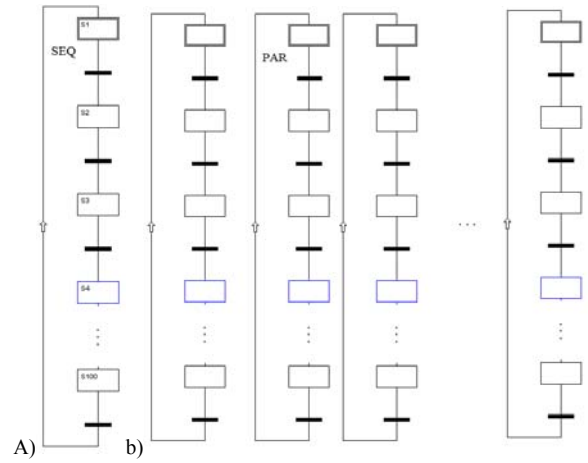


Fig. 8. SFCs Library

7. Technique Evaluation

7.1 Execution Platform

We have implemented the techniques in the Java language using the Java Real-time Specification (Bollella & Gosling 2000.) and following some ideas presented in (Piedrafita & Villarroel 2006a), (Piedrafita & Villarroel 2006b) and (Piedrafita & Villarroel 2007). In our implementations, we used the Real Time Java Virtual Machine JamaicaVM v2.7 (Aicas 2007). The target hardware was a personal computer with a Pentium IV processor at 1.7GHz, running Red Hat Linux 2.4. The Coordinator is implemented as a Periodic Real Time Thread of high Priority. The execution is made in a single processor and threads are scheduled following a static priorities policy without round-robin.

In the implementations developed here, the program loads the SFC structure from an XML file generated by an SFC editor. The implementation is independent of the SFC, and is therefore an interpreted implementation.

A library of Sequential Function Charts has been developed for carrying out the tests. The library is based on four base models which can be scaled using a parameter. These models represent most of the cases developed in industrial control: sequential systems and concurrent systems. The library comprises the following SFCs:

- SEQ. SFC with one sequential process composed of 1 to 100 steps (Fig. 8.a).
- PAR. SFC with p (1..100) sequential processes with 20 steps (Fig. 8.b).

7.2 Real Time Execution of ETC

The ETC controller has been tested with all the SFCs in the library and also with a real control application. This is a Flexible Manufacturing Cell in the Computer Science Department of the University of Zaragoza. The ETC obtains a high degree of success in all of the performed experiments, but here for the sake of brevity we present the results of the three most representative experiments. However, an exhaustive report of the experiments can be consulted in (Piedrafita 2008), accessible via the Internet.

The execution of the ETC takes place in the Real Time Java Virtual Machine Jamaica. This is implemented as Periodic Real Time Thread of high Priority with 20 ms of period. The execution is made in a single processor and threads are scheduled following a static priorities policy without round-robin.

The first experiment that we present is over a sequential SFC of 35 steps (see Fig. 9 left) and illustrates that the SRP algorithm is always the best in this experiment (Piedrafita & Villarroel 2007).

Fig. 9.a, shows the Real Time execution of The Execution Time Controller (ETC), the Real Time estimation of the same algorithm (SRP), and the Real Time estimation of one alternative algorithm(ET in this case). Fig. 9.b, shows also the Real Time execution of the algorithm SRP, and the Real Time execution of the algorithm ET.

The ETC chooses the SRP from the start since the estimation of the execution time of this algorithm is smaller than that of the ET algorithm. Because SRP is always better than ET, the integral cost function I(k) remains permanently null and therefore no algorithm change is performed.

The second experiment that we present is over a concurrent SFC compound of 10 sequential SFCs (see Fig. 9 c and d) and illustrates that the SRP algorithm is the best in this experiment (Piedrafita & Villarroel 2007) .

Fig. 9.c, shows the Real Time execution of The Execution Time Controller (ETC), the Real Time estimation of the same algorithm (SRP), and the Real Time estimation of one alternative algorithm(ET in this case) and the integral cost function I(k). Fig. 9.e, shows also the Real Time execution of the algorithm SRP and the Real Time execution of the algorithm ET.

As in the first experiment, the ETC chooses the SRP from the start since the estimation of the execution time of this algorithm is smaller than that of the ET algorithm. Because SRP is always better than ET, the integral cost function I(k) remains permanently null and no algorithm change is performed.

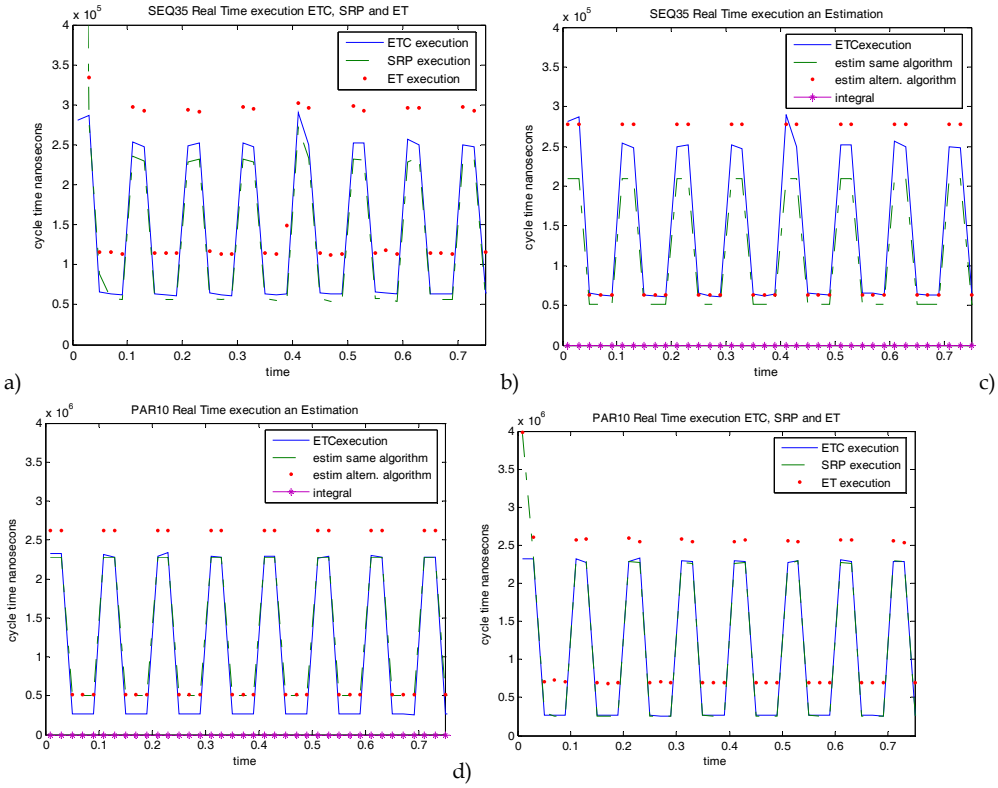


Fig. 9. Real Time Execution of the ETC with a sequential SFC of 35 states and a concurrent SFC compound of 10 sequential SFCs.

The third experiment that we present is over a concurrent SFC compound of 40 sequential SFCs (see Fig. 10 a and b) and illustrates that the SRP algorithm is the best algorithm in this experiment when not firing transitions, and ET is the best algorithm when all possible transitions are firing.

In the first 0.4 seconds all possible transitions fire and the best algorithm is ET, while in the next 0.4 seconds no transitions fire and the best algorithm is SRP. The event sequence is cyclically repeated.

At the beginning, the ETC executes the ET algorithm which, as we have seen, is the better algorithm in the first 0.4 seconds. However, at the instant 0.4 no events reach the SFC and SRP becomes better, therefore I(k) increases and the ETC changes to SRP at instant 0.48. At instant 0.8, the events reach the SFC and ET becomes better, therefore I(k) increases again and the ETC changes to ET at instant 0.9. For the whole evolution, ETC changes to SRP at instants 0.48, 1.28 and 2.08 and to ET at instants 0.9, 1.7 and 2.5. With the observed behaviour, the ETC achieves the minimum possible execution time of the SFC.
In industrial control it is very common that, in many cycles, events do not reach the SFC, and so no transition is fired, and when they are fired their quantity is variable. We can therefore differentiate between two operation regimes:

- Without events regime (static). No transitions are fired and the algorithm only runs the enabling test.
- With events regime (dynamic). Transitions are fired and the algorithm must run all the phases: enabling test, firing and updating of lists.

If desired, the ETC can choose the algorithm that has the shortest computation time in the enabling test (without events regime). The integral cost function is only calculated when no transitions are fired. The integral cost function is:

$$\varepsilon = ExT_{calculated}(running\_a\lg) - ExT_{estimated}(alternative\_a\lg)$$

If (FTnumber==0) {

$$\varepsilon = ExT_{calculated}(running\_a\lg) - ExT_{estimated}(alternative\_a\lg) \tag{10}$$

$$I(k) = \begin{cases} I(k-1) + \varepsilon(k) & if \quad I(k-1) + \varepsilon(k) > 0 \\ 0 & if \quad I(k-1) + \varepsilon(k) \le 0 \end{cases}$$

}

The execution of the ETC with this cost function can be seen in Fig. 10 c and d. It can be observed that the integral is only calculated when no transitions fire. At the instant 0.48 the ETC changes to the SRP algorithm, which has the shortest computation time in the without events regime.
If it is required that the ETC achieve the shortest reaction time for events, the integral cost function is only calculated when transitions fire. In this way the ETC chooses as the best algorithm that which has the shortest computation time in the firing of transitions. The integral cost function is:

$$\varepsilon = ExT_{calculated}(running\_a\lg) - ExT_{estimated}(alternative\_a\lg)$$

If (FTnumber >0) {

$$\varepsilon = ExT_{calculated}(running\_a\lg) - ExT_{estimated}(alternative\_a\lg) \tag{11}$$

$$I(k) = \begin{cases} I(k-1) + \varepsilon(k) & if \quad I(k-1) + \varepsilon(k) > 0 \\ 0 & if \quad I(k-1) + \varepsilon(k) \le 0 \end{cases}$$

}

The execution of the ETC with this function is shown in Fig. 10 e and f. It can be seen that the integral is only calculated when transitions are fired. At the instant 0.48 el ETC chooses the ET algorithm, which has the shortest computation time in the with events regime.

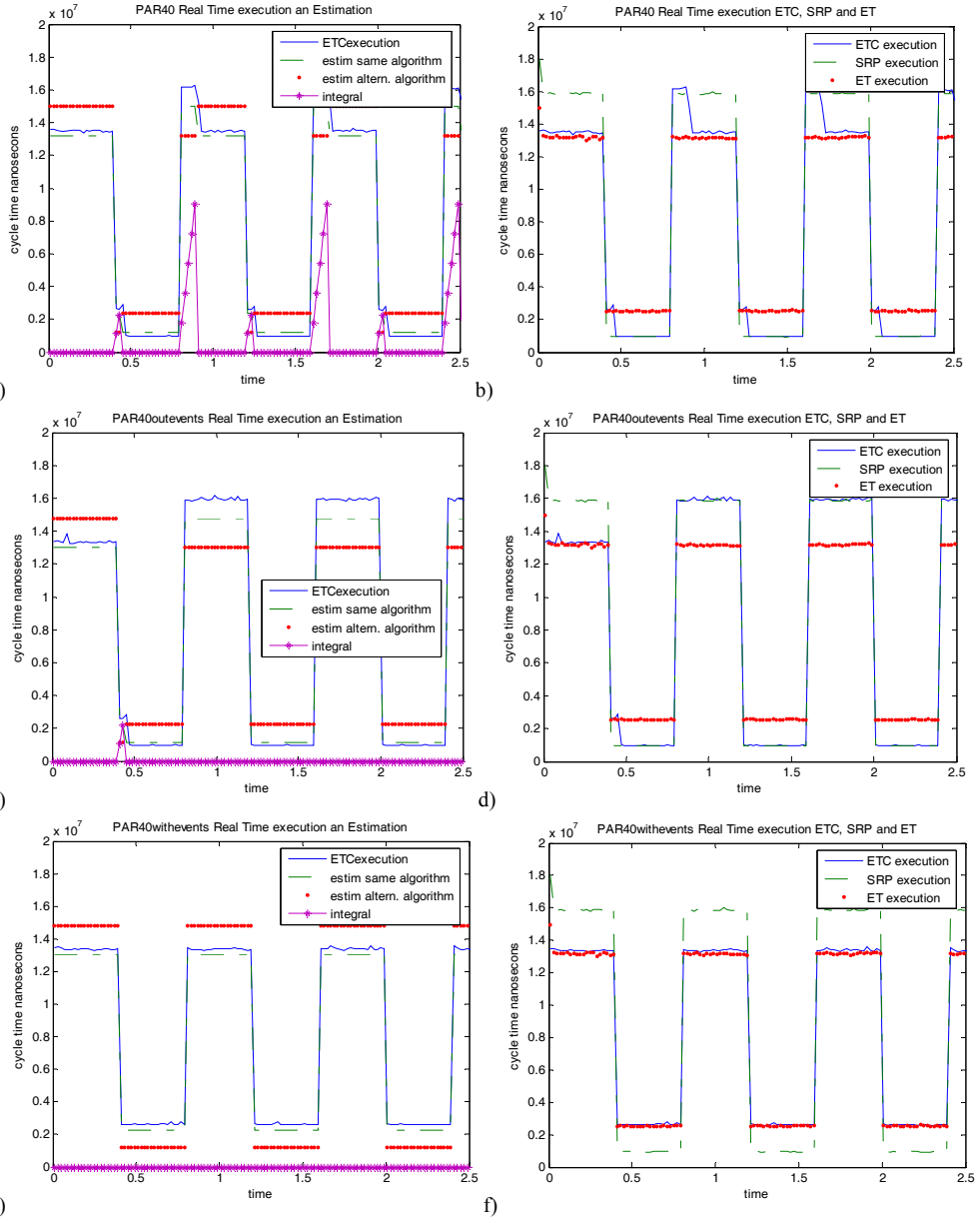


Fig. 10. Real Time Execution of the ETC with a concurrent SFC compound of 40 sequential SFCs.

8. Conclusions

In this work we have developed an adaptive implementation of Discrete Event Control Systems, the Execution Time Controller, which allows choosing in real time the most suitable algorithm to execute a Sequential Function Chart. The main function of the ETC will be to determine which algorithm executes a SFC the fastest. The proposed technique is analyzed with the two most important algorithms (from the point of view of performance): the enabled transitions and the static representing places. However, the ETC can work with any SFC implementation algorithm.

The ETC executes the chosen algorithm and estimates the execution time of other non-executed algorithms, deciding the best one in line. The execution of a SFC without a suitable algorithm can lead to a significant increase in the execution time, together with a less satisfactory and slower answer in control applications. The technique has been tested on a wide SFC library. Moreover, the ETC has also been tested in a real control application. The technique has a high success rate in the choice of the best implementation algorithm.

In the first version of the ETC, it was necessary to measure the size of the treatment and formation lists. In this second version of the ETC is only necessary to measure the size of the treatment lists, since the size of the formation lists is calculated from the number of transitions fired FTNUMBER

The ETC ensures that the control system to react in the shortest time possible, increasing quality control. In many cases this can also reduce the period of the task which implements the SFC interpreter. This implies an increase in quality control. This reduction in execution time may also be allocated to the execution of other tasks with heavy run-time such as the graphical interface or communications.

The ETC allows faster reaction times in SFC based control systems and also minimizes the power consumed by the controller.

9. References

Aicas, (2007). JamaicaVM Realtime Java Technology. http://www.aicas.com/jamaica.html.

Bollella, G. & J. Gosling (2000.). "The Real-time Specification for Java." Computer 33(6): 47-54.

Briz, J. L. & J. M. Colom (1994). "Implementation of Weighted Place/Transition Nets based on Linear Enabling Functions." Application and Theory of Petri Nets 815: 99–118.

Briz., J. L. (1995). "Técnicas de implementación de redes de Petri. ." PhD thesis, Univ. Zaragoza.

Bruno, G. & G. Marchetto (1986). "Process-translatable Petri nets for the rapid Prototyping of Process-Control Systems." Ieee Transactions on Software Engineering 12(2): 346-357.

Cassandras, C. G. (1993). Discrete event systems, Springer.

Colom, J. M., M. Silva, et al. (1986). "On software implementation of Petri nets and colored Petri nets using high-level concurrent languages." Seventh European Workshop on applications and theory of Petri nets, Oxford, July 86: 207-241.

David, R. (1995). "GRAFCET - A powerfull Tool for specification of logic controllers." Ieee Transactions on Control Systems Technology 3(3): 253-268.

Garcia, F. J. & J. L. Villarroel (1999). Translating time Petri net structures into Ada 95 statements. Reliable Software Technologies - Ada-Europe' 99. Berlin, Springer-Verlag Berlin. 1622: 158-169.

Hellgren, A., M. Fabian, et al. (2005). "On the execution of sequential function charts." Control Engineering Practice 13(10): 1283-1293.

IEC (1988). Preparation of Function Charts for Control Systems publication 848.

ISO/IEC (2001). "International standard IEC 61131-3 (2nd ed.). Programmable logic controllers—Part 3. ISO/IEC (final draft). ."

Klein, S., G. Frey, et al. (2003). PLC Programming with Signal Interpreted Petri Nets. ICATPN 2003, Eindhoven, Srpinger Verlag.

Lewis, R. W. (1998). "Programming industrial control systems using IEC 1131-3." IEEE control engineering series 50.

Peng, S. S. & M. C. Zhou (2004). "Ladder diagram and Petri-net-based discrete-event control design methods." Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34(4): 523 – 531.

Piedrafita, R. (2008). Experimentation with the Execution Time Controller (Research Report). http://automata.cps.unizar.es/realtime. Zaragoza, Departamento de Informática e Ingeniería de Sistemas.: 176.

Piedrafita, R. & J. L. Villarroel (2006 a). "Implementation of time petri nets in real-time Java." Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems: 178-187.

Piedrafita, R. & J. L. Villarroel (2006 b). Petri Nets and Java. Real-Time Control of a flexible manufacturing cell. Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on: 1246-1253.

Piedrafita, R. & J. L. Villarroel (2007). "Performance evaluation of petri nets execution algorithms." Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on: 1400-1407.

Piedrafita, R. & J. L. Villarroel (2008 a). Evaluation of Sequential Function Charts Execution Techniques. The Active Steps Algorithm. Emerging Technologies and Factory Automation. IEEE Conference on. Hamburg, Germany.

Piedrafita, R. & J. L. Villarroel (2008 b). Adaptive Petri Nets Implementation. The Execution Time Controller. 9th International Workshop on Discrete Event Systems 2008, Gothenburg, Sweden.Silva, M. (1985). Las Redes de Petri en la Automatica y la Informatica. Editorial AC, Madrid, 1985, Spanish.

Silva, M. and S. Velilla (1982). "Programmable logic controllers and Petri nets: A comparative study." IFAC/IFIP Symposium on Software for Computer Control, Madrid, Spain, October 1982: 83–88.

Taubner, D. (1988). "On the implementation of Petri nets." Lecture Notes in Computer Science 340: 418-439.

Uzam, M. and A. H. Jones (1996). Towards a Unified Methodology for Converting Coloured Petri Net Controllers into Ladder Logic Using TPLL: Part I - Methodology. International Workshop on Discrete Event Systems - WODES'96. Edinburgh, UK: 178 - 183.

Automated production monitoring and control system engineering by combining a standardized data format (CAEX) with standardized communication (OPC UA)

Miriam Schleipen
Fraunhofer IITB Germany

1. Introduction

Production monitoring and control systems are deployed to monitor and control the complex processes executed during the assembly of a car. In this context, a production monitoring and control system stands for a complex central or decentral IT system for collecting, aggregating and processing process signals and values in real time. It has a controlling effect on manufacturing and assembling processes, either in an automated way or by means of user interventions. In line with the definition by (Polke, 1994), a control system is meant to support shop-floor staff in managing their equipment and in controlling and monitoring the production processes. ProVis.Agent® (Sauer & Sutschet, 2006) is one example for such a system. In the following publication, engineering is viewed as a process to which multiple parties contribute using a wide variety of tools (see (Drath, 2008)). Today's world is marked by continuous change and enhancements. To remain competitive, plant operators have to respond quickly to the situation and requirements of the market. Technology has to support this. Any change in the production process has to be considered and the equipment has to be re-configured and re-customized. Today, efficient modifications to manufacturing systems are really difficult and result in an increased demand on adaptivity. For an efficient information exchange, all systems involved have to interact as seamlessly as possible in the heterogeneous environment. This is called interoperability. The cost pressure in modifications to or re-engineering of existing plants and the development of new plants is increasing constantly. This process is both cost-intensive and error-prone and includes a multitude of technical as well as human interfaces. To enhance the mutability of plants, it is necessary to find generic, reusable solutions to meet these very challenges. (Fay et al., 2008) therefore describes an approach to systematically assess engineering organisations and, on this basis, to identify means which are particularly suited to improve engineering within this organisation.
Production monitoring and control systems monitor and control manufacturing plants and systems. Before they can be taken into operation, they have to be configured. To this end, the topology and the structure of the relevant production plant as well as information about the incoming and outgoing interfaces of the components involved in the production process

have to be customized. In addition, process control images are required to visualize the current production activities. At present, the customization of marketable production monitoring and control systems is associated with considerable manual effort since a staff member has to enter plant planning information in the customization tool of the control system. Today, production it takes place at the end of the plant planning process. Potential planning errors are not recognized until the functionality of the production monitoring and control system is tested in the real-world plant. This is very time and cost-intensive. The vision is a plant which can simply be linked with the process by 'plug-and-produce'. The pre-requisite is a self-description containing all information required by the production monitoring and control system. There has to be a consistent, standardized data exchange format that can be used throughout all phases of the life cycle. The potential data sources that have been identified for control technology are the digital factory and the preceding planning phases (see (Bär et al., 2008)). Production monitoring and control system engineering of the future will therefore require a consistent neutral data format. In addition, standardized communication and processing mechanisms are necessary. Various sectors of industry have already specified their formats in detail, e. g. the Weihenstephan standards. Since the approach presented here is to be applied universally, the standards that have been developed for a specific sector and that can only be applied there are unacceptable. The existing XML-based standards in automation technology are largely limited with respect to the potential quantity of information that can be processed. NE100, FDCML and EDDL, for instance, are restricted to information about field devices. However, to obtain all the information required by control technology, data from multiple sources is required. This includes information about the process signals from the relevant field devices as well as plant visualization data and topology information from plant planning or from layout planning. For this reason, a universal format allowing all information from various sources to be collected and retrieved, as necessary, is needed. To avoid the development of just another new, hardly accepted data format, an existing industry standard should be used rather than establishing a completely new one. The independent XML-based CAEX (Computer Aided Engineering Exchange, (IEC 62424)) data exchange format was originally developed in process engineering. Fraunhofer IITB, however, has proved that the format is also suited for the efficient exchange of data in production engineering. In CAEX, the plant data can be managed and transmitted using library structures. This allows norms and standards to be modelled as CAEX libraries as well, satisfying those users that have already made their data comply with these standards. CAEX data can be processed further on the basis of a knowledge-based system, universal rules or analytical, computer-based tasks. The format takes account of the problem that the standardization of the tools available on the market only makes sense to a certain degree, if it makes sense at all. Therefore, a general approach has to be chosen. The exchange of CAE planning data between various systems is structured and organized, and in this application it is used as a standardized data format for the automated engineering of production monitoring and control systems.

In this application, CAEX was complemented by OPC UA (OPC Unified Architecture, (IEC 62541)), the service-oriented successor to the industrial OPC communication standard. OPC UA allows for the communication, synchronization and processing in the prototypical engineering framework. The architectural approach is based on a central OPC UA server and its address space, on which the individual components of the production monitoring and control system, of the associated customization and the process visualization act as OPC

UA clients. The underlying idea is a consistent standard interface, standardized communication with all systems involved, service-oriented processing, investment protection in view of supplier-specific formats and ultimately the enhancement of the quality of data through automated processes.
Combining both standards to form one framework boosts the strengths of each and opens up new potentials for the red-hot topic of "automation of automation". This chapter will present a solution which combines the two standards to form one framework.

2. The CAEX data format (Computer Aided Engineering Exchange)

Now that proprietary, stand-alone software solutions have been superseded, industry calls for a consistent information flow between the individual systems (see Fig. 1). As (RWTH, 2008) has shown, the definition and implementation of universal standard is only possible and useful to a certain degree. The main challenge lies in the vast complexity of application models and aspects.

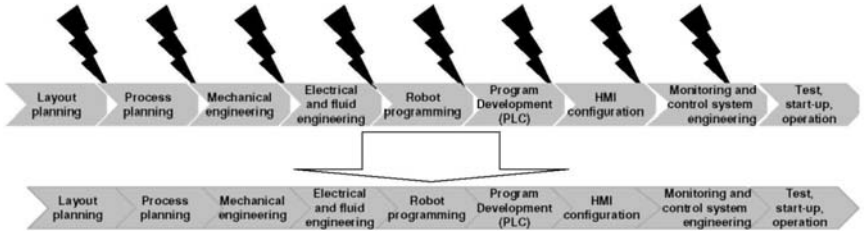


Fig. 1. Consistency of data, a prerequisite for boosting efficiency

CAEX was developed in cooperation between the Department of Process Control Engineering of the University of Technology Aachen, Germany (RWTH Aachen) and the ABB Research Center in Ladenburg. The definition of CAEX has been taken down in the (IEC 62424) standard. CAEX is a semi-formal description language which is based on XML (Extensible Markup Language). The requirements for a data exchange format have been specified in (Draht & Fedai, 2004a). First and foremost, the format should support library concepts and object-oriented approaches. It should be possible to integrate libraries from users and suppliers as well as project libraries. In addition, both a top-down and a bottom-up system design has to be supported.
(Epple, 2003) describes the motivation behind the development of CAEX. It is supposed to be independent of specified standards and protect investment in view of supplier-specific formats. Furthermore, CAEX is to boost the efficiency in data exchange. CAEX originated in process engineering. Nevertheless, it is equally suitable for production or manufacturing technology, as (Schleipen et al., 2008) has shown. In line with the aforementioned requirements, CAEX has been designed as a tool-neutral, object-oriented data format for storing hierarchical plant information. The exact structure and elements of CAEX are described in more detail in (Draht & Fedai, 2004b). CAEX supports concepts such as encapsulation, classes, instances, inheritance, hierarchies and attributes. The CAEX format has been defined as an XML schema.
The plant data can be managed and transmitted in the form of library structures. This allows norms and standards to be modeled as CAEX libraries as well, thus meeting the need of

those customers whose data comply with these norms already. CAEX data can be processed further by means of a knowledge-based system using universal rules or by means of analytical, computer-based tasks. In this context, the abstraction from a specific equipment entity or a specific data structure is essential.
CAEX allows for the standardized storage of planning data (in syntactical terms), supports object-oriented concepts and even enables the storage of unfinished planning stages. The meta modeling techniques simplify the creation of a standard data exchange. Using standardized interfaces, every system involved has to know all structures oder possible data which can be exchanged. Instead of using generalized structures, CAEX defines its own model structures, which are managed in libraries. During data exchange, the CAEX libraries can be transmitted to other systems together with the data. All systems supporting the CAEX data model are capable of exporting the structural and semantic information from the libraries and to interpret the received data. This reduces complexity. Furthermore, the format is extremely scalable. This is useful while modeling whether complex production lines or one single machine. There are multiple advantages of a standardized, tool-independent format like this. Engineers, for example, can re-use valuable engineering results from existing systems, either in full or in part. Thus, they benefit from the experience of the past when planning a new plant or carrying out a new project. In view of the increasing time and cost pressure, this is a significant advantage over starting from scratch. A standardized data format renders production systems more transparent, which makes them easier to compare. In addition, a consistent format facilitates the computer-assisted evaluation and processing of these CAEX data.
At the top level, the CAEX data model (Fig. 2) consists of four areas: three libraries - InterfaceLibrary, RoleLibrary and SystemUnitLibrary - and the specific plant structure - SystemHierarchy.

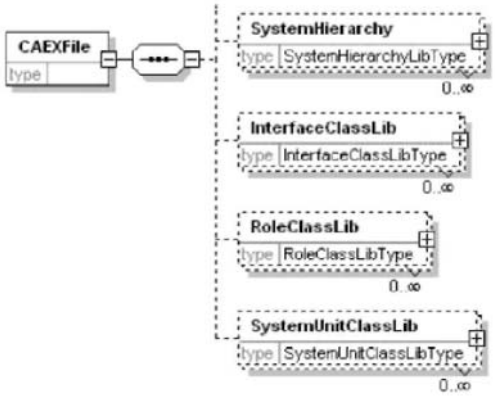


Fig. 2. CAEX data model (top level)

The libraries are specification collections. The 'SystemHierarchy' models the actual topology of the plant. To represent the plant, the instances of the models predefined in the libraries are used, which are then added by up-to-date parameter values. The basic components can be interlinked by means of various pre-defined XML elements within CAEX, allowing various semantic links to be modeled. Interfaces can be allocated to elements from the

RoleLibrary, for example, enabling the creation of links. Elements from the SystemHierarchy get an internal structure through allocated SystemUnits and a functional meaning through allocated roles. As far as the entire CAEX file is concerned, its structure is shown in Fig. 3. It will visualize the connection between the individual basic components and serve as an illustration for the explanations below.

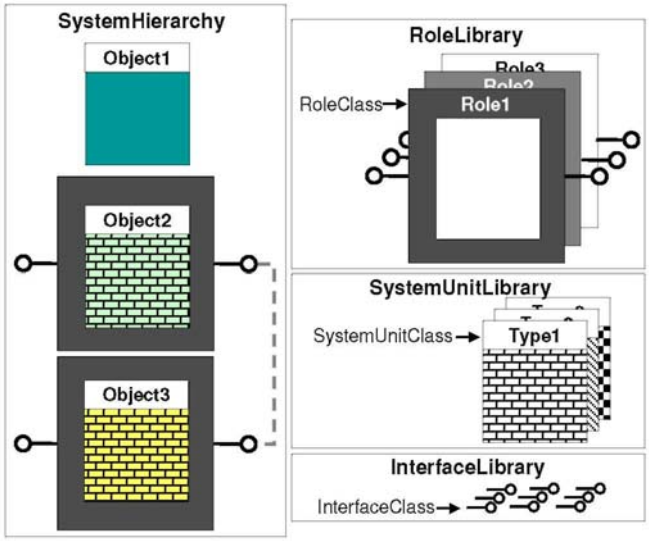


Fig. 3. Basic CAEX components and their connections

The InterfaceLibrary is the library of the interface classes. A class defines a type of interface, the link between two elements. The implemented interfaces allow two objects to be interconnected. The interface determines the type of relation and the semantic meaning of this connection. Interfaces may contain attributes. A plant topology interface, for example, may possess the attribute 'direction'.

In the RoleLibrary, the functional roles of the objects are described, e. g. a conveyor belt, a welding cell or a turntable. The role of an equipment component does not possess any information about the internal structure of the system. Instead, it describes the semantics as well as the available interfaces and general attributes. The abstraction through the role allows the symbolic parameter of a conveyor belt, for example, and thus the graphical object for a conveyor belt, to be assigned to a piece of equipment. The technical realization, the installation and implementation can be carried out at a later point in time, independently from previous steps. Nevertheless, this feature enables all conveyor belts, for example, to be visualized by a consistent graphical object. Both a unidirectional and a bidirectional conveyor belt, for instance, may be allocated the role of 'conveyor belt'. The roles thus implement an 'I am a ...' relationship.

The SystemUnitLibrary describes complex physical or logical equipment components ('SystemUnits'). They are similar to classes in object-oriented programming. Unlike roles, these elements possess information regarding the functionality and the internal structure of the elements. These elements can be compared with types or, in other words, they can act as

a combination of assembly units. The 'SystemUnit' node introduces a new structural description. Each structure can be made up hierarchically and use other, predefined structures as sub-elements. Therefore, it is not necessary to define the contents of the individual sub-elements at the type level. For this purpose, there are system classes that go beyond the usual CAEX structures in the application presented here. The system classes describe the elements that exist within the system in detail, including the contents and the meaning they have. The type only refers to this. In addition, the equipment element can implement a role defined in the role class library. The role allocation may result in other role-specific attributes and interfaces being added to the element, determining the way in which it interacts with other elements. The SystemUnitLibrary only defines the structure of the elements, while it does not allocate any attribute values. The only possible value specifications in the structural definition are those of the default values for the attributes. However, they only serve as initial values and thus form part of the definition; they may be overwritten in the description of the specific plant.

In addition to the aforementioned libraries, the CAEX meta model has a SystemHierarchy, where specific objects (InternalElements) and their planning data are managed and stored hierarchically. If a SystemUnit has been defined in the SystemUnitLibrary, a specific object of this system unit can then be included in the SystemHierarchy. This allows the real-world plant to be viewed as a structure made up of instances of SystemUnits. However, it is also possible to define instances without underlying SystemUnits. Subsequently, specific values are allocated to the attributes of the elements. In addition, the connections (called InternalLinks) between the elements can be modeled. They allow the plant topology, specific process sequences or assembly sequences, for instance, to be mapped. A good example of this kind of connection would be the link between an equipment component and its successor or specific products manufactured there.

The CAEX data model applied by Fraunhofer IITB has one more special feature. To allow for the electronic description of all ‘elements’ involved in the production process, these are broken down into three categories in practice: resources, products and processes. Control technology usually focuses on resources. The engineering framework classifies all existing system elements into three categories - products, processes and resources (PPR approach). Products stand for all products and product components, processes include all kinds of activities, and resources comprise equipment, staff, software, etc. This classification brings about additional semantic meaning for the system elements, such as 'I am a product', 'I am a resource' or 'I am a process'. The individual types of elements can be linked with each other, with resources being the central component in this model as resources execute processes and resources process products.

In addition to describing the production systems more accurately, the further classification is also aimed at a more detailed description of the products and processes involved. According to (Schleipen et al., 2008) , this makes sense because this information is then available for the 'control technology of the future' or for 'components of manufacturing execution systems (MES)'. For the classification, the RoleLibrary, the SystemUnitLibrary and the SystemHierarchy of the CAEX model are relevant. In these areas, the structures of all three types - products, processes and resources - can be defined and/or applied. The InterfaceLibrary allows interfaces available to all types of elements to be defined. For this reason, it is not broken down into different categories. The classification takes place at the highest level in each case, before the individual elements are defined or deployed. This

ensures that all elements defined at a lower hierarchy inherit the semantic meaning of the common parent node. Fig. 5 shows how the data is embedded in the CAEX PPR model. In spite of the enhancements introduced by the description semantics developed by Fraunhofer IITB, the resulting CAEX file complies with the CAEX schema, of course.

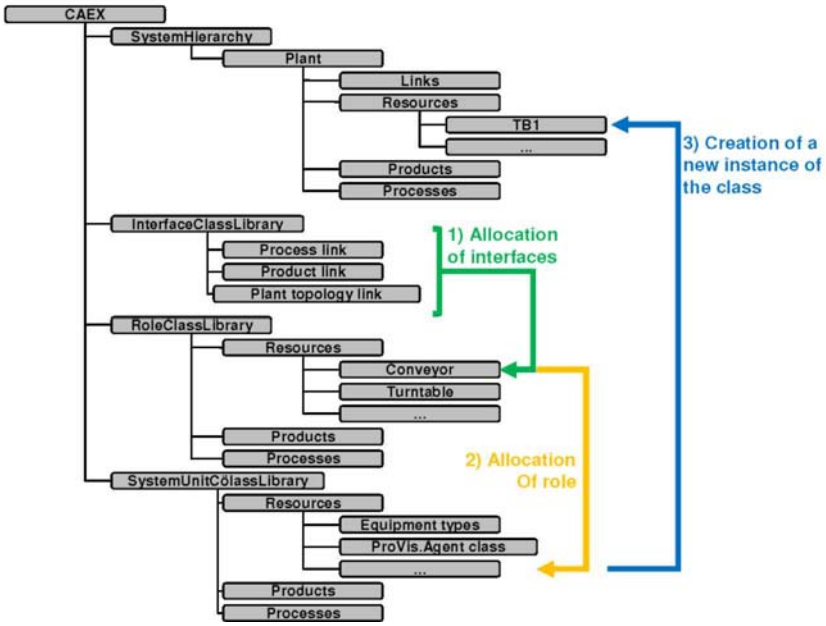


Fig. 4. CAEX PPR data model

The InterfaceLibrary is broken down into the elements 'plant topology link', 'product link' and 'process link'. The interfaces defined in this area allow two elements of the other structures to be interconnected. To be able to link two elements with each other using InternalLinks, both of them have to implement the same type of interface, i. e. support the same kind of connection. The 'plant topology link' enables two resources to be linked. The 'product link' connects a resource with a product, while the 'process link' links a resource with a process. Fig. 5 shows a sample situation in which the TBI conveyor belt transports a car shell and is topologically linked with the DT1 turntable. Interfaces are not assigned directly to the elements; instead, they are allocated by means of the functional roles. Roles are defined in the RoleLibrary. On the one hand, roles specify the class to which an element belongs, such as conveyor belt or turntable. On the other hand, they determine the way in which it can communicate with other elements by implementing the interfaces. At the topmost level, all roles are classified into product, process and resource roles. Each lower level implies a more detailed specification of the role features. This behaviour is in line with the concept of inheritance in object-oriented programming. By being allocated a role, the 'TBI' element (Fig. 5), for example, gets the following information. 'I am a resource, namely a conveyor belt. I can execute processes, process products and form part of a plant topology.'

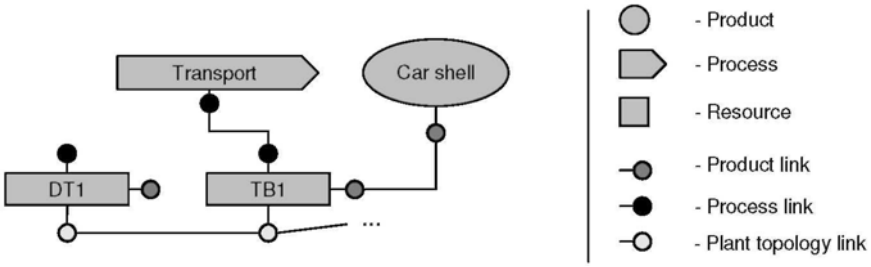


Fig. 5. Linking the elements by means of the implemented interfaces

The SystemUnitLibrary is also broken down into product, process and resource. In addition, the three top categories each have a class for their type basis. The resource class thus has an 'equipment type basis', where the available types of equipment are defined. This is where structural templates are specified as well, which reflect the basic structure of the tool. In the case of 'ProVis', this would be a 'ProVis' class, for example, which describes all potential types of process variables.

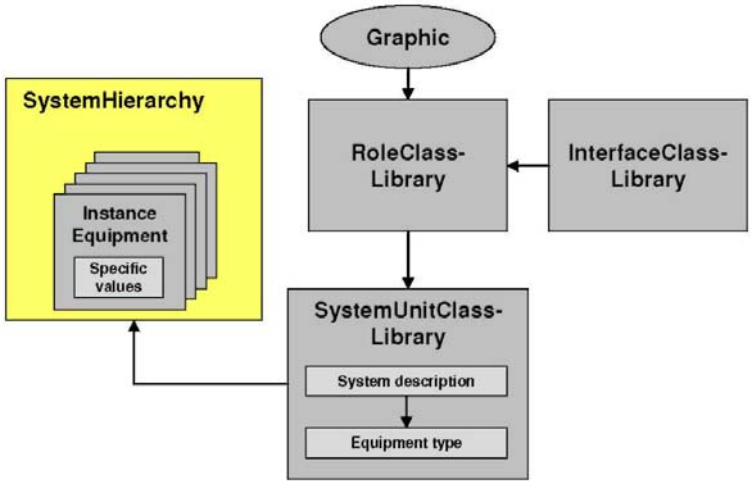


Fig. 6. Creating a specific equipment object

To create a specific equipment object, e. g. a 'conveyor belt', the following steps should be taken. First of all, a type of equipment defined in the SystemUnitClassLibary is instantiated. A role is, in turn, allocated to the type of equipment. Furthermore, it is possible to assign a graphical object to the role, which allows the conveyor belt to be visualized graphically (see Fig. 6). Within the framework of the SystemHierarchy, specific values and attributes are assigned to this element.

3. The OPC UA communication standard (OPC Unified Architecture)

In 1996, the specification of standardized communication originated. It was meant as an alternative to proprietary automation buses and allow for communication between applications stemming from different suppliers. This is how OLE for Process Control (or OPC) was created. OPC, however, does not stand for just one partial functionality. Instead, it provides multiple features. These include OPC Data Access (DA), which enables PLCs to transmit real-time data to visualization devices. Other examples are Alarm&Events, Batch, Historical Data Access or XML Data Access. The OPC specification is implemented by means of software. To enable communication between applications, OPC relies on DCOM (Microsoft Distributed Component Object Model), a state-of-the-art technology of that period. It has a client-server-based architecture in which the server provides data, which is, in turn, accessed by the client.

In view of the fact that the original OPC specifications had evolved historically and the technological capabilities had been enhanced in the course of the years, the OPC Foundation presented the successor of OPC (OPC Foundation, 2009) in 2006. The OPC Unified Architecture (or OPC UA) standardizes the existing specifications. Instead of continuing to use DCOM, OPC UA is based on a service-oriented architecture (in short SOA), which is independent of platforms and operating systems. OPC UA allows for both the horizontal and the vertical exchange of data in a multi-functional server. This ensures that it can be used more universally not only at a field level but also at the level of MES or ERP systems. In addition, OPC UA supports asynchronous and distributed communication while ensuring security, reliability and redundancy by providing appropriate features. At the heart of OPC UA (Fig. 7), there are the abstract method descriptions, also called base services, which are translated into a protocol by the transport layer.

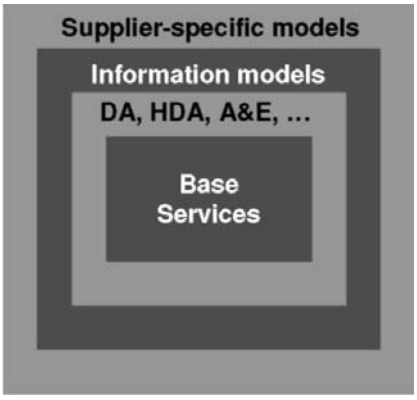


Fig. 7. OPC UA model

The information model does not only consist of a hierarchy of 'folders', 'items' and 'properties' as was the case with OPC. Instead, it is a full-mesh network made up of nodes allowing all kinds of meta and diagnosis information to be transmitted. In this context, a node is an object containing attributes for read access (similar to Data Access (DA) and Historical DA). It owns methods that can be called (similar to Commands) and events that can be sent (similar to DA DataChange). The address space specified by the information

model includes a type model allowing all types of data to be described. On this basis, various other organizations such as EDDL are specifying their own information models. The integration of the CAEX meta model is another example of this trend. As a consequence, any world model can be modelled in the address space of the OPC UA server. Thus, there is no limit to the ways in which it can be applied in multiple sectors of industry.
Applications for OPC UA are developed with the help of APIs, which the OPC Foundation provides to software developers. Owing to the multitude of options OPC UA offers, it was chosen as the standard on which the communication within the framework should be based. The architectural concept of the engineering framework is based on a central OPC UA server and its address space, on which the individual components of the framework operate as OPC UA clients. The information model of the address space was modeled in line with recommendations for the specification (OPC Foundation, 2006). The clients (i. e. the individual components) are capable of integrating additional information in the address space, such as creating new nodes. In this context, the components only observe those parts of the address space that are relevant to them (see Fig. 8).

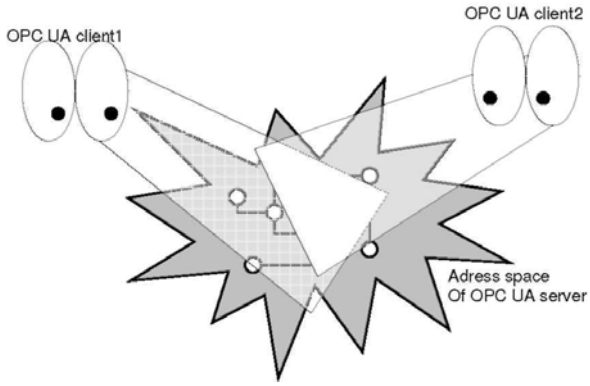


Fig. 8. Detail of the architectural concept

They are notified by the server as soon as some information that is important to them has been received. Hence, the OPC UA server supports the synchronization of processes in the monitoring and control system. If new, CAEX-compliant data is integrated in the engineering framework, the appropriate OPC UA client will instruct the server to create a new node. The client in charge of transforming the CAEX data is notified when the new node has been generated and receives the information it requires. This mechanism is identical for all other clients. The implementation is based on the associated base services (OPC Foundation, 2007). For further details, please see (Schleipen, 2008).

4. Combination of both standards: the engineering framework for the production monitoring and control system

If two systems are to communicate with each other, the two decisive issues are

- "What is communicated?" - The required contents have to be structured. In addition, the meaning of the contents have to be clear.

- "How does communication work?" - The communication mechanisms have to be specified. This includes both the process and the methods, etc.

CAEX is the answer to the first question. It structures data and fills them with semantic meaning. OPC UA, by contrast, is the answer to the second question. It assists communication and information processing in the engineering framework. In this context, CAEX is responsible for the static part, whereas OPC UA handles the dynamics of the procedures and data. The application of this combination of standards is not restricted to production monitoring and control systems and their process visualization. Instead, it can be applied in multiple ways (see Fig. 9).

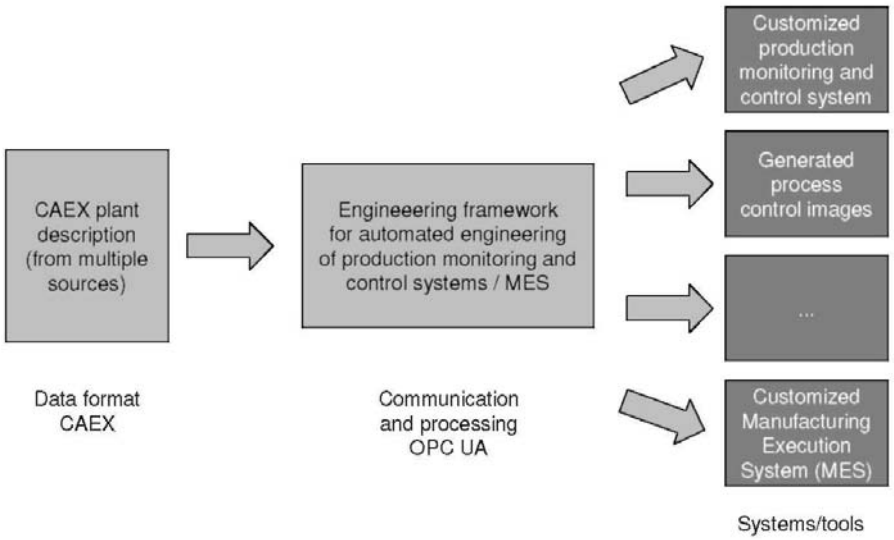


Fig. 9. Functionality of the engineering framework

To evaluate the engineering framework, various sample applications have been created. These include the customization of the production monitoring and control system and the generation of process images, which are parameterized by means of a layout manager and do not only represent resources but also additional products and processes. These sample cases have been tested using various sample systems. For one thing, there are various conveyor systems at hand to validate customization and simple visualization. They are shown in Fig. 10 and include conveyor belts and turntables as well as a test station or welding cell.

For another, a hierarchical example (Fig. 11) was developed to test the layout manager. The topmost hierarchical level consists of two equipment aggregates called TA1 and TA2. They are aggregations of several pieces of equipment. At the second hierarchical level, the TA1 equipment aggregate consists of two conveyor belts (TB1 and TB2), a DT1 turntable and another equipment aggregate called TA3. The TA2 equipment aggregate includes two conveyor belts (TB3 and TB4) and a DT2 turntable. The third and thus lowest hierarchical level is represented by the TA3 equipment aggregate. It is made up of a DT3 turntable, a TS1 test station and a TB5 conveyor belt. Thus, the sample application consists of twelve pieces

of equipment in total, three of which are aggregates (the equipment aggregates TA1 to TA3) and nine of which are 'genuine' pieces of equipment.
To allow for the overall visualization of resources, processes and products, one of the conveyor belt examples was complemented by processes and products (Fig. 12).

Fig. 10. Sample pplications

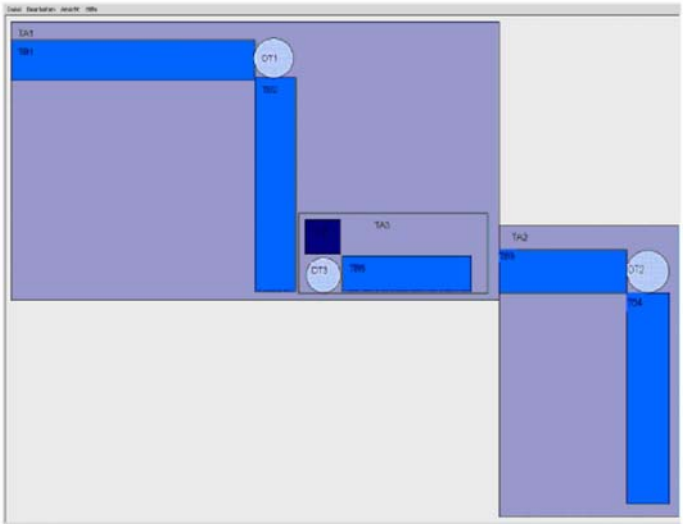


Fig. 11. Hierarchy-based sample application

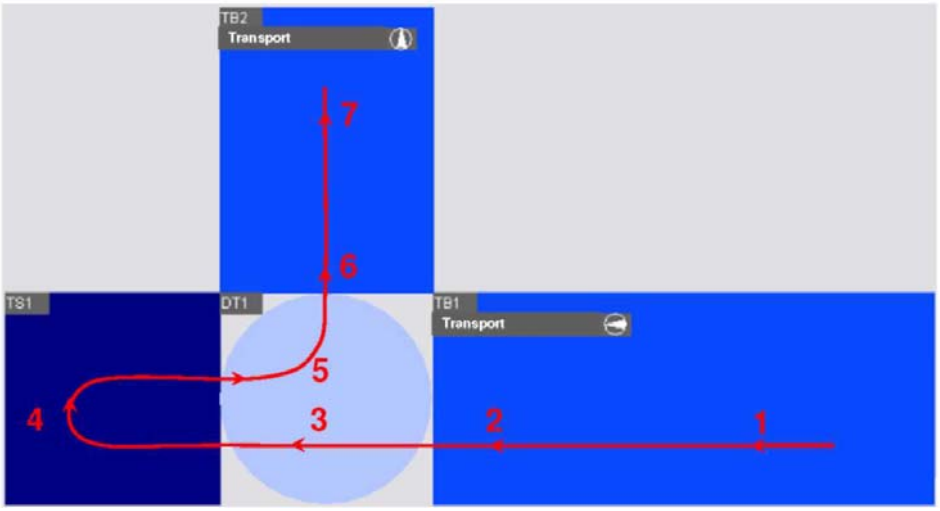


Fig. 12. Basic image of the sample application for resources, products and processes

The following paragraphs will outline the workflow applied for engineering within the framework. If a CAEX description is available for the equipment (see Fig. 13), it is transmitted to the change management of the production monitoring and control system on the basis of OPC UA.

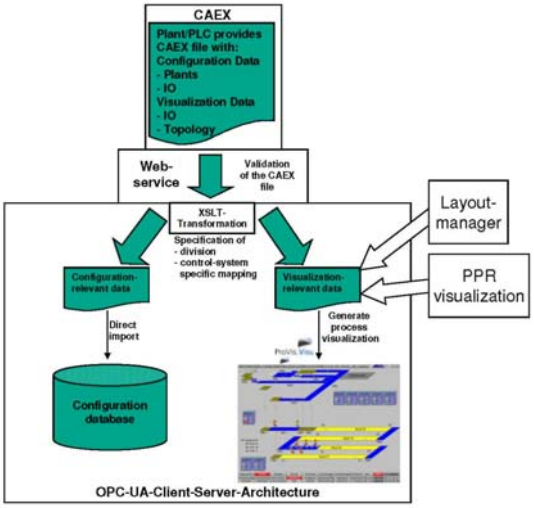


Fig. 13. The engineering framework

The provided CAEX data is validated against the CAEX XML schema with respect to structural correctness. Subsequently, it can be processed further on the basis of the structure and semantics. For this purpose, the mapping based on system descriptions or templates is

used to convert the data into a production monitoring and control system specific CAEX format. To generate this kind of mapping between two CAEX files from different suppliers and/or levels, the structural templates of the SystemUnitLibrary are considered. In the case of a mapping between a 'ProVis' CAEX file and the CAEX file provided by the equipment or PLC, the 'ProVis' class and the PLC class in which the available structures including all attributes are specified are considered. To make things easier, the data types and the units of the relevant attributes are specified there, too. At first sight, this resembles a proprietary interface. However, the mappings are independent of the actual data structure in the tool and can be generated and converted more easily thanks to XML and the predefined structures, since the converter using the mappings has to be created just once. In addition, graphical support is available, making things even easier for users.
Assisted by the central OPC UA server and several OPC UA clients, the pre-processed data is used both to customize the production monitoring and control system and to generate the process control images for ProVis.Visu® in an automated way using a layout manager. The CAEX document is split into data relevant for configuration and visualization (see left and right path of Fig. 13). The resulting sets of data are transmitted to the configuration and visualization components of the ProVis production suite respectively. The visualization data is used to generate the process images. The configuration-relevant part is imported into a database. The system can use this data to perform the I/O and plant customization for the process image of the runtime system (see (Schleipen et al., 2008)). This considerably reduces the manual and thus error-prone part of customization, as the production monitoring and control system (CS) plant configuration, the CS I/O customization and the CS image customization are, in part, performed automatically (also see (Sauer & Ebel, 2007)). In this process, the generation of the process control images as the human-machine interface has to be considered above all. In manufacturing, an automated system for image customization will only be accepted if the user interface is user-friendly and intuitive. The special field of human engineering (Syrbe, 1970) aims to adapt machinery and other technical equipment to humans to optimize their cooperation. The characteristics, potentials and requirements of human beings are taken into account, and the visualization of machinery and/or equipment is based on these conditions. For this reason, human engineering deals with both the physical/ physiological and the mental characteristics of human beings (Charwat, 1994).
In (Syrbe, 1970), the following seven rules are presented which form the basis of the high-quality design of the human-machine interface:

- "Mind the properties of the sense organs"
- "Depict process states in a task-dependant way"
- "Choose an attractive design which directly corresponds to the task"
- "Avoid information unnecessary for fulfilling the task (noise information)"
- "Mind the unconscious attention control of human beings"
- "Mind population-stereotypical expectations"
- "Design correlating display and operation elements in a strikingly similar way and those that do not correlate in a particularly divergent way"

In this process, visualization is to be based on ergonomic guidelines. In addition, appropriate algorithms have to be developed to position the existing equipment components as well as I/O signals on the process control image in line with the actual layout. Finally, users should be able to adapt the process control images to their personal requirements.

If users create process control images manually, the very same process may be depicted differently depending on the preferences of the person who has drafted them. In contrast to process engineering, there are no standards such as P&I diagrams in DIN EN ISO 10628 in production and manufacturing technology specifying the layout of certain components. As a consequence, the same processes do not necessarily look identical in visualization. In addition, the manual generation of images has the drawback that it is time-intensive and error-prone. Thus, the process control images should be as standardized as possible, while, at the same time, being as individual as necessary.
The layout of the visualization was defined in various views (Schleipen & Schick, 2008), allowing for a topological and a structural overview of the entire plant and making it possible to zoom into the equipment. Moreover, the equipment can be operated in line with the potentials of the system. The topological view visualizes the topology of the equipment to be monitored. In this context, it should be possible to zoom into the equipment. To this end, a hierarchical level model was created allowing several equipment aggregates to be combined to form a larger system. Fig. 14 illustrates this approach. It enables entire production halls to be visualized clearly in just one image while ensuring that the most important information such as faulty states in the aggregations, also called 'collective alarms', are displayed.

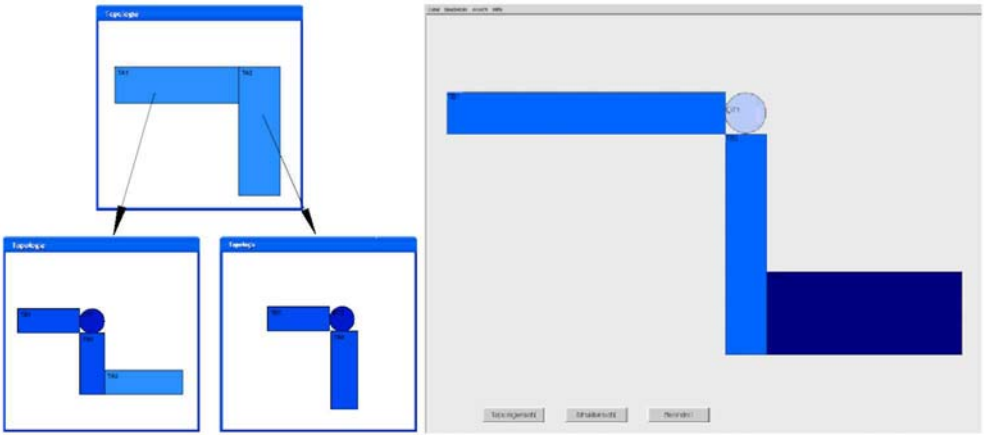


Fig. 14. Concept and implementation of the topological view (Schleipen & Schick, 2008)

The structural view (see Fig. 15) provides an overview of the signals of the existing pieces of equipment to users. Every line stands for a piece of equipment contained in the overall plant. The other elements represent the linked process variables, their slots and their current values.

Fig. 15. Structural view (Schleipen & Schick, 2008)

The operational view shown in Fig. 16 allows the users to operate the plant they monitor. It only displays the process variables of one piece of equipment rather than those of the entire equipment, as is the case in the structural view.

Fig. 16. Operational view (Schleipen & Schick, 2008)

The design of all views is based on ergonomic requirements (also see DIN EN ISO 9241-12). To enter the user-specific settings, a graphical user interface was created. Nevertheless, the basic structure is maintained when the process control images are generated. Otherwise, there would be the problem that the images vary considerably depending on who has created them, as was the case with the manual generation of the process images. For the topological view, users can determine the piece of equipment that forms the highest hierarchical level/the most interesting part of the plant. In a second step, it is possible to

define the process variables and slots that are to be represented in the structural and/or operational view. The last part, the 'representation', defines the color specifications or the path to the bitmap graphic that is to be used to visualize the equipment. This information can, in part, be extracted from the CAEX descriptions. In addition, users may store all the settings they have chosen.

In addition to visualizing equipment, ongoing processes and processed products have to be represented. An approach to the practice-oriented representation of products and processes has been developed and implemented. Combined with a product identification system, the products can thus be mapped at their current position. This allows the products and the progress of the process to be traced by linking ident systems, for example, with control technology. In addition to visualizing plants, in this component, participating processes and products can be visualized as well. This allows the products to be traced and the progress of the process to be monitored based on dynamically changed CAEX data. If the CAEX model is contained in the address space of the OPC UA server, it is at all times possible to identify the processes currently executed and the product processed by the system. The structure of elements, equipment and products within the images is made up dynamically, as is the allocation of products and processes to a resource (piece of equipment). To visualize movements of products and changes in current processes, it is required to map the current production situation at regular intervals. Changes in processes and product positions do not have to be visualized in real-time for production monitoring and control technology. To update product and process representations in process visualization, intervals of five seconds (or a maximum of ten seconds) are sufficient in this case. The process signals, by contrast, continue to be visualized in real-time. The presented information has to be as clear as possible, allowing even inexperienced users to interpret them intuitively. Image generation is to comply with the engineering general approach. In addition, the universality of the CAEX model should not be restricted by image generation. Since control technology only visualizes abstracted production process information, there is no need to visualize complete products or processes there. Rather, it is sufficient to visualize products as bitmaps at discrete positions of the resources. Text-based process information providing an option to access additional data will do. In addition, the visualization of the direction of the process (flow) is important because it can provide valuable hints for detecting potential faults in the production process.

The resource and process names are shown in a text field. To ensure that the provided information does not conceal the image elements located next to the resource, they are positioned in the top left corner of the resource. For visibility and readability reasons, the texts are presented against a neutral background in the form of information bars. Depending on the information provided by the resource, the relevant bar is either shown or hidden completely. The layout of the information bars consists of a dark gray background and white fonts. This colour combination ensures that the text can be read clearly (see Fig. 17, top).

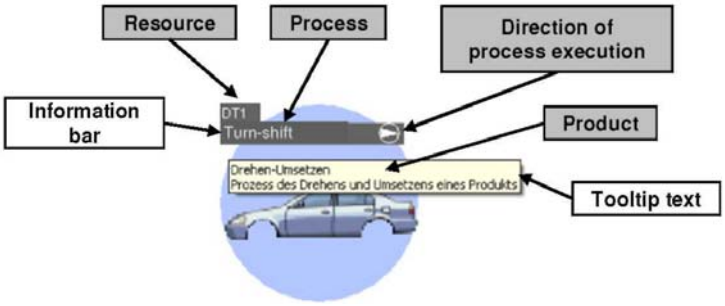


Fig. 17. Information bar and tooltip text for process, resource and product

In the case of products, the information cannot be shown in a bar. In most cases, the objects that represent products are too small and would be hidden by the bars. This problem has been solved by using the tooltip text property of the graphical elements for all additional information. Process elements and resources can possess additional information other than the name. This information is also visualized by tooltip texts placed directly upon the graphical elements representing the product. To enable users to understand how the process works in the real world, another attribute called 'direction' is introduced for the resource definition. This attribute is to indicate the direction in which the process is executed in the plant. In visual terms, the direction can be symbolized by an arrow (see Fig. 17). Fig. 18 shows a generated resource process product visualization at the point in time t. At that time, the car shell kar0011 is on the TBI conveyor belt (with the state 'transport to the left'), on its way to the TS1 test station. Another car shell that has been tested already is on the DT1 turntable, having the state 'turn-shift to the right'.

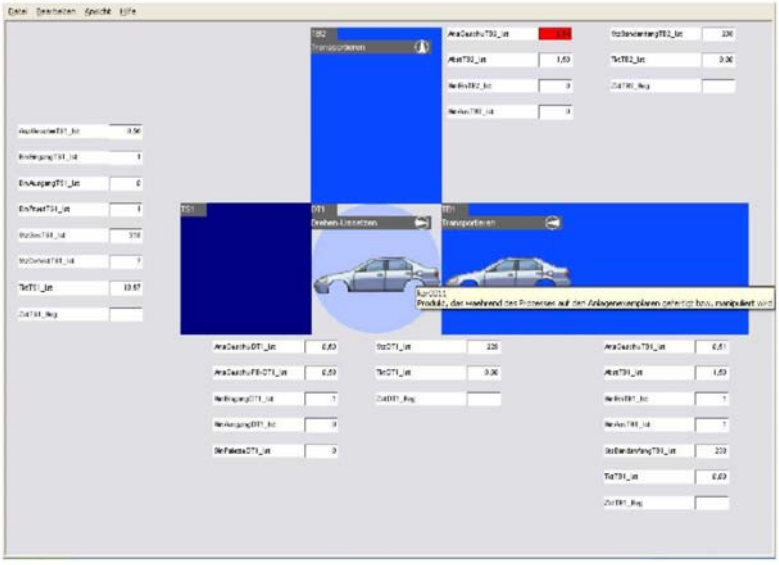


Fig. 18. Generated product process resource visualization

5. Conclusion

The engineering framework presented in this contribution allows data to be processed and communicated electronically for production monitoring and control technology. This ensures that a fault-resistant, semi-automated production monitoring and control system engineering can be realized. Hence, control technology as a representative of IT systems in plant operation can be linked with planning at an early stage. Fig. 19 shows the benefit of the earlier coupling of planning and the customization of production monitoring and control systems. It increases efficiency in time and thus costs in the delivery and customization of production monitoring and control technology

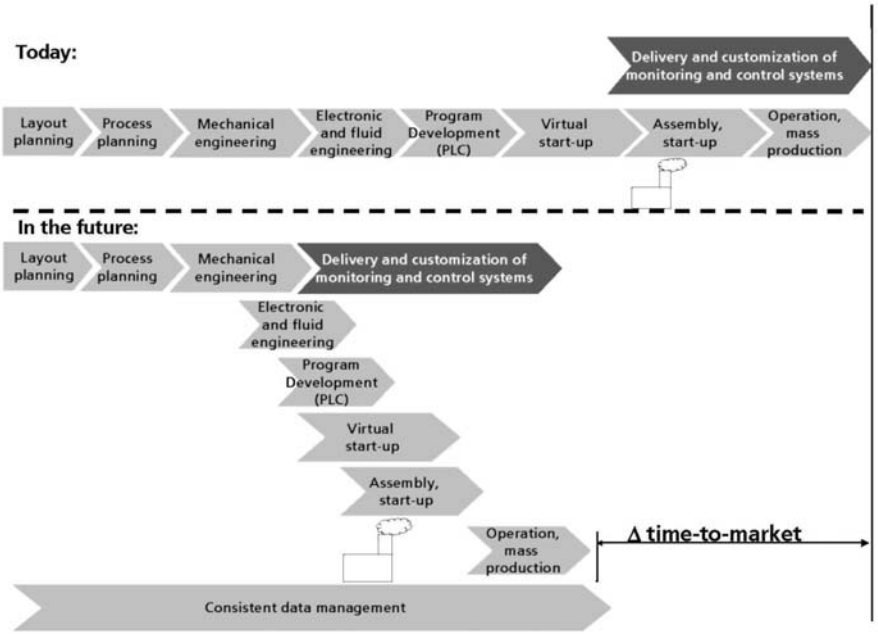


Fig. 19. Early coupling of planning and the customization of production monitoring and control systems according to (VDI 4499-2)

In parallel to the engineering framework, there are new potential fields of deployment. When starting up a system, complex production monitoring and control system can be parameterized and tested using simulations even before the software is actually taken into operation. To this end, the system simulation has to be controlled by a PLC that does not form part of the simulation program so that control technology can access the simulation signals. Control technology can then be based on these real signals using the well-known communication mechanisms of automation technology, e. g. OPC. Fig. 20 provides a schematic overview of this kind of coupling. For the production monitoring and control system, it is insignificant whether the OPC signals stem from a real-world production process or from a PLC linked with simulation. If this kind of link is in place, the data processed and/or generated in the production monitoring and control technology can be used to improve the input data of simulation. This enables control technology to be tested at

an early stage. Furthermore, it serves as an additional data source for simulation. This type of data includes evaluations from the production monitoring and control system, for instance. An evaluation of the process data in the production monitoring and control system allows for the provision of quality features for executable configurations in simulation.

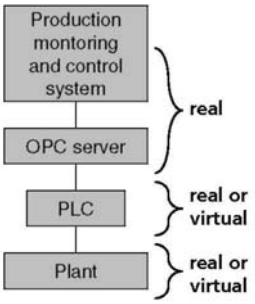


Fig. 20. Link between simulation and production monitoring and control system using OPC

The development of the engineering framework includes considerations regarding the legal consequences that result from the findings for the individual users. These consequences can be classified as follows: contractual problems, product liability, safety of equipment and industrial law. As a general rule, the results generated automatically should, at any rate, be approved by technical specialists, in accordance with legal experts. Staff members should receive appropriate training and be familiar with topics such as responsibility for defaults, product liability and CE marking to be able to perform a plausibility check for the created software and configuration and to judge it. In this process, they can be supported by a checklist that has to be observed in this case. This ensures that important and necessary topics and aspects are taken into account. As far as liability is concerned, there are various parties dealing with these topics or problems. These include the software suppliers who are liable for the software they provide. In addition, they include the manufacturers or suppliers of machinery who are responsible for the machine. Finally, it is the operator of the plant who is liable once the system has been taken into operation. Vis-ä-vis the final customer, these parties may have joint liability and take responsibility for flaws in the products produced. As a consequence, there are aspects other than technological potentials that play an important part and that have to be considered and observed.

6. References

Bär, T.; Mandel, S.; Sauer, O.; Ebel, M. (2008) Durchgängiges Datenmanagement durch plug-and-work zur virtuellen Linieninbetriebnahme, *Proceedings of 2. Karlsruher Leittechnischen Kolloquium,* pp. 105-122, 978-3-8167-7626-0, Karlsruhe, Mai 2008, Fraunhofer IRB Verlag, Stuttgart

Charwat, H. J. (1994) *Lexikon der Mensch-Maschine-Kommunikation,* Oldenbourg, 3486226185, München

Drath, R.; Fedai, M. (2004) CAEX - ein neutrales Datenaustauschformat für Anlagendaten - Teil 1, *atp - Automatisierungstechnische Praxis,* Vol. 46 *(2004),* No.2, (February 2009) 52-56, 0178-2320

Drath, R.; Fedai, M. (2004) CAEX - ein neutrales Datenaustauschformat für Anlagendaten - Teil 2, *atp - Automatisierungstechnische Praxis,* Vol. 46 (2004), No.3, (March 2009) 2027, 0178-2320

DIN EN ISO 9241-12:2000-08 Ergonomische Anforderungen für Bürotätigkeit mit Bildschirmgeräten - Teil 12: Informationsdarstellung (ISO 9241-12:1998), Deutsche Fassung EN ISO 9241-12:1998, Beuth, Berlin

DIN EN ISO 10628:2001-03 Fließschemata für verfahrenstechnische Anlagen – Allgemeine Regeln (ISO 10628:1997, Beuth, Berlin

Drath, R. (2008) Die Zukunft des Engineering - Herausforderungen an das Engineering von fertigungs- und verfahrenstechnischen Anlagen, *Proceedings of 2. Karlsruher Leittechnischen Kolloquium,* pp. 33-40, 978-3-8167-7626-0, Karlsruhe, Mai 2008, Fraunhofer IRB Verlag, Stuttgart

Epple, U. (2003) Austausch von Anlagenplanungsdaten auf der Grundlage von Metamodellen, *atp - Automatisierungstechnische Praxis,* Vol.45 *(2003),* No.7, (July 2009) 61-70, 0178-2320

Fay, A.; Schleipen, S.; Mühlhause, M. (2009) Wie kann man den Engineering-Prozess systematisch verbessern?, *atp - Automatisierungstechnische Praxis,* Vol.52 *(2009),* No.1-2, (January 2009) 80-85, 0178-2320

IEC 62424: Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE-CAE tools, text English.

IEC 62541: OPC Unified Architecture

OPC Foundation (2006) OPC UA Part 3 - Address Space Model 1.00 Specification, July 2006

OPC Foundation (2007) OPC UA Part 4 - Services DRAFT 1.01.05 Specification, February 2007

OPC Foundation (2009) OPC Unified Architecture, http://www.opcfoundation.org, April 2009

Polke, M. (edit.) (1994) *Prozeßleittechnik,* Oldenbourg Verlag, 3486225499, München

RWTH Aachen, Lehrstuhl für Prozessleittechnik (2008) ACPLT: CAEX-IEC62424, http://www.plt.rwth-aachen.de/index.php?id=228&L=1ks

Sauer, O.; Sutschet. G. (2006) ProVis.Agent: ein agentenorientiertes Leitsystem - erste Erfahrungen im industriellen Einsatz, *Proceedings of VDE-Kongress 2006,* pp. 297302, 978-3-8007-2979-1, Aachen, October 2006, VDE Verlag, Berlin-Offenbach

Sauer, O.; Ebel, M. (2007) Engineering of production monitoring & control systems, *Proceedings of 52nd IWK - Computer science meets automation,* pp.237-244, 3939473170, Illmenau, September 2007, TU Ilmenau Universitätsbibliothek, Illmenau

Schleipen, M. (2008) OPC UA supporting the automated engineering of production monitoring and control systems, *Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation ETFA,* pp. 640-647, 1-4244-1506-3, Hamburg, September 2008, IEEEPress

Schleipen, M. ; Drath, R.; Sauer, O. (2008) The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system, *Proceedings of IEEE International Symposium on Industrial Electronics - ISIE 2008,* pp.1786-1791, 978-1-4244-1666-0, Cambridge, June 2008, IEEEPress

Schleipen, M.; Schick, K. (2008) Self-configuring visualization of a production monitoring and control system, *Proceedings of CIRP International Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME 08,* 978-88-900948-7-3, Naples, July 2008

Syrbe, M. (1970) Anthropotechnik, eine Disziplin der Anlagenplanung, *Elektrotechnische Zeitschrift,* Vol. 91 (1970) 12, No. A, (1970) 692-697, 00137359

VDI 4499-2: Digitaler Fabrikbetrieb, Gründruck, 2009

Real-time Obstacle Avoidance Using Potential Field for a Nonholonomic Vehicle

Hiroaki Seki, Yoshitsugu Kamiya and Masatoshi Hikizu
Kanazawa University
Japan

Fig. 1. Autonomous wheelchair moving through a narrow space.

1. Introduction

Obstacle avoidance is an important function for intelligent vehicles and mobile robots. Let's discuss about the obstacle avoidance for a nonholonomic vehicle (mobile robot) like an autonomous wheelchair (Fig. 1). It has two independently driven wheels and a body with a certain shape. If a vehicle can be treated as an omnidirectional movable point, numerous methods have been proposed and applied for it (Fig. 2). Collision free path can be easily found by artificial potential field (Khatib, 1986; Rimon & Koditsuchek, 1992), graph theory (Ulrich & Borenstein, 2000), sensor based method and so on. The problem for a nonholonomic vehicle with two independently driven wheels can come down to that for an omnidirectional point by approximating vehicle's shape to a circle with the center at the midpoint of two wheels. As shown in Fig. 3, obstacles should be expanded by the radius of the vehicle's circle and the

vehicle should be contracted to a point. However, it isn't reasonable to regard the rectangular body like a wheelchair as a circle and its circle sometimes can't pass through the narrow place where the original body can do.

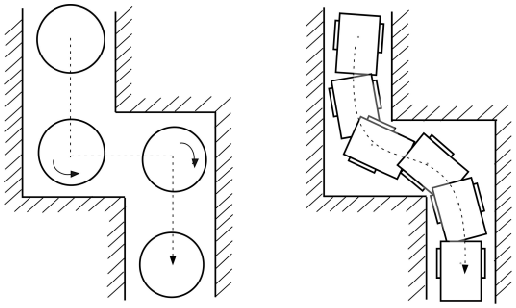
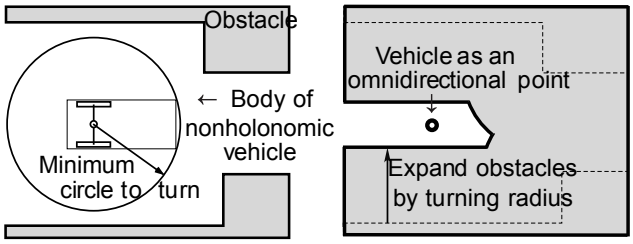


Fig. 2. Obstacle avoidance is easy for an omnidirectional vehicle, however, it is difficult for a vehicle with motion constraint and rectangular body.



(a)Before expand obstacles (b) After expand obstacles

Fig. 3. Approximation of vehicle's shape by a circle for path planning.

In case of an omnidirectional (holonomic) vehicle, "configuration space" can be used for its path planning when the vehicle's shape is considered explicitly (Strobel, 1999). This problem is named "piano movers' problem" (Schwartz & Sharir, 2983). A set of position and orientation where a vehicle body doesn't collide with obstacles is represented by three dimensional configuration space (Fig. 4). A path of vehicle's position and orientation should be searched in this space by probabilistic roadmap method (Kavraki et al., 1996) for example. There are some studies considering both shape of vehicle's body and nonholonomic motion (Kondak & Hommel, 2001; Minguez et al., 2006; Ramirez & Zeghloul, 2001). It is very difficult problem to search a path in the configuration space under the motion constraint. Laumond (Laumond et al., 1994) solved this by modifying the collision free path obtained without motion constraint so as to satisfy motion constraint. Latombe (Latombe, 1991) proposed that the configuration space is divided into cells, the cells where a nonholonomic vehicle can move by simple motion such as turning, going straight, pulling over are connected by graph, and a path is searched in the graph. Anyway, these methods are too complicated for real-time obstacle avoidance using real sensor information although these ensure the solution of collision free path. Specially, calculation of configuration space needs much computing power.

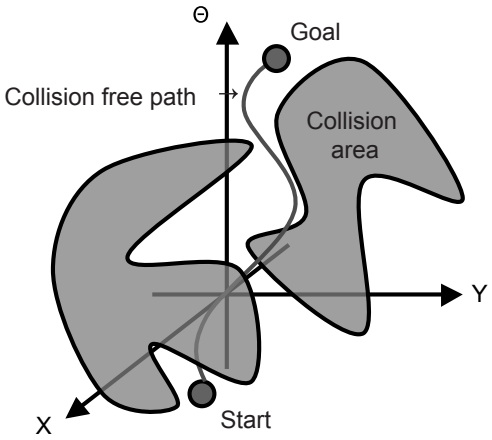


Fig. 4. 3D Configuration space for a vehicle with a certain shape

Therefore, we propose a practical method of local obstacle avoidance for a nonholonomic vehicle with rectangular body. Simple potential field using local sensor information of surrounding obstacles is applied.

2. Problem Statement

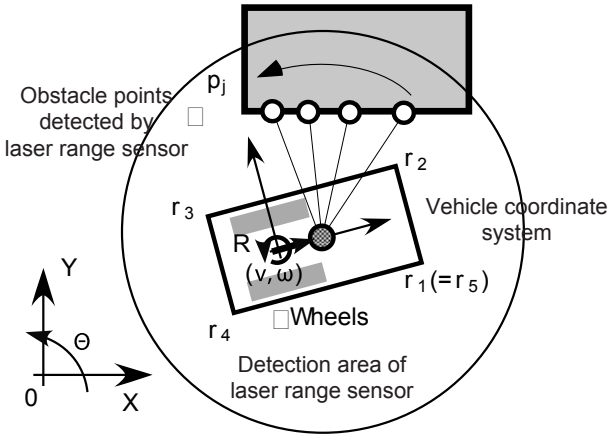


Fig. 5. Model of nonholonimic vehicle with a laser range sensor.

The obstacle avoidance problem to be solved is stated as follows.

1. We consider a nonholonomic vehicle with two independently driven wheels as shown in Fig. 5. It moves in a planar environment. The configuration of a vehicle is defined by $\boldsymbol{R} = (X, Y, \Theta)^T$ in the base coordinates, where (X, Y) is the position of the midpoint of

two wheels' axis and Θ is its orientation. The discrete kinematic model of this vehicle is written as

$$\begin{cases} X_n = X_{n-1} + v\Delta t \cos(\Theta_{n-1} + \frac{\omega \Delta t}{2}) \\ Y_n = Y_{n-1} + v\Delta t \sin(\Theta_{n-1} + \frac{\omega \Delta t}{2}) \\ \Theta_n = \Theta_{n-1} + \omega \Delta t \end{cases} \quad (1)$$

where Δt is the sampling time for control and $(v, \omega)^T$ are translational and rotational velocity. Suffix $n-1, n$ denote positions before and after the sampling time.

2. The shape of a vehicle is (or can be approximated by) a rectangle. Let the vertexes of the vehicle's shape be $\boldsymbol{r_i}(i = 1, 2, ..., n_r)$ in the vehicle coordinates.
3. A laser range sensor is mounted on the vehicle to detect obstacles. It has a circular detection area. Obstacles are scanned by this sensor every a certain angle. Let the detected points on the outline of obstacles be $\boldsymbol{p_j}(j = 1, 2, ..., n_p)$ in the vehicle coordinates. These points are called "obstacle points".
4. Global path planning is given. After the goal position of a vehicle $\boldsymbol{R_G} = (X_G, Y_G, \Theta_G)^T$ is given relatively near the start position, a local path to avoid obstacles is found. We explain the case that the start position is behind the goal position and a vehicle go forward to the goal. When a vehicle go backward to the goal, the front and back of the vehicle should be swapped.

3. Algorithm for Local Obstacle Avoidance

3.1 Outline

A method of local obstacle avoidance for a vehicle with two driven wheels and rectangular body is proposed. This outline is shown in Fig. 6. Basically, simple potential field is applied. Both an attractive force form the goal and repulsive forces from obstacles act on the vehicle and the resultant force moves the vehicle (Fig. 7). Main differences between the general method using potential field and our proposed method are following two points.

- In order to consider the motion constraint that a vehicle can't move just beside, two points of action where the attractive and repulsive forces act are placed on the front and rear body of a vehicle. Their forces at two points are treated as they work on a "lever" of which the fulcrum is the midpoint of two wheels.
- In order to consider the shape of vehicle's body, repulsive forces form obstacles are determined by the distances between obstacle points and the outline of vehicle's body.

This idea can simply introduce the consideration about the motion constraint and the vehicle's shape into the potential field method. Proposed method needs almost same computing power as general potential field method because their calculations have little difference. Since the data of a laser range sensor (obstacle points) can be used directly, this method is suitable for real-time obstacle avoidance. However, this also has a disadvantage of the local minima problem.

Then, our proposed method is explained in detail in the following sections. The generation of forces and the determination of vehicle's velocity are treated on the vehicle coordinate system.

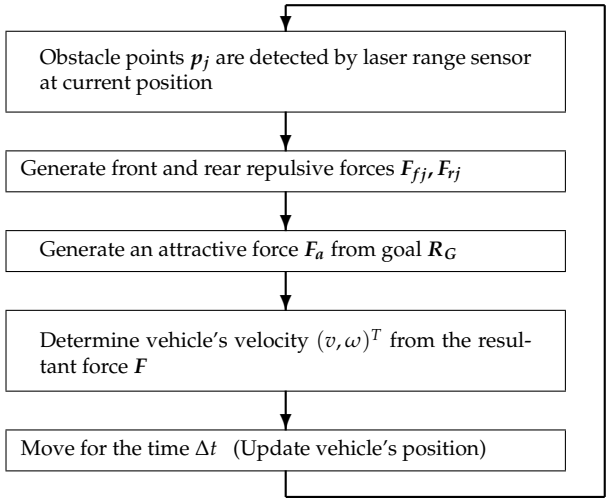


Fig. 6. Flowchart of proposed algorithm for obstacle avoidance.

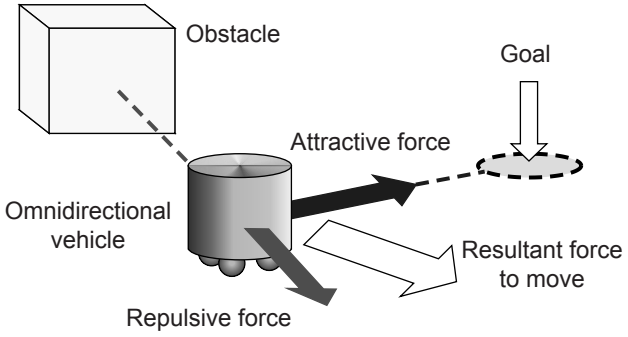


Fig. 7. Basic potential field method for omnidirectional vehicle to avoid obstacles

3.2 Generation of attractive and repulsive forces

Two action points of forces are placed at the front end and the rear end of a vehicle's body as shown in Fig. 8. Let the front end be $\boldsymbol{r_f} = (x_f, 0)^T$, and the rear end be $\boldsymbol{r_r} = (-x_r, 0)^T$ in the vehicle coordinate system. These points should not always be placed at the ends of a vehicle, however, acting forces at the ends makes vehicle's motion stable. When a obstacle point $\boldsymbol{p_j} = (p_{jx}, p_{jy})^T$ is detected in front of the line of two wheels' axes (y axis), a repulsive force $\boldsymbol{F_{fj}}$ is generated at the front point of action. When a obstacle point is behind this line, a repulsive force $\boldsymbol{F_{rj}}$ is generated at the rear point of action. The magnitudes of their forces are changed in inverse proportion to the squares of the distances between obstacle points and a

vehicle's body. Then, their forces are given by

$$\boldsymbol{F}_{fj} = \frac{K}{|\boldsymbol{q}_{fj} - \boldsymbol{p}_j|^2} \frac{\boldsymbol{r}_f - \boldsymbol{p}_j}{|\boldsymbol{r}_f - \boldsymbol{p}_j|}, \quad \text{if } p_{jx} > 0 \tag{2}$$

$$\boldsymbol{F}_{rj} = \frac{K}{|\boldsymbol{q}_{rj} - \boldsymbol{p}_j|^2} \frac{\boldsymbol{r}_r - \boldsymbol{p}_j}{|\boldsymbol{r}_r - \boldsymbol{p}_j|}, \quad \text{if } p_{jx} < 0 \tag{3}$$

where $\boldsymbol{q}_{fj}$, $\boldsymbol{q}_{rj}$ are the intersections of the vehicle's body and the segments between obstacle points and the action points $\boldsymbol{r}_f$, $\boldsymbol{r}_r$ respectively. K is the coefficient of repulsive force.

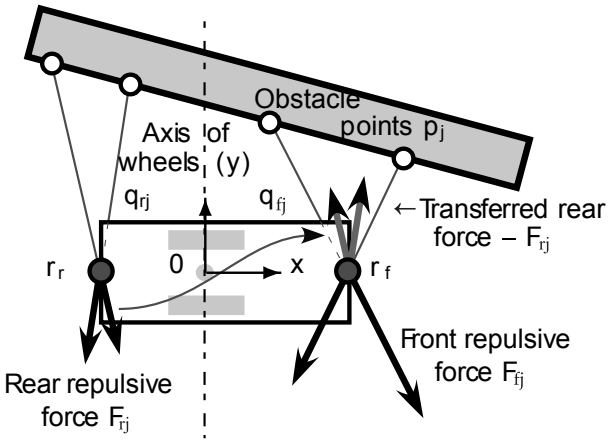


Fig. 8. Generation of repulsive forces from obstacle points.

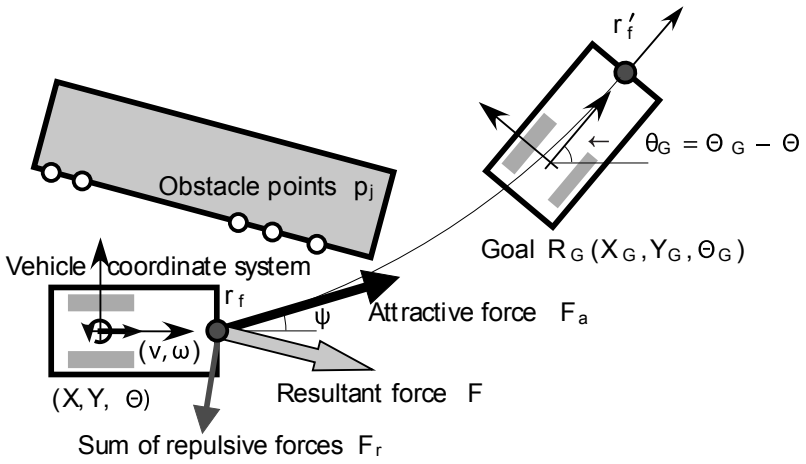


Fig. 9. Generation of attractive force and determination of velocity for avoidance.

Next, an attractive force $\boldsymbol{F_a}(|\boldsymbol{F_a}| = 1)$ pulls the front action point $\boldsymbol{r_f}$ of the vehicle toward the goal position as shown in Fig. 9. This attractive force is the tangential vector at the front action point $\boldsymbol{r_f}$ to the circle which comes in contact with the goal orientation of the front action point $\boldsymbol{r'_f}$. Without any obstacles, the vehicle moves on this circle and arrives at the goal position $\boldsymbol{R_G} = (X_G, Y_G, \Theta_G)^T$. The attractive force $\boldsymbol{F_a} = (\cos\psi, \sin\psi)^T$ is given by

$$\psi = 2\,\mathrm{atan2}\,(y'_G, x'_G) - \theta_G, \qquad \theta_G = \Theta_G - \Theta \tag{4}$$

$$\begin{bmatrix} x'_G \\ y'_G \end{bmatrix} = R(-\Theta) \begin{bmatrix} X_G - X \\ Y_G - Y \end{bmatrix} + R(\theta_G)\boldsymbol{r_f} - \boldsymbol{r_f} \tag{5}$$

where $R(\theta)$ is a rotation matrix by angle θ.

3.3 Resultant force and determination of vehicle's velocity

A resultant force $\boldsymbol{F}$ is obtained from the attractive and repulsive forces $\boldsymbol{F_a}$, $\boldsymbol{F_{fj}}$, $\boldsymbol{F_{rj}}$. Since the action points of their forces are not same, we can't simply add their force vectors. After the repulsive forces at the rear action point $\boldsymbol{F_{rj}}$ are transferred to the front action point by inverting their vectors $-\boldsymbol{F_{rj}}$, all force vectors are added at the front action point, because the front action point should be moved in the opposite direction of the rear repulsive force in order to move the rear body of the vehicle away from the rear obstacle point. That is, the front and rear action points have a relation like a "lever" of which the fulcrum is the midpoint of two wheels. Then, the resultant force at the front action point $\boldsymbol{F}$ is defined by

$$\boldsymbol{F} = \boldsymbol{F_a} + k_f \sum_{p_{jx}>0} \boldsymbol{F_{fj}} - k_r \sum_{p_{jx}<0} \boldsymbol{F_{rj}}, \qquad k_f + k_r = 1 \tag{6}$$

where the coefficients k_f, k_r represent the action rate of the front and rear repulsive forces. The determination of these coefficients are mentioned later.

Finally, the resultant force $\boldsymbol{F}$ pulls the front action point to move the vehicle. In other words, the translational and rotational velocities of the vehicle $(v, \omega)^T$ are determined in order that the front action point $\boldsymbol{r_f} = (x_f, 0)^T$ moves in the direction of the resultant force $\boldsymbol{F}/|\boldsymbol{F}| = (f_x, f_y)^T$.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = C \begin{bmatrix} f_x \\ \frac{f_y}{x_f} \end{bmatrix} \tag{7}$$

where C is the velocity coefficient. Since only the rate of translational and rotational velocities is obtained, suitable coefficient C should be given according to some limitations of velocity or acceleration. For example, when the maximum of the rotational velocity ω_{max} is specified, C becomes

$$C = \omega_{max} \frac{x_f}{f_y}, \qquad \text{if } |\omega| > \omega_{max} \tag{8}$$

3.4 Action rate of front and rear forces

How to determine the action rate of the front and rear repulsive forces k_f, k_r is discussed. When a vehicle avoids a block of obstacle as shown in Fig. 10, the force action rate doesn't affect vehicle's motion so much because repulsive forces mainly work at either action point. When a vehicle moves between walls on both sides, vehicle's motion isn't also sensitive to the action rate because repulsive forces at two points turn the vehicle in the same way. The case where the action rate affects vehicle's motion relatively is wall following. Repulsive forces are

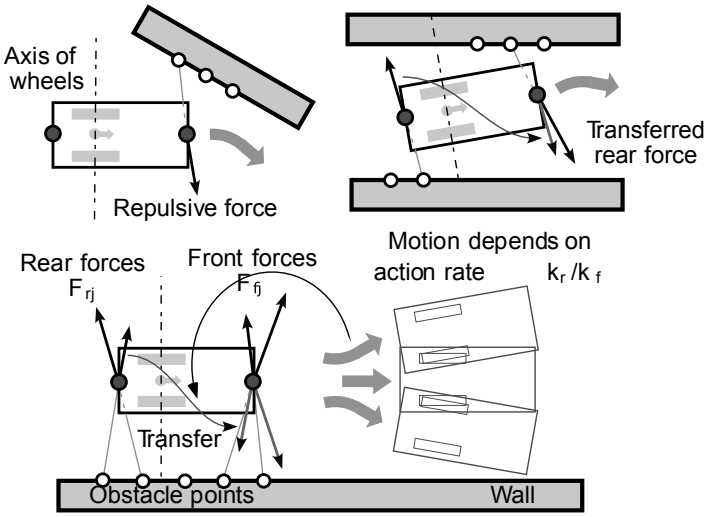


Fig. 10. Effect of action rate between front and rear forces on vehicle's motion. (Wall following is sensitive and others are not.)

generated at two action points to move their points away from the wall and their direction to turn the vehicle is different because they are treated like a lever of which the fulcrum is the midpoint of two wheels. During wall following, larger action rate of front forces k_f makes the vehicle turn away from the wall and larger action rate of rear forces k_r makes the vehicle turn close to the wall as shown in Fig. 10. Therefore, the force action rate should be determined so as that the vehicle goes straight along a wall, i.e. the resultant force vector $\boldsymbol{F}$ should be parallel to the wall without considering the attractive force $\boldsymbol{F_a}$. Let the components of repulsive force vectors at the front and rear action points in the vertical direction to the wall be F_{fyj}, F_{ryj} respectively, this condition becomes

$$k_f \sum F_{fyj} - k_r \sum F_{ryj} = 0 \tag{9}$$

Then, the action rate

$$\frac{k_r}{k_f} = \frac{\sum F_{fyj}}{\sum F_{ryj}} \tag{10}$$

is obtained. This depends on the shape of a vehicle, the detection area of a laser range sensor and so on.

The action rate for a rectangular vehicle is concretely calculated as shown in Fig. 11. Let the front length, rear length, width of a vehicle be a, b, $2c$, respectively. Let the distance between the wheels' axis and the laser range sensor be s_0 and the detection limit distance of the sensor be s. We can get the sum of components of repulsive force vectors in the vertical direction to the wall after repulsive forces, which are inversely proportional to the squares of the distances between the vehicle's body and the wall, are calculated. When the gap between a vehicle and a wall is d, they are given by

$$\sum F_{fyj} = \frac{KD^3}{d^2} I(a, -\phi_0, \phi_1) + KDI(a, \phi_1, \phi_3) \tag{11}$$

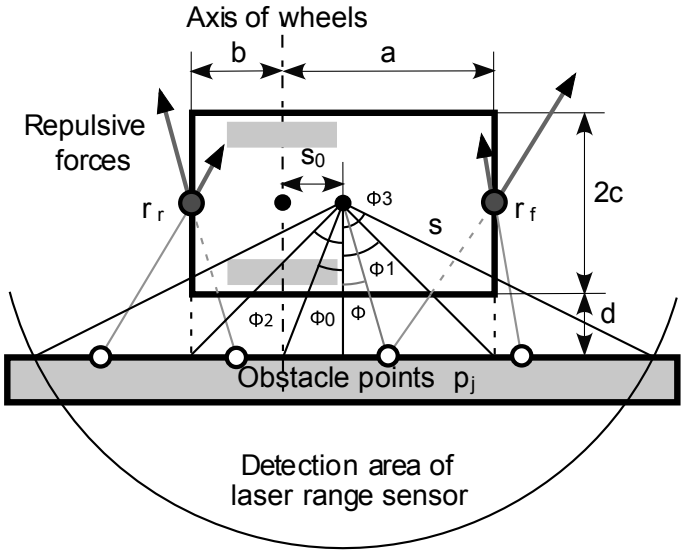


Fig. 11. Geometry of vehicle's body and repulsive forces during wall following.

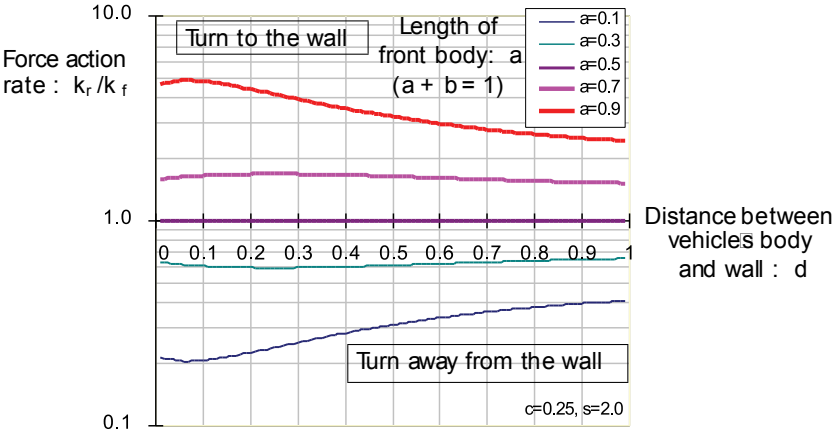


Fig. 12. Force action rate k_r/k_f to go straight along a wall.

$$\sum F_{ryj} = \frac{KD^3}{d^2} I(-b, -\phi_2, -\phi_0) + KDI(-b, -\phi_3, -\phi_2) \tag{12}$$

$$I(\alpha, \phi_s, \phi_e) = \int_{\phi_s}^{\phi_e} \frac{d\phi}{((\alpha - s_0 - D\tan\phi)^2 + D^2)^{\frac{3}{2}}}, \qquad D = c + d \tag{13}$$

where ϕ is the angle from the sensor to an obstacle point on the wall, ϕ_0, ϕ_1, ϕ_2, ϕ_3 are the angles from the sensor to the intersections between the wall and the wheels' axis, the front line of the body, the rear line of the body, the circle of detection limit of the sensor. Finally, the

action rate becomes

$$\frac{k_r}{k_f} = \frac{D^2 I(a, -\phi_0, \phi_1) + d^2 I(a, \phi_1, \phi_3)}{D^2 I(-b, -\phi_2, -\phi_0) + d^2 I(-b, -\phi_3, -\phi_2)} \tag{14}$$

This value is calculated by numerical integration.
Fig. 12 shows the relation between the distance from a wall d and the action rate k_r/k_f for a vehicle to go straight along the wall. In this calculation, it is assumed that the sensor is placed at the center of the vehicle ($s_0 = (a-b)/2$) and the length of the vehicle is normalized ($a + b = 1$). Some cases of the front and rear length of the vehicle with the width $2c = 0.5$ are shown in the graph. It can be seen that the action rate for the vehicle to go straight does not change so much according to the distance from the wall if the driven wheels are not close to either end of the body ($a = 0.3 \sim 0.7$). Even if the wheels are close to the end, there is no problem for obstacle avoidance because the action rate below these curves in the graph makes the vehicle turn away from the wall. When the front length a is short (Ex. $a = 0.1$), the minimum of the curve should be taken for the action rate. When a is long (Ex. $a = 0.9$), the value on the curve at a certain distance should be taken for the action rate because it makes the vehicle turn away from the wall if the vehicle goes inside its distance.

4. Simulation

Range of laser range sensor	$0 \sim 1$ [m]
Directional resolution of lager range sensor	1 [deg.]
Sampling time for control: Δt	0.1 [s]
Coefficient of repulsive force: K	0.004
Coefficient of velocity: C	0.2
Maximum angular velocity: ω_{max}	0.2 [rad/s]

Table 1. Standard parameters for simulation

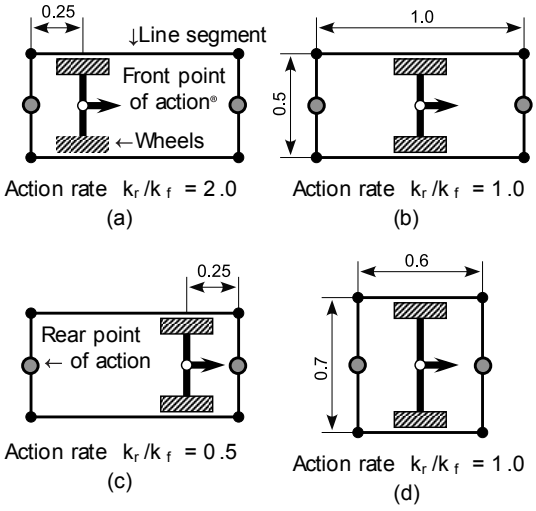


Fig. 13. Shape of vehicles for simulation

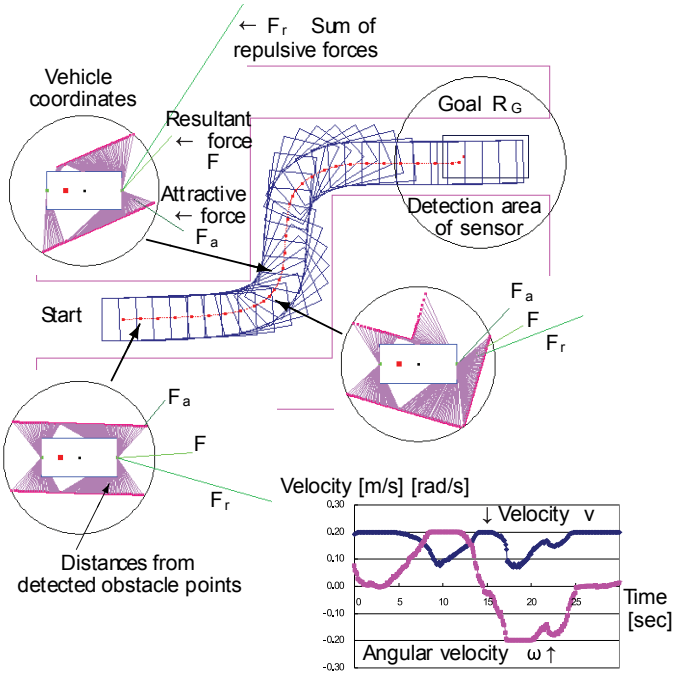


Fig. 14. Simulation result of the vehicle with shape (a).

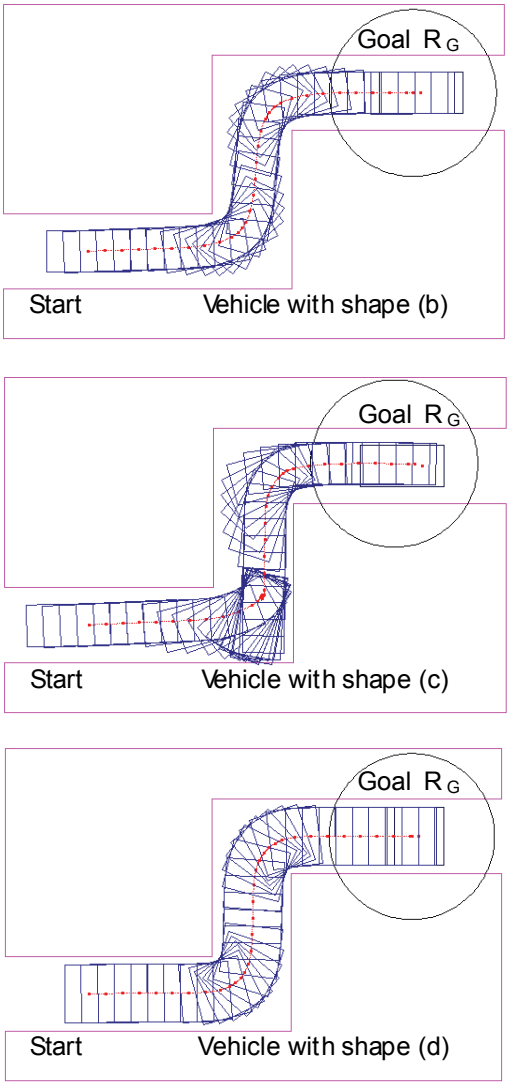


Fig. 15. Simulation results of vehicles with various shape.

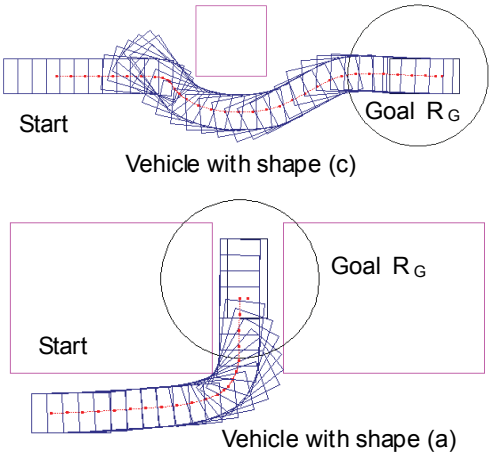


Fig. 16. Simulation results for various environment.

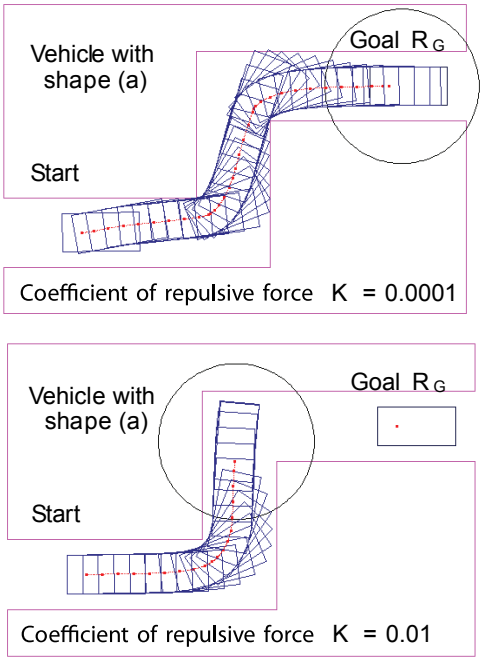


Fig. 17. Simulation results by using various coefficient of repulsive force.

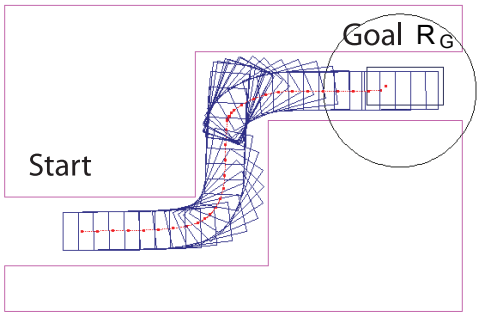


Fig. 18. Escape from local minimum by decreasing coefficient of repulsive force (When the vehicle stopped at Fig.17, coefficient of repulsive force K was temporarily decreased from 0.01 to 0.001 in 1 second.)

Our proposed method of local obstacle avoidance has been tested. All simulation programs were written in C language on PC Linux system. Table 1 shows standard parameters for the simulation. We assumed the following situation. A laser range sensor is mounted on the center of the rectangular body of a vehicle. Since the scan resolution angle is 1 degree, the max. number of detected obstacle points is 360. An obstacle point is calculated as the nearest intersection of obstacles and a direction of a laser range sensor within its detection area. Scan time is short enough to be neglected as compared with vehicle's speed. 4 types of vehicle's bodies were prepared as shown in Fig. 13. The action rate of the front and rear repulsive forces k_f, k_r was determined for each body by Equation (14).
Fig. 14 $\sim$ 17 are simulation results. Start and goal position were given as shown in each figure. Fig. 14 shows the generated path for the vehicle with shape (a) to pass through a narrow crank course. It can be seen that a smooth collision free path considering both rectangular body and motion constraint is generated by our proposed method. Obstacle points detected by the laser range sensor $\boldsymbol{p_j}$, distances between the vehicle's body and them, sum of repulsive forces $\boldsymbol{F_r}$, attractive forces $\boldsymbol{F_a}$ and resultant forces to avoid obstacles $\boldsymbol{F}$ are also shown in the vehicle coordinate system at some positions (See each circle in Fig. 14). A collision free direction can be determined from the sensor information directly. Moreover, the translational and rotational velocities of the vehicle v, ω are plotted in the graph and we can see that they changes smoothly.
Fig. 15 shows the cases of other vehicles' bodies and Fig. 16 shows the cases of other environments. It turns out that our proposed method is effective for various situations. Fig. 17 shows the results for various coefficient of repulsive force $K = 0.0001 \sim 0.01$. Larger coefficient generates the path farther away from obstacles, however, it isn't too sensitive (See also Fig. 14 of $K = 0.004$). When the coefficient is large, there seen some cases where the vehicle gets stuck at a local minimum. Many algorithms (Liu et al., 2000) to escape from the local minimum have been already proposed for general potential field method and some of them can be also applied to this case. For example, when a vehicle is stopped for a while like Fig. 17, it can escape from this local minimum by decreasing the coefficient of repulsive force K temporarily (Fig. 18).

5. Experiment

Navigation experiments were made by a powered wheelchair as shown in Fig. 19. This wheelchair has two powered wheels ("JW-I" manufactured by Yamaha Motor Co., Ltd., Wheel: 24[inch], Max. speed: 0.86[m/s]) with rotary encoders (2400[p/r]) and their velocities are controlled by PC (PI control every 0.05[sec]). Two laser range sensors ("URG-04LX" man-

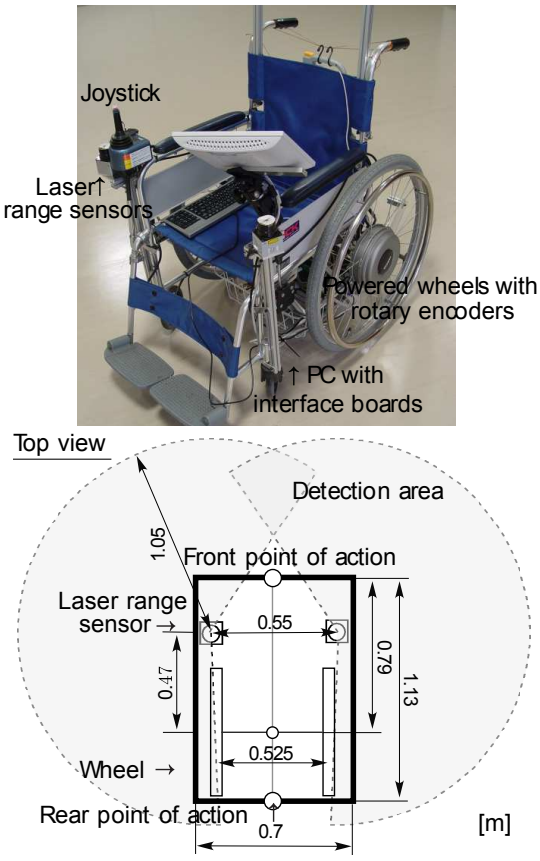


Fig. 19. Wheelchair setup and approximation to rectangular body to for experiment

ufactured by Hokuyo Automatic Co., Ltd., Range: 4[m], Resolution: ±10[mm], RS232C: 115.2[kbps]) are mounted at the both arm ends of the wheelchair not to disturb a user and not to be disturbed by a user. Their heights are 0.67[m] from the floor. After our proposed algorithm of obstacle avoidance was implemented to this wheelchair, navigation experiments were done in the environment as shown in Fig. 20. Table 2 shows parameters for the experiment. The shape of the wheelchair is approximated by a rectangle (Fig. 19), of which size is a little larger ($1 \sim 2$cm) than the real body.

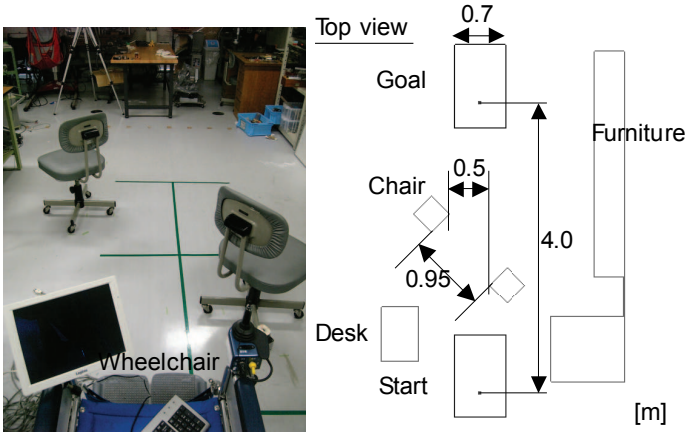


Fig. 20. Environment of navigation experiment

Range of laser range sensor	$0.2 \sim 1.05$ [m]
Directional resolution of lager range sensor	1.08 [deg.]
Sampling time for control: Δt	0.2 [s]
Coefficient of repulsive force: K	0.002
Coefficient of velocity: C	0.2
Maximum angular velocity: ω_{max}	0.2 [rad/s]
Action rate: k_r/k_f	0.6

Table 2. Parameters for navigation experiment

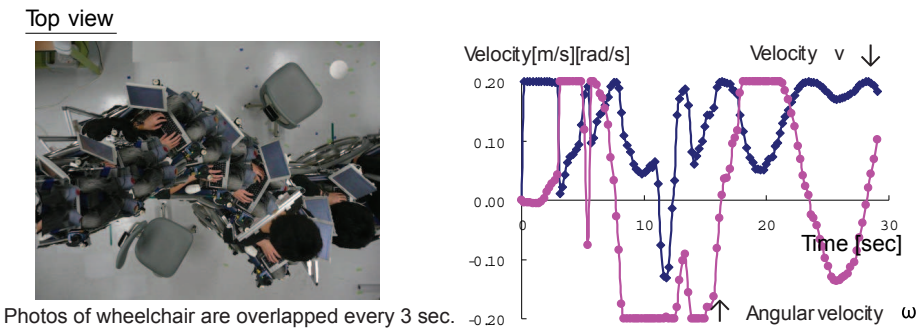


Fig. 21. Experimental results of trajectory and velocity

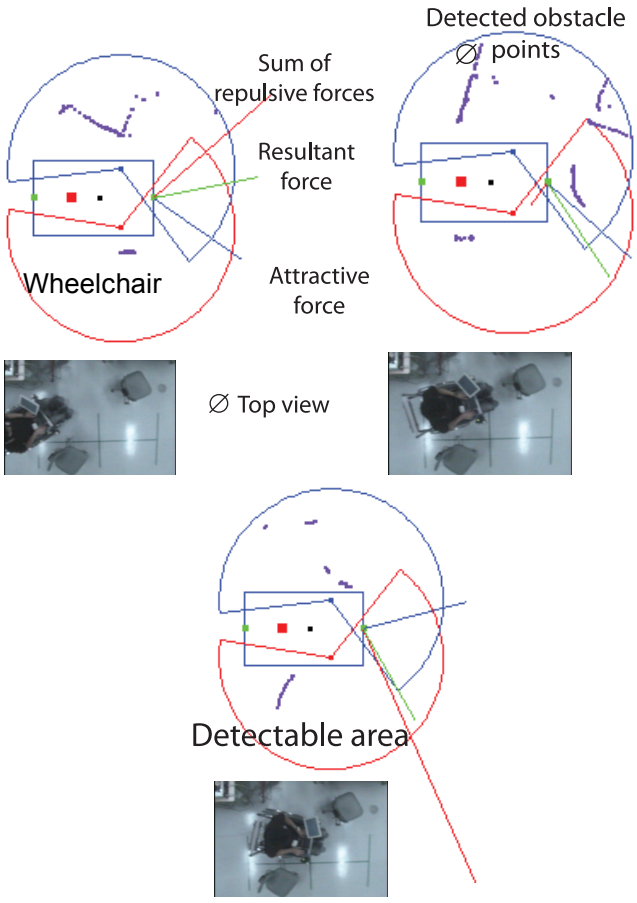


Fig. 22. Experimental results of sensor data at some places

Fig. 21 and Fig. 22 are experimental results. They shows the trajectory, velocity, and sensor data during the navigation. The autonomous wheelchair succeeded to avoid obstacles such as chairs, desks, and furniture and passed smoothly through the narrow space between chairs. The velocity data shows that the velocity was not always smooth in the experiment because the sensor sometimes failed to detect obstacle points. This failures can be seen in the sensor data at some places. One reason is that the sensor can't always catch the reflected laser light owing to the condition of obstacle surfaces. Another reason is that the shapes of obstacles changes according to the height of the sensor. It can be seen in Fig. 22 that the laser range sensor detected the back of a chair, not the seat of it, for example. 3D data of obstacles should be detected for practical use.

6. Application

An application of the obstacle avoidance function for an intelligent wheelchair is presented. It is an assist system of joystick operation to avoid obstacles for wheelchair users. In stead of giving a goal, the direction of the tilted joystick is assigned to the attractive force vector $\boldsymbol{F_a}$ in the proposed potential field method.

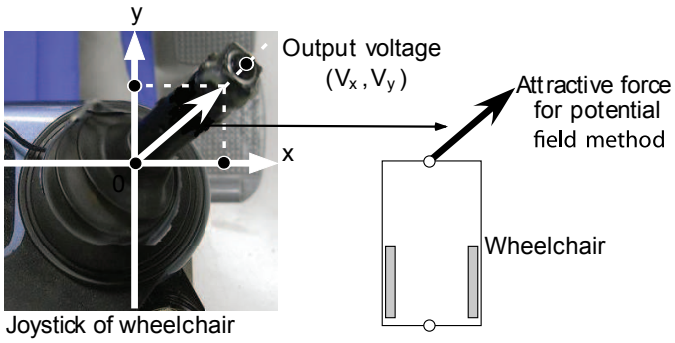


Fig. 23. Assist system of joystick operation to avoid obstacles

Let the 2D output voltages of the joystick device be $(V_x, V_y)^T$, the attractive force becomes

$$\boldsymbol{F_a} = \frac{\boldsymbol{V}}{|\boldsymbol{V}|}, \qquad \boldsymbol{V} = \left(\frac{V_x}{V_{xmax}}, \frac{V_y}{V_{ymax}} \right)^T \tag{15}$$

where $(V_{xmax}, V_{ymax})^T$ is the maximum of the output voltage. Then, the angle of the tilted joystick is assigned to the speed of the wheelchair $(v, \omega)^T$. Instead of Equation (7), the following equation is used.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = |\boldsymbol{V}| C \begin{bmatrix} f_x \\ \frac{f_y}{x_f} \end{bmatrix} \tag{16}$$

When there are no obstacles, the wheelchair moves as operated by the joystick. When wheelchair is going to collide with obstacles, the joystick operation is corrected by the potential field method. This system enables obstacle avoidance without precise operation of the wheelchair.

This assist system of joystick operation was tested in the same environment as the navigation experiment (Fig. 20). The user didn't operate the joystick precisely. Fig. 24 shows the trajectory of the wheelchair, the direction of the joystick, and the angle of the resultant force to move the wheelchair. It can be seen that the wheelchair succeeded to avoid chairs smoothly though the joystick operation by a user is rough. By this assistance of obstacle avoidance, a user can use the wheelchair easier with less joystick operation, even in the place where is difficult for he / her to pass through.

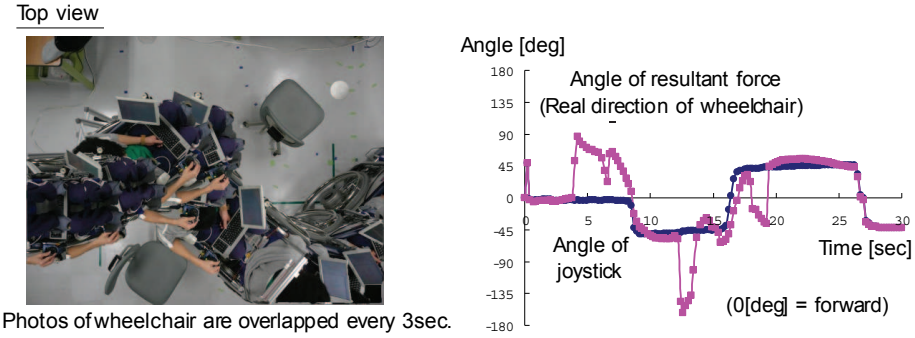


Fig. 24. Experimental results of assist system to avoid obstacles

7. Conclusion

In this chapter, a practical method of local obstacle avoidance for a nonholonomic vehicle with rectangular body has been proposed. Simple potential field directly using local sensor data is applied. Repulsive forces according to distances between obstacles and vehicle's body are generated at either front or rear point of action on the vehicle and their forces are treated like a lever. Both motion constraint and shape of a vehicle can be considered by this simple idea. Simulation results for various situations and experimental results by a wheelchair have proved effectiveness of our algorithm. Although this method has a disadvantage of local minima as well as general potential field method, it is intended for practical use because adequate path for local obstacle avoidance can be obtained with a little computing power. Furthermore, this algorithm may be applied to not only vehicles with two independently driven wheels but also car-like vehicles. Some improvements of the intelligent wheelchair such as 3D obstacle sensing and haptic joystick for obstacle avoidance, and consideration about general shape of vehicles are remained for our further works.

8. References

Kavraki, L. et al. (1996). Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces, *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 4, pp. 566-580

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-98

Kondak, K. & Hommel, G. (2001). Computation of Time Optimal Movements for Autonomous Parking of Non-Holonoimic Mobile Platforms, *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, pp. 2698-2703.

Latombe, J. C. (1991). Robot Motion Planning, *Kluwer Academic Publishers*,

Laumond, J. P. et al. (1994). A Motion Planners for Nonholonomic Robots, *IEEE Transaction on Robots and Automation*, Vol. 10, No. 5, pp. 577-592

Liu, C. et al. (2000). Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation, *Proceedings of 2000 IEEE International Conference on Robotics and Automation*, pp. 983-988

Minguez, J. et al. (2006). Abstracting Vehicle Shape and Kinematic Constraints from Obstacle Avoidance Methods, *Autonomous Robots*, Vol. 20, pp. 43-59

Ramirez, G. & Zeghloul, S. (2001). Collision-free Path Planning for Nonholonomic Mobile Robots Using a New Obstacle Representation in The Velocity Space, *Robotica*, Vol. 19, pp. 543-555

Rimon, E. & Koditschek, D. E. (1992). Exact Robot Navigation Using Artificial Potential Functions, *IEEE Transaction on Robotics and Automation*, Vol. 8, No. 5, pp. 501-518.

Schwartz, J. T. & Sharir, M. (1983). On the Piano Movers' Problem: I. The Case of a Two-dimensional Rigid Polygonal Body Moving amidst Polygonal Barriers, *Communications on Pure and Applied Mathematics*n, Vol. 36, pp. 345-398.

Strobel, M. (1999). Navigation in Partially Unknown, Narrow, Cluttered Space, *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, pp. 28-34

Ulrich, I. & Borenstein, J. (2000). VFH*: Local Obstacle Avoidance with Look-Ahead Verification, *Proceedings of 2000 IEEE International Conference on Robotics and Automation*, pp. 2505-2511

Developing FPGA-based Embedded Controllers using Matlab/Simulink

T. Barlas and M. Moallem
Mechatronics System Engineering
Simon Fraser University
Surrey, BC, Canada
Email (corresponding author): mmoallem@sfu.ca

Abstract
Field Programmable Gate Arrays (FPGAs) are emerging as suitable platforms for implementing embedded control systems. FPGAs offer advantages such as high performance and concurrent computing which makes them attractive in many embedded applications. As reconfigurable devices, they can be used to build the hardware and software components of an embedded system on a single chip. Traditional FPGA design flows and tools, requiring the use of Hardware Description Languages (HDLs), are in a different domain than standard control system design tools such as MATLAB/Simulink. This paper illustrates development of FPGA-based controllers by utilizing popular tools such as MATLAB/Simulink available for the design and development of control systems. The capability of DSP Builder is extended by developing a custom library of control system building blocks that facilitates rapid development of FPGA-based controllers in the familiar Matlab/Simulink environment. As a case study, this paper presents how the tools can be utilized to develop a FPGA-based controller for a laboratory scale air levitation system.
Keywords: Embedded controllers, control firmware/software design, computer aided design tools

1. Introduction

Embedded control systems are found in a wide range of applications such as consumer electronics, medical equipment, robotics, automotive products, and industrial processes (Chan, Moallem, & Wang, 2007). Embedded control systems typically use microprocessors, microcontrollers or Digital Signal Processors (DSPs) for their implementation. For such systems, control algorithms are implemented as software programs that execute on a fixed architecture hardware processor. The processor itself is connected to various peripherals such as memories, Analog to Digital converters, and other I/O devices. Alternatively, FPGAs are increasingly becoming popular as implementation platforms on which the control algorithms can be implemented by programming reconfigurable hardware logic resources of the device. FPGAs have characteristics that make them suitable for realizing

hardware implementations of algorithms and systems. They offer excellent features such as computational parallelism, reconfigurable customization, and rapid-prototyping (Chan, Moallem, & Wang, 2007; Tessier and W. Burleson, 2001). Recently, there has been a growing interest in developing FPGA-based control systems. Casalino, Giorgi, Turetta, & Caffaz (2003), and Oh, Kim, & Lim (2003) used an FPGA as part of an embedded solution for controlling the motion of a four-fingered robotic hand. Koutroulis, Dollas, & Kalaitzakis (2006) implemented a PWM generator on a FPGA that was capable of generating signals of frequencies up to 3.985 MHz with a duty cycle resolution of 1.56%. Tjondronugroho, Al-Anbuky, Round, & Duke (2004) and Jung, Chang, Jyang, Yeh, & Tzou (2002) compared DSP-based and FPGA-based implementations of a multi-loop control strategy to control single-phase inverters. The simulation capability of the system-level tool System Generator from Xilinx was exploited by Ricci & Le-Huy (2002) to build the computational engine for variable-speed drives using FPGAs. Ramos, Biel, Fossas, and Guinjoan implemented a fixed-frequency quasi-sliding control algorithm on an FPGA for control of a buck inverter. In this application, the high switching frequency (ranging from 20 to 40 kHz) required fast computation of the control law, effectively ruling out software-based implementations on microprocessors or DSPs. In addition to control algorithms, FPGAs can also be used to implement various other components of the control system. For example, Zhao, Kim, Larson, & Voyles (2005) used a FPGA to implement a control system-on-a-chip for a small-scale robot.

While microcontrollers and DSPs have had the advantage of lower device cost over FPGAs, the gap in cost is narrowing with advancing technology, making FPGAs more and more attractive devices. Moreover, FPGA-based implementations may reduce overall cost of a system since multiple components of the design can be implemented as a system-on-a-programmable-chip. In some cases, FPGA-based implementations may give higher levels of performance for a design as compared to other implementations. For such cases, FPGAs may be the only choice for implementation because microcontrollers and DSPs would not meet the performance requirements, and Application Specific Integrated Circuits (ASICs) may have too high development costs.

2. Control System Design Tools and FPGA Design Flows

With FPGAs emerging as viable platforms for controller implementation, there exists a gap between what typical control engineers are used to in a control system design tool and what is required from existing FPGA tools and flows. Traditional FPGA design flows often require the use of Hardware Description Languages such as Verilog or VHDL for development. However, hardware description languages are in a low-level domain of bits, registers and logic functions as opposed to the high-level domain of signals, variables and mathematical functions. Nonetheless, high-level development tools for FPGAs are emerging to reduce this gap. Some of these tools are based on existing design environments such as Matlab/Simulink. In this work, a tool designed to work in Simulink, called DSP Builder, is studied as a development tool for FPGA-based controllers. DSP Builder allows the familiar and easy-to-use design environment of Simulink to be used for development for FPGAs. In this way, the gap between the tools used by control engineers and FPGA development environments is narrowed. Hence, FPGA platforms can utilized to build controllers alongside microcontrollers and DSPs.

3. Development of Custom Control Library for FPGAs

The mathematical frameworks for designing DSP systems and digital control systems are similar. This section looks at the DSP Builder tool in more detail and discusses its applicability as a high-level development tool for FPGA-based controllers. The capability of DSP Builder for developing controllers is further extended by developing a Custom Control Library in Simulink. The custom library provides DSP Builder-based blocks that can be used to rapidly develop FPGA-based controllers. Development using DSP Builder requires both Simulink and Quartus II software packages. The Simulink design flow is a block diagram based design flow where predefined blocks, grouped in different libraries such as Math Operations, Logic and Bit Operations, and Continuous/Discrete libraries, can be dragged and dropped into a canvas model file. The blocks are then appropriately interconnected with virtual wires that propagate signals amongst the blocks. Using this method, any kind of static or dynamic system in various domains such as signal processing, control system, imaging, and communications can be specified. The entire system can then be simulated by using various numerical solvers to determine the time domain response of the system or the behavior of internal signals.

The DSP Builder provides special libraries of blocks for use in Simulink that are directly synthesizable into hardware logic for Altera FPGA devices. FPGA-implementable algorithms and systems can be developed by simply dragging and dropping DSP Builder Library blocks into Simulink model file and making desired connections between them. Each DSP Builder block has a direct HDL representation (in either Verilog HDL or VHDL) of the function it performs, i.e., the blocks encapsulate HDL modules. A special block, called SignalCompiler, when invoked, reads the Simulink model file and translates each of the DSP Builder blocks and their interconnections to corresponding HDL representations. Each of the DSP Builder blocks can have their parameters specified via dialog boxes in Simulink, and these parameter choices are propagated into their HDL representations. Finally, SignalCompiler combines the whole design into one HDL top level entity that can be processed through the FPGA design flow stages using Quartus II. The HDL entity will be functionally equivalent to the system in the Simulink model file when it executes on the FPGA (DSP Builder User Guide, 2005).

DSP Builder library blocks also come with bit- and cycle-accurate simulation models that can be invoked by the Simulink discrete-time numerical solvers to perform design simulation within the Simulink environment. Furthermore, for simulation purposes only, existing Simulink blocks (such as input sources) can be interconnected with DSP Builder blocks to perform richer simulations involving real-world subsystems that interact with the FPGA. This has an obvious advantage for embedded control system design in the sense that the expected behaviour of the DSP Builder-based controller can be simulated with a Simulink based plant model. Thus the response of the controller can be conveniently simulated and analyzed. This simulation uses Simulink's numerical solvers and is different from HDL-level simulation using RTL simulators. Nonetheless, because of the bit- and cycle-accuracy, the behaviour of the DSP Builder-based design in Simulink will match the behaviour of the generated HDL system when it executes on the FPGA. Thus the design can be simulated, analyzed, and then modified at a higher level of abstraction than at HDL-level. This process eliminates the need to go through FPGA design flow at each and every design iteration, resulting in reduced development time.

There are five main categories of blocks provided by DSP Builder: Arithmetic blocks, Logical operation and flow control blocks (in the Gate & Control library), Input, Output ports and bus manipulation blocks (in the IO & Bus library), Clock domain and sampling time blocks, and Blocks for storing signals and data (in the Storage library). Given a mathematical description of a controller (particularly in canonical form), the controller can be implemented by using three operators: multipliers, adders and delay elements. Hence any controller can be easily implemented in DSP Builder and realized as hardware on an FPGA. While the Multiplier, Adder and Delay blocks may be sufficient, development using these blocks only may be tedious and time-consuming. A major contribution of this work is to provide more sophisticated control system building blocks by developing what is referred to as the Custom Control Library in this paper.

3.1 Custom Control Library

DSP Builder library blocks offer basic functionalities that can be used to develop custom blocks for performing more complex functions. They can be utilized to develop a custom library of control system building blocks for quick implementation of FPGA-based embedded controllers.

An example of implementing a custom control library component is shown in Figure 1 where the "Parallel Adder" block adds the input *e(k)* to the previous input *e(k-1)*, which is produced from the "input_Delay" block, and then multiplied by *Ts*/2 (=*T*/2). The output of the "Multiplier" block is added to the previous value of the output, which is stored and made available from the "output_Delay" block. Because the current output is fed back to the "output_Delay" block, the bit width of the signal into the block is indeterminate, and so needs to be explicitly defined. This can be achieved by the "AltBus" block in Figure 1. The bit width for "AltBus" is automatically chosen to be the same as the bit width of the result of the multiplier. The bit width for the output signal "output" is also chosen to be the same as that of the multiplier result.

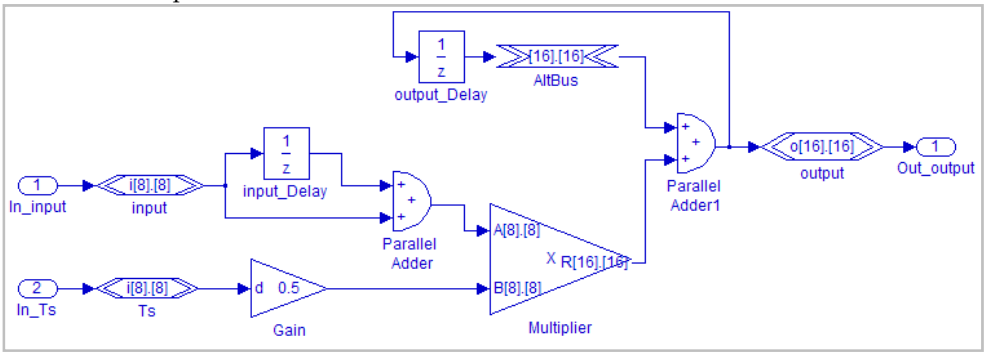


Fig. 1. Implementation of Custom Control Library Integrator (Trapezoidal) block using DSP Builder blocks.

Widely used transfer function blocks such as lead, lag, or PID, can be implemented in the Custom Control Library. For example, Figure 2 shows a discretized PID block as it appears in Simuink. The implementation uses the Integrator block (Trapezoidal implementation) and the Discrete Derivative block from the Custom Control Library. The block has inputs for

specifying the gains (input ports labeled “Kp”, “Kd” and “Ki”), the sample rate (port “Ts”), the sampling frequency (port “fs”) and the input to the block (labeled “input”). The “output” port is the control signal computed by the PID algorithm. The bit widths for all the ports can be specified via the block’s dialog box. Hardware implementation details such as the number of bits can also be specified from the dialog box.

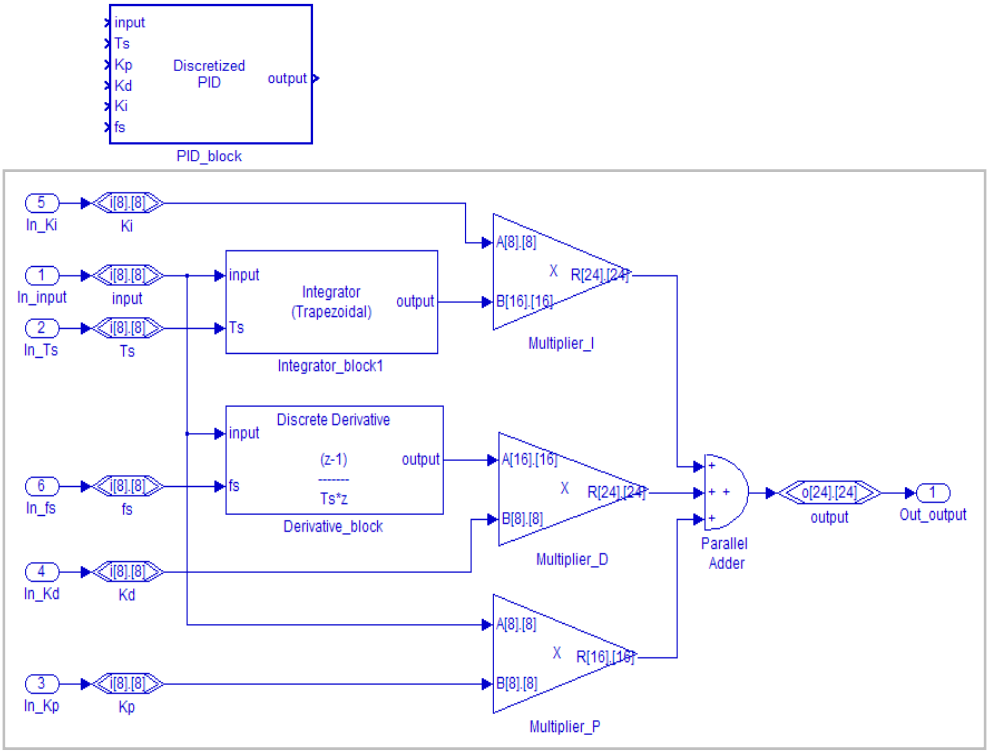


Fig. 2. Discretized PID controller.

3.1.1 Pulse Width Modulation Generator Block

Pulse Width Modulation (PWM) is frequently used in digital control systems, especially in DC motor drives. A digital PWM Generator is implemented for the Custom Control Library. Figure 3 shows the block as it appears in Simulink. In Figure 3, the counter counts up to a value of Clock Frequency divided by the PWM frequency. This value is set as the modulo of the counter so that the counter resets once it has counted up that value. While the count value is less than a value representing the current duty cycle in terms of the number of clock cycles, the output is logical high. Otherwise, the output is logical low. The number of clock cycles representing the current duty cycle is calculated by multiplying the duty cycle input in percentage by a value which, at 100% duty cycle would give the required clock cycles to output a logical high for the entire duration of the pulse period.

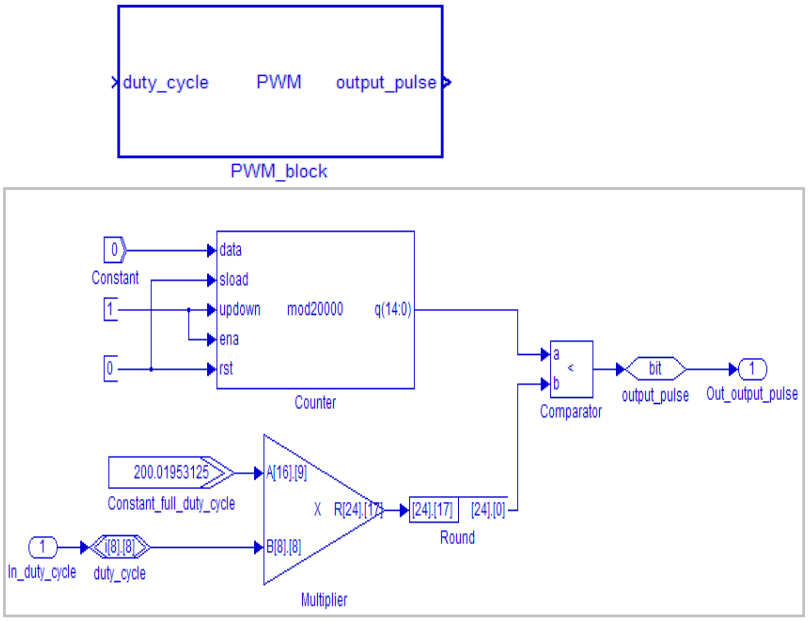


Fig. 3. Implementation of Custom Control Library PWM Generator block using DSP Builder blocks.

3.1.2 Analog to Digital Converter Controller Block

Analog to Digital (A/D or A2D) converters are necessary in digital control systems. There are various types of A/D converters based on different implementation technologies. Nonetheless, the converted data is transmitted to digital devices either as a parallel word over a digital bus or as a serial bit-stream over a single digital line.

Serial A/D converters are cheaper and simpler (in terms of circuit board placement) and therefore more commonly used. Hence, a block was implemented for the Custom Control Library that can be used to interface a FPGA with a serial A/D converter. When connected to a serial A/D converter, the master device (microcontroller, or, in this case FPGA) pulls the CS signal low to create a falling edge, which indicates to the A/D converter to initiate the conversion process. The A/D converter starts the conversion process, and after a brief waiting period, outputs the data word on the DOUT line in the form of a bit-stream starting with the most significant bit and ending with the least significant bit. The data transfer is usually synchronized to the falling edges of the CLK input from the master device. Then there is a wait period in which the DOUT line goes to a high impedance state. After the wait period, the CS signal has to go high again so that the next conversion can be initiated. The rate of data conversion is determined by the CLK clock signal. The goal of the custom ADC interface was to design a parameterizable block that can be used to interface FPGAs with a variety of SPI serial A/D converters.

Figure 4 shows the block as it appears in Simulink. The block has three inputs for configuring the A/D converter's registers ("In_initialize", "In_range_spec" and "In_control_spec"); an input for starting the conversion ("In_acquire"); a reset input; and an

input for the serial converted data that comes from the A/D converter. The block outputs two signals that are inputs to the A/D converter: the "Out_Cs_n" signal is CS; "Out_DIN" is DIN. The third output of the block, "Out_data_out" is the converted data outputted as a parallel word. The input at "In_range_spec" is the binary data that needs to be loaded onto the A/D converters range register (for those converters that have programmable analog input ranges). Similarly, the input at "In_control_spec" is the binary data for the A/D converters control register (which is used to configure the A/D converters if it can operate in multiple modes).

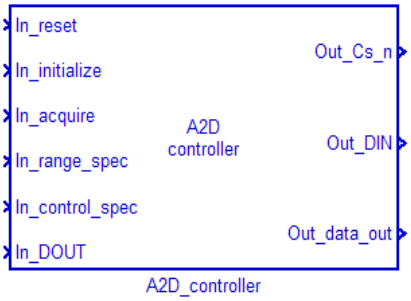


Fig. 4. A/D Controller block of the Custom Control Library.

In order to implement the interfacing process, the state machine shown in Figure 5 was used. The outputs at each state are also given. The converted word in parallel form (at the "Out_data_out" port) is available once the system moves out of the Acquire state. The internal implementation is achieved by using two State Machine blocks, two Parallel to Serial blocks to convert the range and control data into bit-streams for outputting on the "Out_DIN" port, and a Shift Taps block is used to convert the incoming serial data on "In_DOUT" into a parallel form.

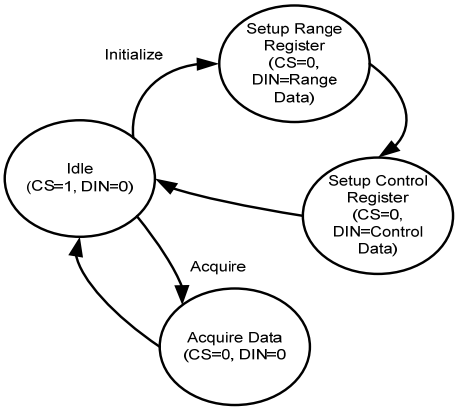


Fig. 5. State machine that implements SPI interface for A/D Controller block of the Custom Control Library.

3.2 Resource Usage of Custom Control Library Blocks

Logic Elements (LEs) and embedded multipliers are the two main resources used frequently in different designs. All FPGAs contain LEs but some devices may not contain embedded multipliers. For such devices, the multiplication operation has to be implemented by a group of LEs. However, multiplication is a resource intensive operation in terms of LE usage; hence FPGAs with embedded multipliers are preferred for control systems because the LEs can be freed up for more efficient use by other tasks. The results reported here are for individual implementation of the blocks on the Stratix EP1S80 FPGA chip. Each block was configured to use its default parameters. The Stratix family of chips contain DSP blocks that can be used to perform multiply, multiply-add, and multiply-accumulate operations found commonly in DSP systems. Each DSP block can implement 9×9 fixed-point multiplication with each input 9 bits wide and the output 18 bits wide, for any fixed-point bus format (unsigned integer, signed integer or signed fractional). For inputs greater than 9 bits, two or more DSP blocks configured as 18×18 multipliers or as 36×36 multipliers are used. Each 18×18 multiplier uses two 9×9 multipliers and each 36×36 multiplier uses eight 9×9 multipliers.

The results indicate that a very small number of LEs are used by the custom blocks when embedded multipliers are used. However, when embedded multipliers are not used, the LEs usage increases dramatically. This illustrates the resource intensiveness of the multiplication operation when implemented by LEs. Given a FPGA chip, a designer may decide to implement some blocks with embedded multipliers and some blocks without embedded multipliers, in order to fit the design into the FPGA chip. For example, the designer may implement the discretized PID block using embedded multipliers, because otherwise it would consume too many LEs (3036 LEs as compared to 205 LEs). However, the designer may implement the PWM Generator block without using embedded multipliers, thereby saving 8 embedded multipliers but only increasing the LE requirement from 40 LEs to 239 LEs.

4. Implementation Case Study

In order to demonstrate usage of the Custom Control Library to develop FPGA-based embedded controllers in the Simulink design environment, a control system was designed and developed for controlling the position of a levitating ball. This chapter describes the air levitation apparatus and the implementation of the control system. It is shown that all the necessary components of the control system are developed using system-level tools and then implemented on a single FPGA chip.

The air levitation apparatus consists of a transparent hollow tube, 43.5 cm in length, held vertically by a base which consists of a fan that blows air upwards into the tube. A table tennis ball is placed inside the hollow tube and has one degree of freedom to move either up or down. The thrust from the air flow causes the ball to be pushed upwards against the downward gravitational pull. The position of the ball is measured by an Infrared (IR) sensor placed at the top of the tube (the sensor does not block the air flow). There is an onboard serial A/D converter that can be used to convert the sensor voltage to a digital word; there are four connectors on the apparatus that enable an external digital device to interface with the converter. Figure 6 shows a block diagram of the apparatus with a picture of the system given in Figure 7.

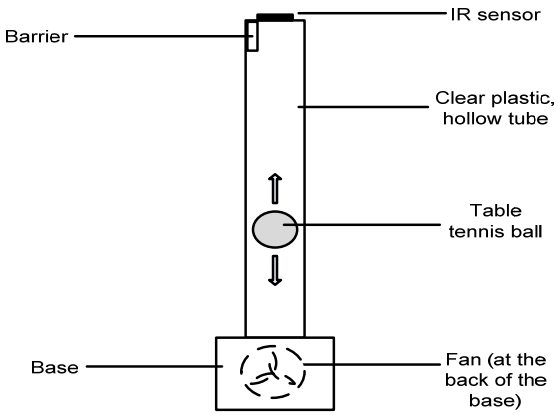


Fig. 6. Block diagram of the Air Levitation apparatus.

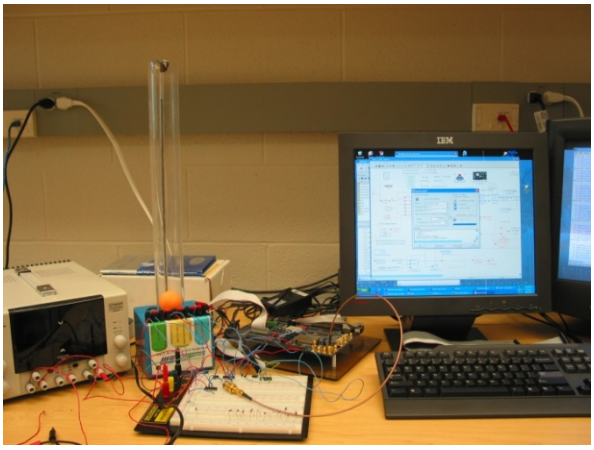


Fig. 7. Picture of the FPGA-based system.

The IR sensor is a Sharp GP2D120 Optoelectronic distance measuring sensor that outputs an analog voltage. When the ball is at the highest position (5.5 cm from the sensor), the sensor output voltage is 2.5 Volts. When the ball is at the lowest position (43.5 cm from the sensor or 38 cm from the barrier), the sensor voltage is 0.2 Volts.

The onboard A/D converter is a 12-bit, 250 kilo-samples-per-second, 2-channel serial converter with a 4-wire SPI compatible interface. It is capable of converting voltage from 0 to 5 Volts; thus it has a resolution of 1.2207 mV ($5V/2^{12bits}$). The four interface pins are: an external clock input for synchronizing the serial data transfer; a data input pin for shifting-in the A/D configuration 2-bit data word into the chip; a data-out pin for shifting-out the A/D converted result; and a convert input used to signal the device to start the conversion. The rate of conversion depends on the clock frequency: it takes 12 clock cycles plus the conversion time (less than 3.2 microseconds) for the data to be available.

The onboard fan uses a power supply of 8 Volts but its speed is controlled by a Pulse Width Modulated (PWM) signal that ranges from 0 to 5 Volts. The fan is constrained to rotate in one direction; therefore it cannot accept negative voltage at its PWM input.
The objective of the control system is to stabilize the position of the ball within the tube at any level. Figure 8 shows the block diagram of a closed-loop control system architecture. The input to the controller is the error signal between the desired voltage (representative of the desired position of the ball) and the measured voltage (representative of the actual position of the ball). The output of the controller is the control signal; which needs to be converted to a PWM signal to control the speed of the fan such that the upward thrust on the ball tries to negate the downward pull due to gravity and keep the ball level at a desired position.

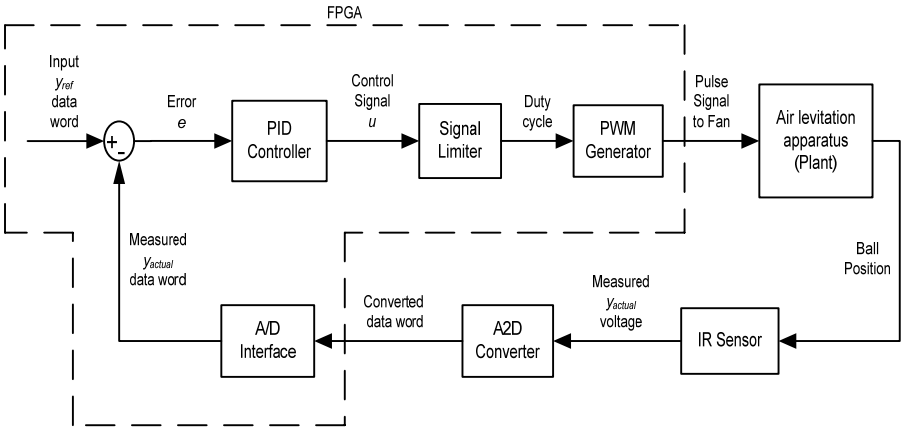


Fig. 8. Control system architecture used for air levitation controller.

The control system components were implemented on a single FPGA chip. The FPGA interfaces with the A/D converter on the apparatus to control its data conversion and acquisition. The 12-bit sampled data word acquired from the A/D converter is multiplied by the resolution of the A/D converter and converted into a fixed-point signed fractional number representing the real value of the sensor's voltage. The reference input y_{ref}, error e, control signal u, and all internal signals are represented in fixed-point signed fractional number format. y_{ref} is the desired value of the sensor voltage. The computed error between y_{ref} and y_{actual} is fed into a discretized PID controller.

4.1 Controller Development on FPGA Board

The air levitation control system was implemented on an Altera FPGA development board. The board was part of the Altera DSP Development Kit Professional Edition, which had the Altera Stratix EP1S80 FPGA chip on it. The Stratix chip had 79,040 Logic Elements, 7,427,520 total RAM bits, 176 9×9 embedded multipliers, 12 PLLs, and 679 I/O pins. The development board had an onboard 80 MHz oscillator that was used as the FPGA's main clock source. The Stratix EP1S80 FGPA is a high-end and expensive chip that is meant for development of

large and complex DSP algorithms and systems. However, other FPGAs such as Cyclone III that are low-end, inexpensive and with fewer number of device resources.
The discretized PID, PWM and A/D controller blocks of the Custom Control Library are all used for the corresponding components of the control system in Figure 8. The system is developed in Simulink by dragging and dropping the custom blocks and DSP Builder library blocks onto a model file, configuring their parameters and making connections between the blocks. Figure 9 shows the Simulink model file of the developed system. The model file was translated into VHDL using the SignalCompiler block of the DSP Builder library and a Quartus II project file was generated. The Quartus II software was subsequently used for the FPGA design flow steps of Analysis, Synthesis, Placement & Routing, and Programming the design onto the FPGA chip.

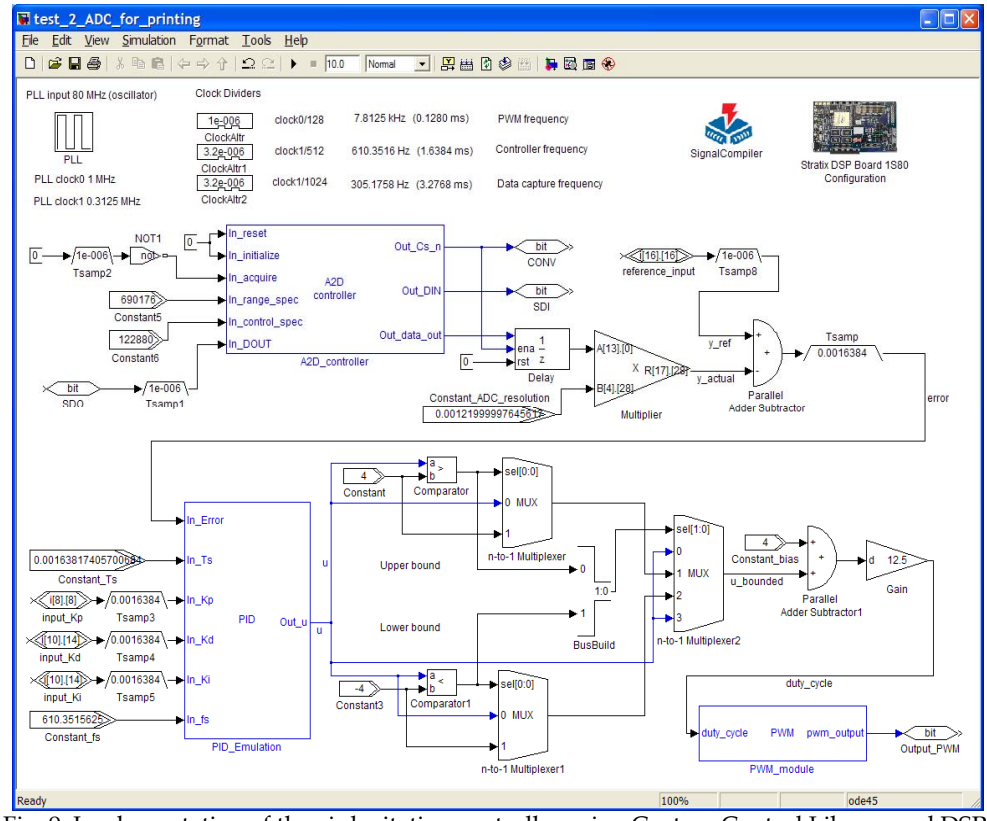


Fig. 9. Implementation of the air levitation controller using Custom Control Library and DSP Builder blocks.

The A/D Controller block is configured to operate at a clock frequency of 1 MHz (which is generated by a PLL block that divides the 80 MHz clock signal from the oscillator on the development board). It takes 21 clock cycles for the entire conversion process (the conversion time plus the serial data read time); hence the rate of sampled data acquisition is

47.619 kHz. The 12-bit sampled data is in unsigned integer format. It is converted to 13 bits signed fractional format but with zero bits for the fractional part (the most significant bit is the sign bit).
The discretized PID block is configured to operate at a frequency of 610 Hz resulting in a controller period of 1.6 milliseconds. The frequency 610 Hz is generated by a clock divider. The error signal into the PID block is represented using 16 bits for the integer part and 16 bits as for the fractional part (to allow for sufficient range and precision). The control signal output of the PID block is represented using 24 bits for the fractional part and 24 bits for the integer part allowing for sufficient range and precision. The control signal is then bounded and shifted into a positive value and then converted into a duty cycle. The duty cycle value, represented using 8 bits for the fractional part and 8 bits for the integer part, is inputted to the PWM block, which generates a single logical pulse whose width corresponds to the duty cycle. The PWM module uses a clock frequency of 1 MHz and the pulse frequency is 7.8125 kHz. This allows the duty cycle signal to have a resolution of 0.78%. The frequency 7.8125 kHz is also generated by a clock divider.

4.2 System Implementation Results

The control system was synthesized, placed & routed, and programmed into the Stratix EP1S80 FPGA. The FPGA-based controller was connected to the air levitation apparatus. Three sets of tests were carried out to evaluate the performance of the control system. The first set investigated the step response, the second set investigated the steady state response, and the third set of tests investigated the response to an external disturbance. A disturbance was created by manually holding a hand at the top of the tube to obstruct the air flow. All three tests used the following gains for the discretized PID controller: $K_P = 2$, $K_I = 0.038$ and $K_D = 3$. Figure 10 shows the response of the system once the ball has stabilized at a desired reference value of 0.6V, which represents the mid-point of the effective length of the tube (19 cm from the bottom).

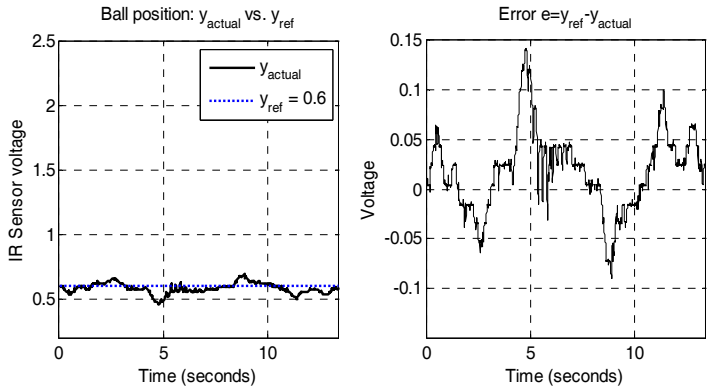


Fig. 10. Steady state response of air levitation controller to a reference input of 0.6V.

4.3 Reduced-size Controller

Large bit widths were used earlier in the development in order to capture signals at higher precisions for purposes of data plotting and analysis. However, such large bit widths were not required for the custom controller to operate correctly. Hence, in order to reduce the embedded multiplier usage, the custom controller of Figure 9 was modified to use smaller bit widths for its signals. The smaller bit widths take into account the possible range of the signals as the controller operates. For instance, the reference input bit width was changed from [16].[16] to [3].[13] because 3 bits for the integer part were sufficient to represent references from 0.2V to 2.5V, and 13 bits for the fractional part were sufficient to represent the reference in increments of $2^{-13}\mu V$. In addition to changes to the bit widths, certain multiplier elements in the custom controller were purposely configured to be implemented as Logic Elements instead of dedicated embedded multipliers. All changes were made in Simulink using the dialog boxes of the blocks, and were propagated to their HDL representations automatically.

The reduced custom controller required 1875 LEs and 14 9×9 dedicated embedded multipliers, as compared to 955 LEs and 70 9×9 embedded multipliers required by the original custom controller. This represents a 96% increase in LE usage, but a 400% decrease in 9×9 embedded multiplier usage. Consequently, the reduced custom controller can fit into a smaller and less expensive FPGA, such as the Cyclone II EP2C5, which contains 4,608 LEs and 26 9×9 embedded multipliers. Thus, the reduction in bit widths and tradeoffs between LE usage and embedded multiplier usage was worthwhile in this case. In conclusion, the reduced controller had more favorable resource usage but equivalent performance as the original controller.

5. Conclusions

The use of the system-level tool DSP Builder for high-level development of FPGA-based controllers was studied. The capabilities of the DSP Builder tool were further extended by developing the Custom Control Library. The custom library is comprised of widely used components such as discretized integrators, PID controller, PWM generator, and A/D controller. DSP Builder and the Custom Control Library together can be used to rapidly develop controllers in the familiar and standard Simulink design environment for FPGA implementation. An implementation case study demonstrated usage of DSP Builder and the Custom Control Library to develop a FPGA-based controller for an air levitation system in the Matlab/Simulink environment.

6. Acknowledgement

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Canadian Microlelectronics Corporation (CMC), Kingston, ON, Canada.

7. References

Altera Corporation (2005). DSP Builder User Guide, Version 5.1.0.

Casalino, G., Giorgi, F., Turetta, A., & Caffaz, A. (2003). Embedded FPGA-based control of a multifingered robotic hand, *IEEE Int. Conf. on Robotics & Auto.m,* pp. 2786-2791.

Chan, Y.F., Moallem, M. & Wang, W. (2007). Design and implementation of modular FPGA-based PID controllers, *IEEE Transactions on Industrial Electronics,* 54(4), pp. 1898-1906.

Jung, S-L., Chang, M-Y., Jyang, J-Y., Yeh, L-C., & Tzou, Y-Y. (1999). Design and implementation of an FPGA-based control IC for AC-voltage regulation, *IEEE Transactions on Power Electronics,* 14(3), pp. 522-532.

Koutroulis, E., Dollas, A., & Kalaitzakis, K. (2006). High-frequency pulse width modulation implementation using FPGA and CPLD ICs," *Journal of Systems Architecture,* 52(6), pp. 332-344.

Oh, S-N., Kim, K-I., & Lim, S. (2003). Motion control of biped robots using a single-chip drive", *IEEE Int.l Conf. on Robotics & Autom.*, pp. 2461-2465.

Tessier, R., & Burleson, W. (2001). Reconfigurable computing for digital signal processing: A Survey, *The Journal of VLSI Signal Processing*, 28(1), pp. 7-27.

Tjondronugroho, A., Al-Anbuky, A., Round, S., & Duke, R. (2004). Evaluation of DSP and FPGA based digital controllers for a single-phase PWM inverter," in *Australasian Universities Power Engineering Conference (AUPEC 2004)*, Sept. 2004. [Online]. Available: http://www.itee.uq.edu.au/~aupec/aupec04/papers/PaperID56.pdf. [Accessed: May 26, 2007].

Ramos, R.R., Biel, D., Fossas, E., and Guinjoan, F. (2003). A fixed-frequency quasi-sliding control algorithm: application to power inverters design by means of FPGA implementation, *IEEE Transactions on Power Electronics*, 18(1), pp. 344-355.

Ricci, F., & Le-Huy, H. (2002). An FPGA-based rapid prototyping platform for variable-speed drives, *28th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1156-1161.

Zhao, W., Kim, B.H., Larson, A.C., & Voyles, R.M. (2005). FPGA implementation of closed-loop control system for a small-scale robot," *12th International Conference on Advanced Robotics*, pp. 70-77.

Model Reference Control of Human-Operated Mobile Robot for Object Transportation

Naoki Uchiyama and Tatsuhiro Hashimoto
Toyohashi University of Technology
Japan

1. Introduction

Robotic systems are expected to engage in various types of tasks, such as housework, nursing and welfare work, and industrial work done by skilled workers. Although fully automated robots are desirable, it appears difficult to produce such robots from the viewpoints of cost efficiency and the technologies available currently. Human-operated robotic systems are a good compromise, and hence are widely studied. Objectives of these robots include extending human mechanical power (Kazerooni & Steger, 2006), providing precise and smooth operation for human workers in difficult tasks (Bettini et al., 2001) (Peshkin et al., 2001), and executing a task in remote or hazardous environment (Anderson & Spong, 1989) (Lawrence, 1993).

In human-operated robotics systems, controllers are required to incorporate the human operator's command and compensate for the operator's mistakes without reducing the ease of operation. For this purpose we propose a model reference control approach, in which the reference model generates a desired trajectory according to the operator's input and constraints such as collision avoidance. This approach is applied to a two wheeled mobile robot that transports an object. This type of robot has various applications in many areas. Because transporting objects is a fundamental task of robotic systems, we realize a function to prevent slip and tumble of the transported object even when the human operator makes mistakes during operation. Fixing the transported object to the robotic system to prevent the object from tumbling requires extra time to transport the object and reduces the operational ease. This is because fixing is a time-consuming and inconvenient task. In particular, supposing that the robot is operated by elderly or disabled people, this function will be necessary for providing easy and safe operations. In addition, a collision avoidance function is implemented by the proposed model reference approach.

Many studies have been conducted into the obstacle avoidance of mobile robots (Bonnafous & Lefebvre, 2004) (Fox et al., 1997) (Khatib, 1986) (Ö gren, P. & Leonard, 2005). Most of the existing approaches provide sophisticated algorithms that minimize some objective functions, such as required time to reach the goal and moving distances. However, these methods assume the fully automatic motion of robotic systems, and hence, the human

operator's commands cannot generally be incorporated in real-time. In addition, tumble avoidance of the transported object is not considered in current methods. In the case of human-operated robot, a simple algorithm for real-time calculation, rather than optimization, is required because the time-consuming processing required may reduce the robot's operativity. The effectiveness of the proposed approach is demonstrated by experimental results, where ten unskilled operators operate the robot with/without the proposed method.

2. Human-Operated Mobile Robot

In this chapter, we consider a control problem of a general type two-wheeled mobile robot that transports an object as shown in Fig. 1. Human operators are enabled to handle the robot using control sticks. They can give command signals for driving forces of each wheel u_l and u_r by inclining left and right sticks, respectively. The magnitudes of driving forces are proportional to the inclined angles of the sticks. The robot dynamics is given as follows:

$$I\ddot{\phi} = (u_r - u_l)L, \tag{1}$$

$$M\dot{v} = u_r + u_l, \tag{2}$$

where I and M are the inertia and the mass of the robot, respectively. The symbol L is the half distance between the two wheels. The symbols v and ϕ are the translational speed and rotation angle of the robot, respectively. The slip of wheels is not considered in this study. The shape of the robot is assumed as a circle for simplicity. Distance sensors to detect obstacles are located symmetrically with respect to the centerline parallel to the translational direction of the robot as shown in Fig. 1. The distance from the center of the robot to each sensor is denoted by R. The located direction of each sensor from a line that links wheel centers is denoted by ψ_i. Note that ψ_i has a positive value.

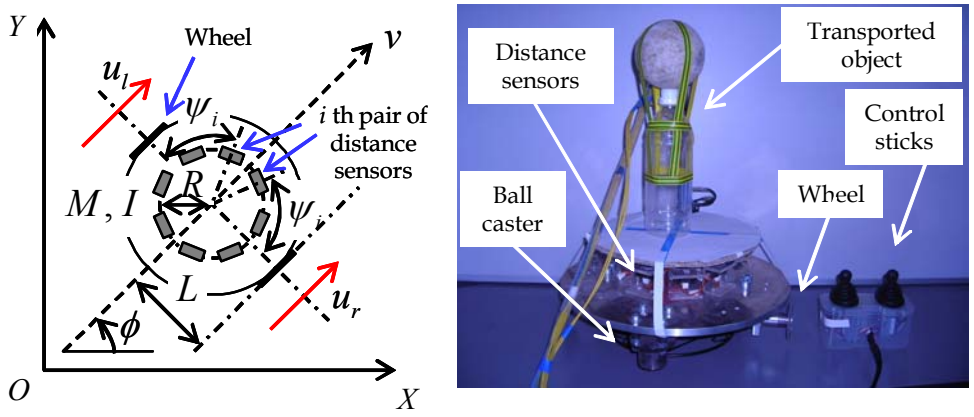


Fig. 1. Human –operated mobile robot that transports an object

3. Controller Design

3.1 Model reference control for obstacle avoidance

To consider the nonholonomicity of the robot and incorporate the operator's command, we propose an obstacle avoidance algorithm based on the model reference approach as shown in Fig. 2, where the reference model generates the desired angles of each wheel, $\dot{\theta}_{ld}$ and $\dot{\theta}_{rd}$, according to the operator's command input and distance sensor information. The reference model, which has a similar dynamics with the mobile robot except for an obstacle avoidance function, is given as follows:

$$I\ddot{\phi} + C_{\phi}\dot{\phi} = (u_r - u_l)L + \sum_{i=1}^{m} \frac{\alpha_i}{\sqrt[n]{d_{ri}}} - \sum_{i=1}^{m} \frac{\alpha_i}{\sqrt[n]{d_{li}}}, \tag{3}$$

$$M\dot{v} + C_v v = u_r + u_l - \sum_{i=1}^{m} \frac{\beta_i}{\sqrt[n]{d_{ri}}} - \sum_{i=1}^{m} \frac{\beta_i}{\sqrt[n]{d_{li}}}, \tag{4}$$

where C_{ϕ} and C_v are the virtual viscous friction coefficients. The viscous friction terms generally exist due to the actuator viscous friction and increase the system stability. We use these terms to increase the control system stability as shown in the analysis in Section 3.2. The symbols d_{li} and d_{ri} are the distances between sensors at angle ψ_i and the obstacle, where the subscripts l and r mean that the sensor is located at the left and right wheel side, respectively. Only sensors that are located in the same half side of the robot body with moving direction v are active. The symbol m denotes the half number of the active sensors. The last two terms on the right-hand side of Eq. (3) give an effect of steering. The magnitude of the steering depends on the distances to the obstacle d_{li} and d_{ri}. The last two terms on the right-hand side of Eq. (4) play a role of brake. The magnitude of braking force also depends on the distances to the obstacle. The symbols α_i and β_i are constant parameters for changing the effects of these steering- and brake-like functions. The nth roots of the distances are employed in these terms for varying the response to the obstacle, and their effects are shown in Fig. 3. Decreasing the value of n increases the effects of steering- and brake-like functions. The reference motion of the robot is obtained by numerically integrating Eqs. (3) and (4). The values of $\dot{\phi}$ and v in Eqs. (3) and (4) are converted into wheel reference signals as follows:

$$R_w\dot{\theta}_{ld} = v - L\dot{\phi}, \; R_w\dot{\theta}_{rd} = v + L\dot{\phi}, \tag{5}$$

where R_w is the radius of wheel.

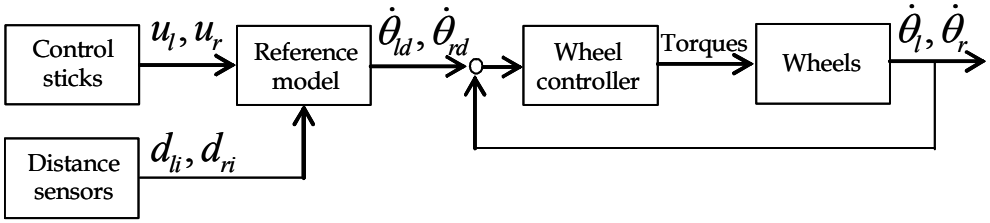


Fig. 2. Block diagram of model reference control approach

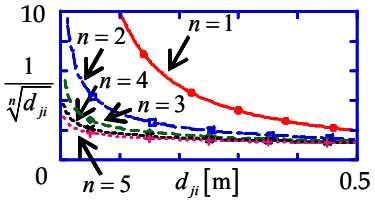


Fig. 3. Properties of functions for obstacle avoidance

3.2 Stability analysis based on linear model

This section presents a stability analysis based on a linear model of the proposed reference model in Eqs. (3) and (4). In this analysis, we consider the case in Fig. 4, where the two parallel walls are obstacles. It is assumed that the mobile robot moves almost along the centerline between the two walls with a velocity $v = v_0 + v_s$, where v_0 is a desired constant and v_s is a small-sized variable. Because the mobile robot is in an almost straight line motion with a constant velocity, it is reasonable to assume that the input from the operator satisfies the relation $u_r + u_l \simeq C_v v_0$ and $u_r \simeq u_l$. We also assume that both the shift from the centerline x_s and the inclination from the lateral line ϕ_s in Fig. 4 are small-sized variables.

The distance between each sensor and walls are given by

$$d_{li} = \frac{D + x_s}{\cos(\psi_i - \phi_s)} - R, \quad d_{ri} = \frac{D - x_s}{\cos(\psi_i + \phi_s)} - R \tag{6}$$

where D is the half distance between the walls. Because ϕ_s and x_s are small-sized variables, the following linear approximation is reasonable:

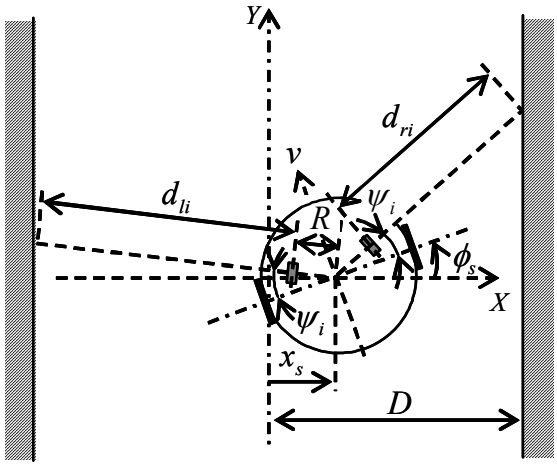


Fig. 4. Schematic for stability analysis

$$d_{li} = a_i - b_i\phi_s + c_i x_s\,,\ d_{ri} = a_i + b_i\phi_s - c_i x_s\,, \tag{7}$$
$$a_i = \frac{D}{\cos\psi_i} - R\,,\ b_i = \frac{D\sin\psi_i}{\cos^2\psi_i}\,,\ c_i = \frac{1}{\cos\psi_i}$$

Note that a_i, b_i and c_i are positive constants.

The following linear approximation is also reasonable because ϕ_s and x_s have small magnitudes:

$$\frac{1}{\sqrt[n]{d_{ji}}} = -p_i d_{ji} + q_i, \quad j = l, r, \tag{8}$$

where p_i and q_i are positive constants.

Equation (3) is written as follows from Eqs. (6) - (8) and the assumptions given above:

$$I\ddot{\phi}_s + C_\phi\dot{\phi}_s = \sum_{i=1}^{m}\alpha_i p_i\left(-2b_i\phi_s + 2c_i x_s\right) \tag{9}$$

The dynamics on x_s is given as:

$$\dot{x}_s = -v\sin\phi_s \simeq -\left(v_0 + v_s\right)\phi_s \simeq -v_0\phi_s \tag{10}$$

Equation (4) is linearized as:

$$M\dot{v}_s + C_v v_s = 2\sum_{i=1}^{m} \beta_i \left(p_i a_i - q_i \right) \tag{11}$$

Because Eq. (11) has no coupling term on ϕ_s and x_s, we consider Eqs. (9) and (10) in the stability analysis. It should be noted that Eq. (11) is stable because M and C_v are positive and the right-hand side is bounded.

Defining a vector $z = \left[\phi_s, \dot{\phi}_s, x_s \right]^T$, we have the following linear dynamics from Eqs. (9) and (10):

$$\dot{z} = Az \tag{12}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{-2}{I}\displaystyle\sum_{i=1}^{m} \alpha_i p_i b_i & \dfrac{-C_\phi}{I} & \dfrac{2}{I}\displaystyle\sum_{i=1}^{m} \alpha_i p_i c_i \\ -v_0 & 0 & 0 \end{bmatrix}$$

The characteristic polynomial of the system Eq. (12) is

$$\left| sI - A \right| = s^3 + \frac{C_\phi}{I} s^2 + \frac{2}{I}\sum_{i=1}^{m} \alpha_i p_i b_i s + \frac{2v_0}{I}\sum_{i=1}^{m} \alpha_i p_i c_i \,. \tag{13}$$

Because all the coefficients of the right-hand side of Eq. (13) are positive, the stability condition is given by:

$$\frac{C_\phi}{I}\sum_{i=1}^{m} \alpha_i p_i b_i - v_0 \sum_{i=1}^{m} \alpha_i p_i c_i > 0 \,. \tag{14}$$

From Eqs. (7) and (14), the following sufficient condition for the stability is derived:

$$C_\phi D \sin\psi_i - I v_0 \cos\psi_i > 0, \quad i = 1, \cdots, m \,. \tag{15}$$

By assigning the coefficient C_ϕ such that Eq. (15) is satisfied, the stability of the linearized dynamics in Eq. (12) is guaranteed.

3.3 Object transportation control

Because transporting objects is a fundamental task of robotic systems, we include a function to prevent slip and tumble of the transported object in the reference model block in Fig. 2 even when the human operator makes mistakes during operation. Fixing the transported

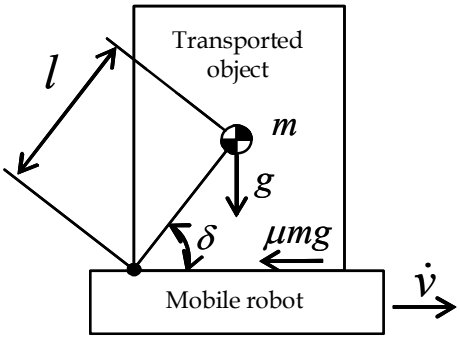


Fig. 5. Derivation of conditions to avoid slip and tumble

object to the robotic system to prevent the object from slip and tumble requires extra time to transport the object and reduces the operational ease.

Because the value of M in Eq. (4) does not necessarily have an exact value of the mass of robot, we change this value in real time to adjust the reference acceleration to prevent the object from slip and tumble. Increasing this value reduces the magnitude of the reference acceleration.

In this study, we assume that the slip and tumble of the transported object is caused mainly by the translational acceleration, although the acceleration normally includes the centrifugal and the Coriolis terms. The slip of the object is prevented if the inertial force is smaller than the static friction force as follows:

$$m|\dot{v}| \le \mu m \mathrm{g}\,, \tag{15}$$

where m is the mass of the transported object, μ is the static friction coefficient of the contacting surface between the object and the robot, and g is the gravitational acceleration. Figure 5 is the schematic of this relation. Hence, we have the allowable acceleration $\dot{v}_{s\max}$ to avoid the slip as follows:

$$\dot{v}_{s\max} = \mu \mathrm{g}\,. \tag{16}$$

Next, we consider the allowable acceleration to avoid the tumble. We assume that the object starts to rotate at the end point of the contacting surface with the robot as shown in Fig. 5. Considering the equation around the center of rotation, we obtain the following condition for preventing the object from starting to tumble.

$$ml|\dot{v}|\sin\delta \le ml\,\mathrm{g}\cos\delta, \tag{17}$$

where δ is the angle from the contacting surface line to the center of gravity of the transported object, and l is the distance between centers of rotation and gravity, as shown in Fig. 5. Hence, we obtain the acceleration limit for avoiding the tumble $\dot{v}_{t\max}$ as follows:

$$\dot{v}_{t\max} = g\cot\delta \tag{18}$$

From Eqs. (16) and (18), the allowable acceleration to avoid the slip and tumble is given as:

$$\dot{v}_{\max} = \min\left(\dot{v}_{s\max}, \dot{v}_{t\max}\right) \tag{19}$$

To avoid the tumble, we propose to adjust the mass coefficient $M(t)$ as follows from Eq. (4) :

$$M(t) = \begin{cases} \dfrac{1}{\dot{v}_{\max}}\left\{-C_v v + u_r + u_l - \displaystyle\sum_{i=1}^{m}\frac{\beta_i}{\sqrt[n]{d_{ri}}} - \sum_{i=1}^{m}\frac{\beta_i}{\sqrt[n]{d_{li}}}\right\}, & \text{if } |\dot{v}| > \dot{v}_{\max}, \\ M_0, & \text{otherwise.} \end{cases} \tag{20}$$

where M_0 is the initial value of the mass coefficient $M(t)$.

4. Experiment

The effectiveness of the proposed controller is experimentally verified in a corridor-like space shown in Fig. 6. Parameter values for the experiment are given in Table 1. Parameters for obstacle avoidance α_i, β_i and n are determined in a trial and error manner. DC servo motors (20 [W]) are employed for each wheel motion. Rotary encoders (500 [PPR]) attached to the motors are used for measuring the position and orientation of the robot. Infrared distance sensors, whose measurable ranges are 4 - 30 [cm], are employed to measure the distance to the obstacle.
To verify the effect for the operational easiness, ten unskilled persons (students) are employed to operate the robot with the transported object $\left(\delta = 75\left[\deg\right]\right)$ in Fig. 1 under the following conditions:

(a1) Manual control
(a2) Control with the obstacle avoidance function presented in section 3.2.
(a3) Control with the obstacle avoidance and the tumble avoidance functions in section 3.3.

Parameter	Value	Parameter	Value	Parameter	Value
I	0.056 [kgm^2]	C_ϕ	3.0 [Nms/rad]	α_i	4 [Nm$^{5/4}$]
M	5.4 [kg]	C_v	10.0 [Ns/m]	β_i	10.0[Nm$^{1/4}$]
R	0.11 [m]	R_w	0.029 [m]	n	4
L	0.17 [m]	m	2		

Table 1. Parameter values in experiment

In (a3), only the tumble is considered because $\dot{v}_{t\max} \gg \dot{v}_{s\max}$ in this experiment.

Figures 7 - 9 show the obtained robot trajectories by one operator under conditions (a1) – (a3), respectively. In case (a1), as negative values of u_l and u_l are shown in Fig. 7 (a), backward motions were required to pass through the course. The backward motion is confirmed in Fig. 7 (d). In addition, both collision and tumble occurred in this case. The latter is caused by a large magnitude of acceleration as shown in Fig. 7 (b).

In case (a2), although u_l and u_l were almost constant during operation as shown in Fig. 8 (a), the robot changed its orientation ϕ in Fig. 8 (c) by the obstacle avoidance function. In addition, no backward motion was required as shown in Fig. 8 (d). However, a large magnitude of the acceleration in Fig. 8 (b) caused the tumble of the transported object.

In case (a3), the robot was enabled to smoothly pass through the course by an almost constant inputs in Fig. 9 (a) without requiring a large magnitude of acceleration as shown in Fig. 9 (b).

Table 2 summarizes experimental results by ten unskilled operators (students), where no collision occurs in cases (a2) and (a3), and no tumble occurs in case (a3), for all operators. Figure 10 summarizes the control time required to pass through the course. The control time is largely reduced for almost all operators by the model reference control approach, because they do not have to consider the obstacle or tumble avoidance during operation. The control time in case (a3) increases little compared to case (a2), although the acceleration magnitude is reduced to avoid the tumble.

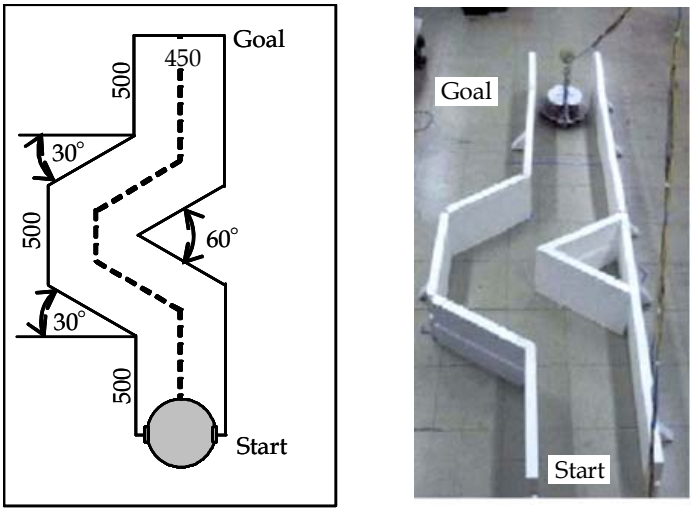


Fig. 6. Experimental environment

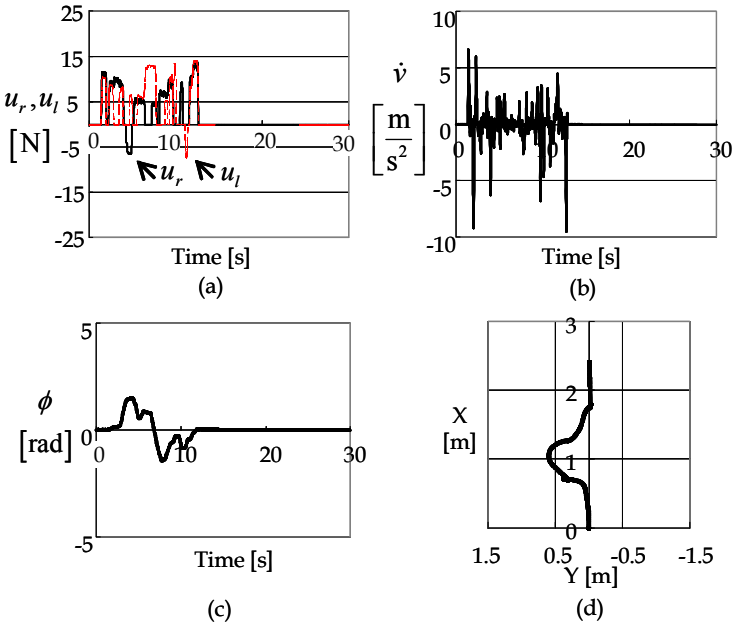


Fig. 7. Experimental results (Manual control)

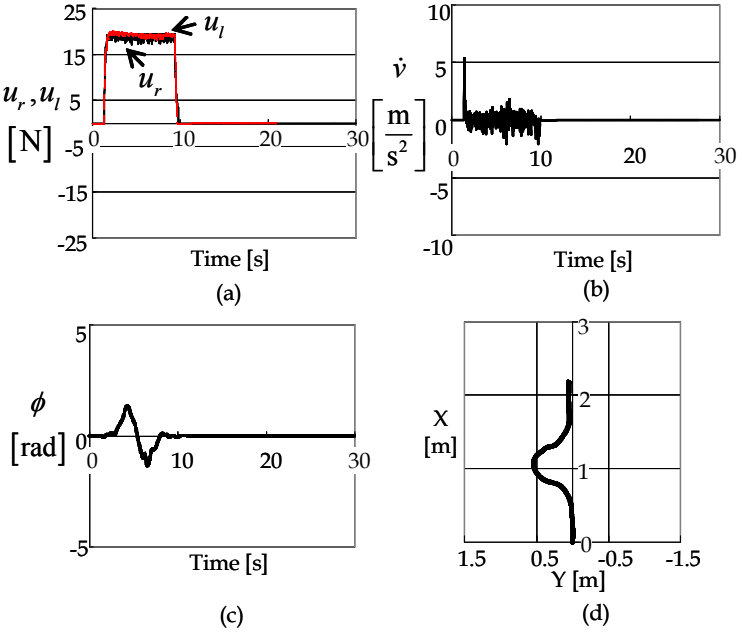


Fig. 8. Experimental results (Obstacle avoidance control)

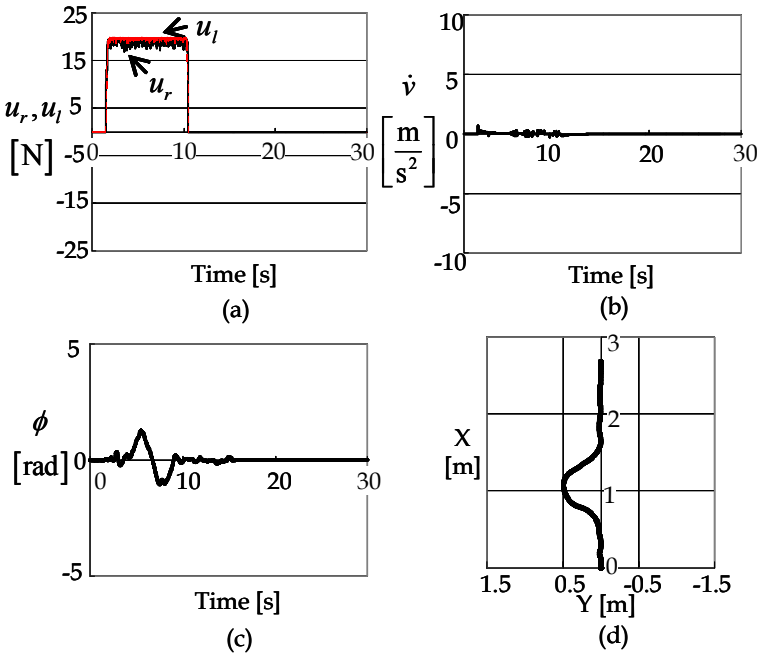


Fig. 9. Experimental results (Obstacle and tumble avoidance control)

○: Not occur, ×: Occur

Operator's No.	(a1) Manual control		(a2) Obstacle avoidance		(a3) Obstacle and tumble avoidance	
	Collision	Tumble	Collision	Tumble	Collision	Tumble
1	×	×	○	×	○	○
2	○	×	○	○	○	○
3	×	×	○	×	○	○
4	○	×	○	○	○	○
5	○	×	○	○	○	○
6	○	×	○	○	○	○
7	○	×	○	×	○	○
8	○	○	○	○	○	○
9	○	○	○	×	○	○
10	○	○	○	×	○	○

Table 2. Summary of experimental results

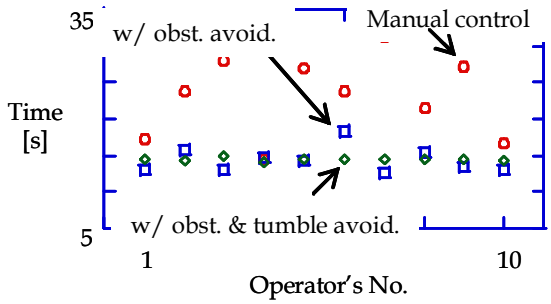


Fig. 10. Required time to pass through course

5. Conclusion

This chapter presents a model reference control approach for a human-operated mobile robot that transports an object. This type of robot has wide applications in industrial and household tasks. The operational easiness of the robot is verified by experiments where operators are required to operate the robot with a transported object to pass through a corridor-like space. Because even young students failed to operate the robot, a function to support the operation is obviously required. The operational easiness is improved by the proposed approach, with which all operators succeeded in transporting the object without collision nor tumble of the object.

6. References

Anderson, R. J. & Spong, M. W. (1989). Bilateral Control of Teleoperators with Time Delay, *IEEE Trans. Automatic Control*, Vol. 34, No. 5, pp. 494-501.

Bettini, A.; Marayong, P.; Lang, S.; Okamura, A. M. & Hager, G. D. (2001). Vison-Assisted Control for Manipulation Using Virtual Fixtures, *IEEE Trans. Robotics and Automation*, Vol. 20, No. 6, pp. 953-966.

Bonnafous, F., D. & Lefebvre, O. (2004). Reactive Path Deformation for Nonholonomic Mobile Robots, *IEEE Trans. Robotics and Automation*, Vol. 20 No. 6 pp. 967-977.

Fox, D., Burgard, W. & Thrun, S. (1997). The Dynamic Window Approach to Collision Avoidance, *IEEE Robotics and Automation Magazine*, Vol. 4, pp. 23-33.

Kazerooni, H. & Steger, R. (2006). The Berkeley Lower Extremity Exoskeleton, *ASME J. Dyn. Syst. Meas., Control*, Vol. 128, No. 1, pp. 14-25.

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *Int. J. Robotics Research*, Vol. 5, No. 1, pp. 90-98.

Lawrence, D. A. (1993). Stability and Transparency in Bilateral Teleoperation, *IEEE Trans. Robotics and Automation*, Vol. 9, No. 5, pp. 624-637.

Ö gren, P. & Leonard, N. E. (2005). A Convergent Dynamic Window Approach to Obstacle Avoidance, *IEEE Trans. Robotics and Automation*, Vol. 21, No. 2, pp. 188-195.

Peshkin , M. A.; Colgate J. E.; Wannasuphoprasit W.; Moore C. A.; Gillespie B. & Akella P. (2001). Cobot Architecture, *IEEE Trans. Robotics and Automation*, Vol. 17, No. 4, pp. 377-390.

Diagnosis of Intermittent Faults and its dynamics

A. Correcher, E. García, F. Morant, E. Quiles and L. Rodríguez
Universidad Politécnica de Valencia
Spain

1. Introduction

Intermittent faults (IFs) are difficult to diagnose and may cause a great disruption in industrial processes. Most IFs are related to gradual degradation of components or systems. For instance, evolution of connection failures is shown in Fig. 1 (Correcher et al., 2004), (Sorensen et al, 1998). Connection failures are rarely repaired so its behaviour worsens over time. Intermittent faults behave as small noise fluctuations in stage 1 of their development. As the amplitude and duration of the fluctuations increase (stage 2), IFs start to occur. The effects of IFs are severe in stage 3.
Therefore, in many instances, the occurrence of IFs in a device is a prelude of permanent failures (PFs). In these cases, if IFs can be detected then appropriate actions could be taken in order to minimise the economic impact.
In (Correcher et al., 2004) an IFs diagnosis tool was presented by the authors. This tool was able to diagnose the failure and recovery events in a system with IFs. This paper presents an extension of the work in (Correcher et al., 2004) which includes not only event detection but also fault dynamics detection, defined as the evolution of IFs occurrence over time. Other approaches to IFs diagnosis (Contant et al., 2004), (Jinag et al. 2003), do not consider IF dynamics.

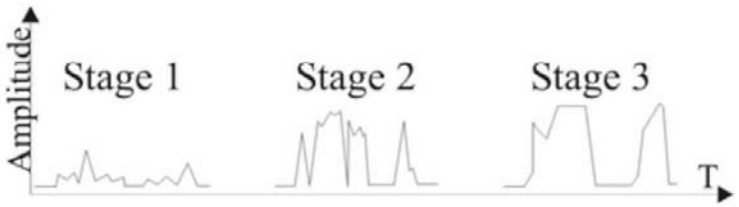


Fig. 1. Connection IF evolution through device life.

However, the existence of IF dynamics is experimentally shown in (Sorensen et al., 1998) and in the destructive tests presented in this paper. Therefore, we can introduce the diagnosis problem to be addressed.

Definition 1: IF diagnosis problem.
Starting from a temporal input *(U)* and output *(Y)* sequence, obtained by means of sensors in the process, compute the presence of any failure *f* (with *f* included in a failure set), its recoveries and its dynamics.
The proposed solution has a clear industrial interest, as not only diagnoses IFs, but also provides valuable information on the fault evolution for maintenance purposes. First, section 2 presents a study about the evolution of the IF during its online diagnosis. This approach is able to extract some characteristic parameters of the IF. These parameters will be useful for estimating the behaviour of the fault in the future. The approach is also applied to experimental data.
Section 3 analyses the effectiveness of the approach in the solution of the problem stated in definition 1 when diagnosing Discrete Event Systems (DES) (Cassandras & Lafortune, 1999). Section 4 presents an application of IF dynamics diagnosis with Coloured Petri Nets. Finally, we present some conclusions in section 5.

2. Temporal modeling

IF dynamics characterization generates useful information for preventive maintenance scheduling. Two complementary parameters are defined in this section: temporal failure density (DF) and pseudoperiod *(Ps)*. The goal of these parameters is to characterize the IF dynamics. *DF* and *Ps* can be on-line computed. *DF* and *Ps* can also predict future behaviour of the faulty device.
DF and *Ps* are computed from IF time occurrence and IF duration (defined as the time difference between fault and recovery). Therefore, two arrays are computed for each fault: the fault time vector ($FT_{Fj}=[FT_{(1)Fj},FT_{(2)Fj},\ldots,FT_{(n)Fj}]$) and the duration time vector ($T_{Fj}=[T_{(1)Fj}, T_{(2)Fj},\ldots,T_{(n)Fj}$]), where F_j stands for a fault included in the fault set and "*n*" is the index number of detected faults. Arrays and parameters are computed recursively on-line considering a moving time window of a given duration.

2.1 Temporal failure density

Temporal failure density *(DF* or density in the rest of the paper) is defined as the average time a particular fault (F*j*) is active within a sliding time window of duration *W*. Therefore, if we define the current time as "*ti*", the density is computed from "*ti-W*" to "*ti*". Therefore, *DF* is computed for time "*ti*" as:

$$DF_{ti} = \frac{\sum_{i=k}^{CNT}\left(T_{(i)Fj} - T_A\right)}{W}; CNT > 0 \tag{1}$$

where *CNT* is the number of faults inside the window, "*k*" stands for the index of the first fault detected inside the window {*k: FT(k)Fj ≥ (ti-W)* }(k: $FT_{(k)Fj} > (ti\text{-}W)$ and $FT_{(k-1)Fj}<(ti\text{-}W)$} if exists, otherwise *{k=CNT+1}* and "T_A" takes into account the duration of a fault occurred before "*ti-W*" which continues active inside the window. Therefore:

$$T_A = FT_{(k-1)Fj} + T_{(k-1)Fj} - (ti - W) \tag{2}$$

Equation (2) is valid only if "T_A" is positive, otherwise "$T_A=0$", as this fact would indicate that the first considered fault is completely outside of the window.
In a real system, *DF* tends to increase with time; thus confirming the hypothesis that IFs progressively damage the faulty device. Figures 2 and 3 show the computed *DF* from experimental data and its filtered signal (low-pass fourth order Butterworth filter). The experimental data has been obtained from ten million operation tests on relays switching a resistive load. As the time between operations was 100 milliseconds, the overall duration of each experiment was 277.8 hours.

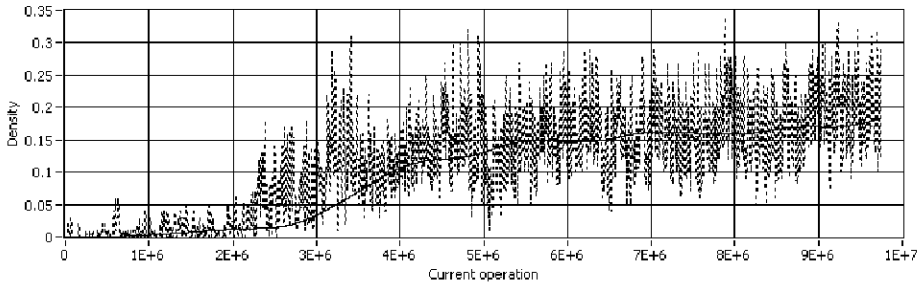


Fig. 2. Fault density. Window size is 10000 operations.

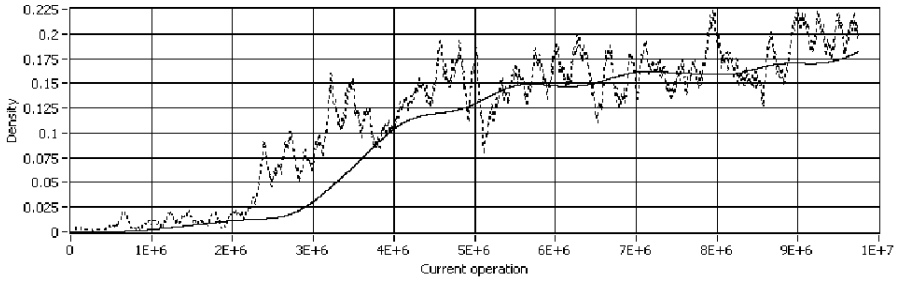


Fig. 3. Fault density. Window size is 100000 operations

The rising characteristic of the density can be used to estimate the optimal time to repair or substitute the faulty device. Effectively, a certain maximum density threshold can be defined as the limit for unacceptable behaviour. Then, an adequate extrapolation model can be used to predict the number of operations the device is capable of carrying out before reaching the unacceptable behaviour limit. Obviously, the unacceptable density threshold should be defined specifically for each process device, depending on its specific functionality and reliability requirements.
The simplest prediction model consists of a density linear increase. In this case, filtered density data can be treated with classical techniques (Lemmis, 1995) such as least squares (LS) or recursive least squares *(RLS)*. Therefore, we obtain a model:

$$D = m_{ti} \bullet t + n_{ti} \tag{3}$$

where D stands for the density, t stands for the time and the subindex ti stands for the time value when density is estimated. If we consider a threshold "D_0", the time (t_{D0}) when the density "D_0" will be reached is:

$$t_{D0} = \frac{(D_0 - n_{ti})}{m_{ti}} \tag{4}$$

Therefore, "t_{D0}" is the time when the faulty device should be replaced. This time is named as Linear Substitution Time at time "ti" (LST_{ti}). In addition, it is possible to define another parameter much more suitable for preventive maintenance: Operations to Replacement at time "ti" (OTR_{ti}). This parameter represents an estimation of the useful operations left on a device, and can be computed as:

$$OTR_{ti} = LST_{ti} - ti \tag{5}$$

Obviously, only positive values of OTR_{ti} are meaningful, otherwise the corresponding OTR_{ti} is considered equal to zero.

The proposed linear prediction model has been found to be suitable for the use with the experimental used through the chapter. However, this kind of model might not be adequate for other devices. For instance, if the fault density follows first order dynamics then $LSTti$ will predict an optimistic substitution time. This problem can be solved by using *RLS* with forgotten factor to fit LST_{ti}. In any case, LST_{ti} and OTR_{ti} reflect the underlying fault dynamics, and could be used to model systems that do not follow linear increase laws.

Therefore, the fault diagnosis system will predict the time when a device must be replaced in two stages. First, fault density will be computed and, from this value, LST_{ti} and OTR_{ti} will be predicted.

As mentioned before, the sliding window size should be appropriately chosen, as short windows will imply high variability and noise in the calculated failure density and long windows, which exhibit a greater filtering effect, involve high computational costs and might mask part of the fault underlying dynamics.

Figures 2 and 3 show the effect of window size in the variability of the density. From calculations with a range of window sizes, it has been found that windows greater than 5000 operations include the same low frequency component, as shown in figs. 2 and 3. Therefore, LST_{ti} and OTR_{ti} computed from window sizes greater than 5000 operations are identical. Figure 4 shows the evolution of OTR_{ti} obtained from the experiments in figs 2 and 3. Figure 4 shows that the device should be replaced when reaching 6 million operations instead of the substitution time recommended by the manufacturer (10 million operations).

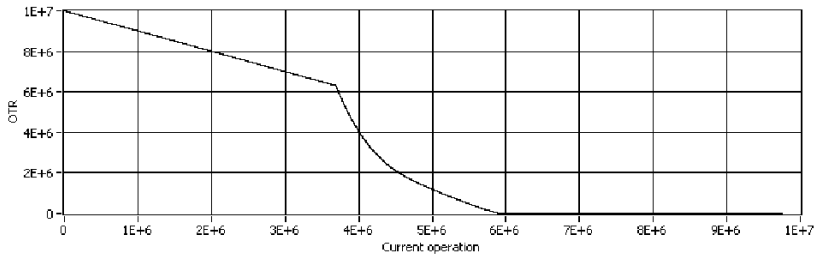


Fig.4. OTR_{ti} for an acceptable density threshold below 15%.

2.2 Pseudoperiod.

Fault density can be used to predict the device substitution time, however it does not completely explain IF dynamics. For instance, fig. 5 shows two cases with exactly the same failure density. However, the effects on the device are clearly different.

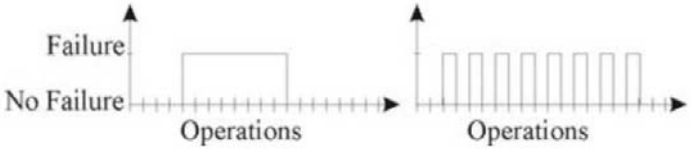


Fig. 5. IF with the same density but different dynamics.

The difference between the two behaviours can be modelled by the time difference between the occurrences of two consecutive faults.

The time difference can be measured with a new parameter, the Pseudoperiod *(Ps)*. *Ps* is defined as the average time difference between faults inside a sliding window. Moreover, *Ps* is normalized by the number of faults, and can be computed at time "*ti*" as:

$$Ps_{ti} = \frac{\sum_{i=j}^{i=k} FT_{i+1} - FT_i}{k - j + 1} \tag{6}$$

where "ti" stands for current time, "*FT*" is the detection time for fault "*i*", and "*j*", "*k*" are the first and last fault indexes in the window, respectively.

Pseudoperiod is clearly a magnitude related to mean time between failures *(MTBF)*, commonly used to model reliability of reparable systems. Moreover, pseudoperiod is a dynamic magnitude. Therefore, we can compute a *Ps* curve for any IF. This curve can be used to predict the substitution time of the device.

The evolution of Pseudoperiod (figs. 6 and 7) shows an increase until a maximum value is reached, to decrease towards a value close to zero. This dynamic behaviour is consistent with the nature of IFs (fig. 1). The computed Pseudoperiod remains in the range 600 to 800 from 4 million operations onwards. Therefore, it is possible to conclude that, in average, the number of failures from the 4 millionth operation remains reasonably constant. Moreover,

the average duration of each failure slowly increases with the number of operations, as the density (fig. 3) increases.

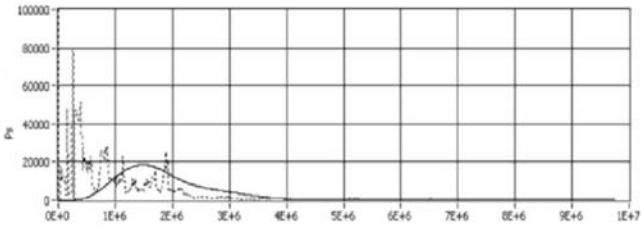


Fig. 6. Pseudoperiod. Window size is 100000 operations.

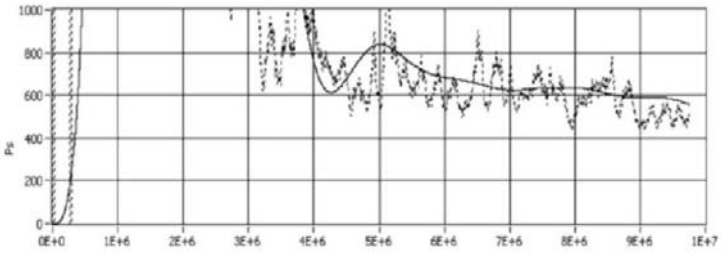


Fig. 7. Zoom in Pseudoperiod. Window size is 100000 operations.

Pseudoperiod can be used to compute some limit in the desirable behaviour of the system. The shape of the Pseudoperiod signal suggests the estimation of this limit with *RLS* with forgetting factor. Another solution could be a mixed model with a polynomial and linear estimation for each side of the signal. This last approach seems to be more promising but its computing is no trivial. Continuity and derivability must be guaranteed. Moreover, the order of the filter used in the Pseudoperiod signal will affect in the delay of the model. These problems will be addressed in future works.

3. Intermittent fault dynamics diagnosability

It is necessary to ascertain if the proposed dynamic parameters can be used to perform the complete fault diagnosis as per definition 1. To complete definition 1, a definition of fault dynamics diagnosis is introduced, based on the definition of discrete-time systems observability (Smolensky et. al, 1996) "A discrete-time system is observable if a finite *k* exists such that knowledge of the outputs to *k-1* is sufficient to determine the initial state of the system. "

Definition 2. IF dynamics diagnosis problem.

Given a temporal fault sequence, $TF = \{tf_i\}\ (i = 0...n)$, and a temporal recovery sequence, $TRF = \{trf_j\}(j = 0...m)$, compute the next value in TF if $m \geq n$ or the next value in TRF if $m < n$.

Definition 2 states that IF dynamics is diagnosable if we can compute the next time when a fault or recovery will occur. IFs are asynchronous and non-deterministic. So, IF dynamics cannot be diagnosed with deterministic precision. Therefore, we propose a relaxed definition.

Definition 3. Bounded IF dynamics diagnosis problem.
Given a temporal fault sequence, $TF = \{tf_i\}\ (i = 0...r)$, and a temporal recovery sequence, $TRF = \{trf_j\}\ (j = 0...s)$, compute the next value in TF *if* $s \geq r$ or the next value in TRF *if* $s < r$, with a bounded uncertainty.

This uncertainty can be used to compare different methods of diagnosis.

3.1 Bounded IF dynamics diagnosis with $DF_{T\tau}$.

IF dynamics diagnosis start from a temporal fault event sequence and a temporal recovery event sequence until the actual time $\tau\ (TF = \{tf_i\}(i = 0...s), TRF = \{trf_j\}(j = 0...r))$. DF can be computed as in equations (1) and (2). Let us assume that $T_A=0$, therefore, the next event will be a fault. In this case, the problem in definition 3 will consist of computing the next fault time. As historical data is known until time τ, it is possible to compute a sequence of fault densities $\{D_l\}(l=0...\ \tau)$. Density *RLS* prediction computes the estimated density for a given instant of time υ: $D_\upsilon = m \bullet \upsilon + n$, where m and n are the results of the *RLS* estimation. Density increases when there is a new diagnosed fault in the system. Moreover, density increases for a maximum of "t_m/w", where "t_m" is the sampling period and "w" is the window size. Therefore, the next failure will occur when the linear model will estimate this density value.

$$D_\tau + \frac{t_m}{w} = m \bullet TF_{s+1} + n \tag{7}$$

So

$$TF_{s+1} = \frac{w \bullet (D_\tau - n) + t_m}{w \bullet m} \tag{8}$$

The estimation error is therefore, the error in the *RLS* estimation (Smolensky, et. al, 1996).
If we want to compute the time for a recovery, ($T_A \neq 0$) the density will decrease in, at least, "t_m/w", so:

$$RTF_{r+1} = \frac{w \bullet (D_\tau - n) - t_m}{w \bullet m} \tag{9}$$

We can conclude that the density allows for the complete diagnosis of the IF dynamics with a bounded error.
In order to compute the IF density, the fault diagnosis system should include the identification of fault starting time and duration. The fault diagnosis system should also be able to compute the corresponding fault densities.

4. Latent Nestling.

The previous section showed that the diagnosis of an IF involves not only the diagnosis of fault and recovery events, but also the diagnosis of its dynamics. This section shows how to diagnose IF dynamics with the methodology based on Coloured Petri Nets (CPN) presented in (Garcia et al., 2008) (Rodriguez et al., 2008). This methodology allows IF dynamics diagnosis because it includes timing information. We also include a complete diagnosis example of an industrial process.

A Coloured Petri Net for Fault Diagnosis (DCPNs) is:

$$D = (P, T, Post, M_0, C, f, PLNf, T_f, PVf) \tag{10}$$

where

- P is a finite set of places.
- T is a finite set of transitions.
- Pre and $Post$ are input and output arc functions.
- M_0 is the initial marking.
- C is the colour set assigned to different identifiers. $C = N \cup f$ is the subset of coloured tokens representing the normal system behaviour.
- $f = \{f_1, f_2, ..., f_i\}$ is the subset of coloured tokens representing fault set.
- $PLNf \subseteq P$ is the subset of fault latent nestling places.
- $PVf \subset P$ is the subset of fault verification places.
- $T_f \subset T$ is the transition subset including coloured functions. T_f links $PLNfi$ with PVf.

Fault verification places are P-timed. Therefore, they include pairs <R, Tim_R>; where R is a coloured mark and Tim_R is a timer. Tim_R will be used to compute IF Density and Pseudoperiod.

Thorough this paper the notation $M(Pl(\langle V \rangle))$\{Pl ∈ P; V ∈ C\} will refer to the marking of place "Pl". This notation represents that there is a "V" coloured token in place "Pl" (for Pre functions). This notation also represents that a "V" coloured token is placed in "Pl" (for $Post$ functions).

4.1 System modelling

The first step of the method consists of the dynamic system modelling. The system model is designed with generalized Petri Nets (PNs) (David & Alla, 2005), for simple systems, or with CPNs (for complex systems) (Hensen, 1997).

Let us show the methodology applied to a rectifying industrial machine. The machine rectifies blocks of a synthetic compound that imitates the natural stone. Figure 8 shows the process scheme. The rectifying process consists of the mechanical elimination of some material in order to achieve the desired width. The system can be divided into four subsystems. Since each milling works with an independent motor, each subsystem will consist of a pair motor-milling with a blade cooling and lubrication system. The blade cooling system consists of a pair pump-valve that pours cutting oil over the millings. The motors can suffer Ifs resulting from fretting corrosion in their electrical contacts. The millings fail when there is any defect in the tool. This failure is a PF that can be due to

maintenance failure. Milling failure will cause a great torque and, therefore, a power consumption greater that usual.

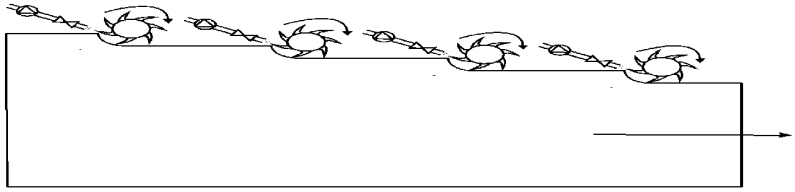


Fig. 8. Artificial Stone rectifying process.

Moreover, a fault in a previous subsystem can cause the same symptoms because the milling will cut more material. The cooling and lubrication system can also suffer IFs. Typical IFs in a pump-valve system are electrical contact failures and valve blocking.
Figure 9 and table 1 show the PN system model. Table 2 shows the relationship between places and system states.

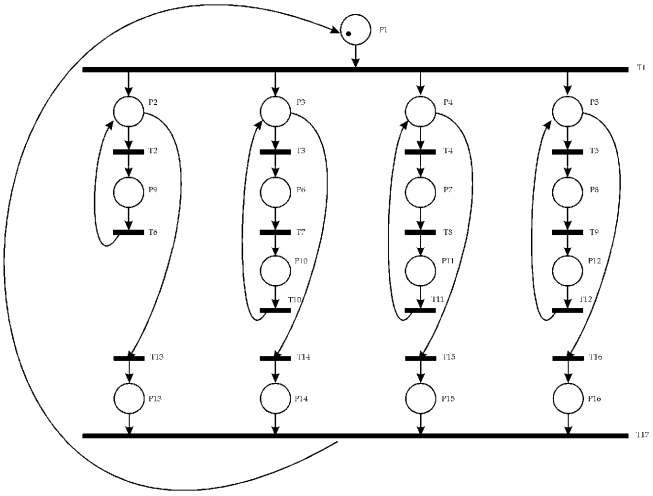


Fig. 9. System PN model.

Tr	Event	Tr	Event	Tr	Event
T1	Start	**T7**	New stone	**T13**	Stop
T2	New stone	**T8**	New stone	**T14**	Stop
T3	*t2/p3/#1s*	**T9**	New stone	**T15**	Stop
T4	*t3/p4/#2s*	**T10**	*tp2/p9/#10s*	**T16**	Stop
T5	*t4/p4/#3s*	**T11**	*tp3/p9/#10s*	**T17**	Stop
T6	*tp1/p9/#10s*	**T12**	*tp4/p9/#10s*		

Table 1. Transitions in PN system model.

Place	System state
P1	Stopped
P2 to P5	Motor started; Milling rotating, not cutting; Pump on; valve closed.
P6 to P8	Motor started; milling rotating, not cutting; Pump on; valve closed; waiting for stone arrival
P9 to P12	Motor started; Milling cutting; Pump on; valve opened
P13 to P16	Motor stopped, Milling stopped; Pump off; valve closed; Shut down control actions

Table 2. Places in PN system model.

The next step consists of the folding to a CPN (Garcia et al., 2008). Figure 10 shows the result. The CPN system model stars in *Pc1*. *Tr1* starts all subsystems and generates normal working tokens in *Pc2*. Arc functions *g* and *g1* denote the relationship between the general normal working token with particular normal working tokens:

$$\left[g\left(\langle \bullet n \rangle\right) = C \right.$$

$$\left[g1\left(\langle C \rangle\right) = \bullet_n \right.$$

$$C = \langle S1_n \rangle + \langle S2_n \rangle + \langle S3_n \rangle + \langle S4_n \rangle$$

where $Sj_n\left(j \in \{1..4\}\right)$ stands for normal subsystem coloured token.

Transition *Tr2* puts a stone in the machine. This action starts the first subsystem (place *Pc4*) and moves the other three subsystems to "waiting for stone arrival" state (*Pc3*). Function *gs1* and *gsk* model the different paths followed by subsystems:

$$\left[gs1\left(\langle S1_n \rangle\right) = \langle S1_n \rangle \right.$$

$$\left[gsk\left(\langle S2_n \rangle + \langle S3_n \rangle + \langle S4_n \rangle\right) = \langle S2_n \rangle + \langle S3_n \rangle + \langle S4_n \rangle \right.$$

Transition $Tr3 = \{T7, T8, T9\}$ starts sequentially the other subsystems by marking *Pc4*. Transition $Tr4 = \{T6, T10, T11, T12\}$ stops sequentially the cutting process of each subsystem. Transition *Tr5* starts the shutdown routine for every subsystem.

4.2 Fault set definition

The next step is the fault analysis of system devices. The goal is to define the faults to be diagnosed in each device. Each fault has a coloured fault token. Therefore, the fault set consists of the union of the coloured fault tokens $f = \{f1, f2, \ldots, fi\}$. Fault isolation will be guaranteed because any fault is associated to a device. Each subsystem includes four devices: motor, milling, pump, and valve. Table 3 shows the faults included in the example:

Fault	Definition	Effect	Nature
f_1^j $j \in \{1..4\}$	Subsystem j Motor fault. Fault in the electrical contacts	Motor stopped	Intermittent
f_2^j $j \in \{1..4\}$	Subsystem j Milling fault.	Milling cuts less material	Permanent
f_3^j $j \in \{1..4\}$	Subsystem j Pump fault. Fault in the electrical contacts	Pump stopped	Intermittent
f_4^j $j \in \{1..4\}$	Subsystem j valve fault.	Valve blocked in opened position	Intermittent
f_5^j $j \in \{1..4\}$	Subsystem j valve fault.	Valve blocked in closed position	Intermittent

Table 3. Fault set definition

4.3 Places of latent nestling faults, *PLNf*

The next step consists of marking all fault coloured tokens in specific CPN places. These places are called "Places of latent nestling faults" *(PLNf)*. Expert's empirical knowledge sets the rules for this operation. Figure 10 shows the marking for this example.

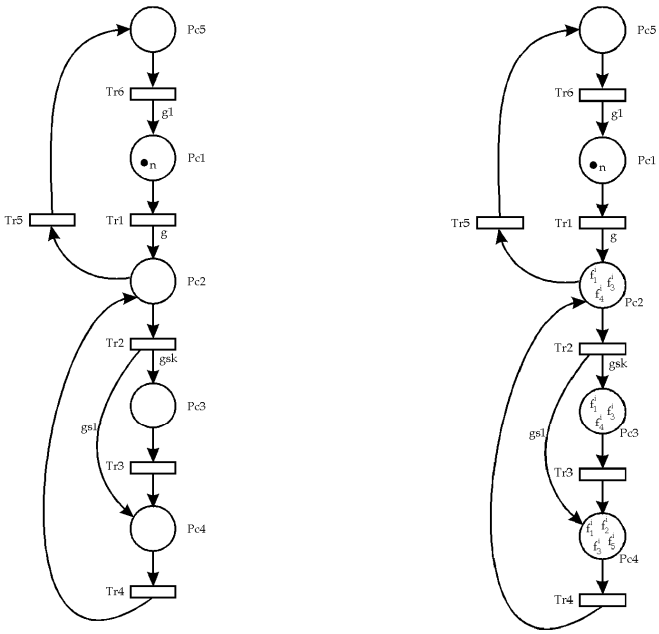


Fig. 10. System CPN model and fault allocation $(j=\{1,2,3,4\})$.

The computing of the thresholds to generate $k(N)$ and $I_i(S)$ events is not trivial, because the sensor will observe an overcurrent when the tool touches the block (normal working). Nevertheless, we can easily generate these events if having a normal working current pattern. Let us suppose that we have a current pattern for each motor: $Pat_t(t)$. The continuous measure of the sensor is $I_i(t)$. Therefore, for each time t_k:

$$|Pat_i(t_k) - I_i(t_k)| < d_{\max} \Rightarrow normal\ working$$
$$|Pat_i(t_k) - I_i(t_k)| > d_{\max} \Rightarrow faulty\ working$$

where d_{max} is the maximum difference allowed between both signals. Therefore, we can define the three current events as:

$$|Pat_i(t_k) - I_i(t_k)| < d_{\max}\ and \begin{cases} |I_i(t_k)| < \gamma \Rightarrow I_i(0) \\ |I_i(t_k)| > \gamma \Rightarrow I_i(N) \end{cases}$$
$$|Pat_i(t_k) - I_i(t_k)| > d_{\max}\ and \begin{cases} |I_i(t_k)| < \gamma \Rightarrow I_i(0) \\ |I_i(t_k)| > \gamma \Rightarrow I_i(S) \end{cases}$$

where y is close to zero and it allows noise filtering.

Flow sensors, Fl_i ($i \in \{1..4\}$) generate two levels: $Fl_i(0)$, no flow; $Fl_i(1)$, flow. Pressure sensors, Pr_i ($i \in \{1..4\}$) generate two levels: $Pr_i(0)$, no pressure; $Pr_i(1)$, pressure.

The set of sensor values is *SM*. The set of possible sensor values combinations for marking M_k is denoted as "$SROV_j(M_k)$" ($j \in \{1...n\}$), where "n" is the number of possible combinations. "$SROV_j(M_k)$" can be split into two subsets: $SROV(M_k)=SROV_{ev}(M_k) \cup SROV_{nev}(M_k)$. "$SROV_{ev}(M_k)$" stands for the subset of expected values and "$SROV_{nev}(M_k)$" stands for the subset of non expected. We use "$SROV_{nev}(M_k)$" to generate the trajectories of fault verification.
We use this information to build the system diagnoser. The diagnoser consists of the extension of the system model by including the trajectories of fault verification. The complete algorithm is shown in (Garcia et al., 2008). Figure 12 shows the CPN diagnoser. Figure 12 duplicates places *Pc2*, *Pc3*, and *Pc4*. Nevertheless, the diagnoser includes only one place *Pc2*, one *Pc3*, and one *Pc4*. We use the representation in figure 12 to increase the readability of the figure.
Table 4 defines the trajectories of fault verification for the diagnoser. "*" stands for any value of other sensors. Function *g2* evaluates place markings and returns fault marks when necessary.

g2(Pl,V) {Pl ∈P; V ∈f} :
if $M(Pl(\langle V \rangle))$ *then null else* $M(Pl(\langle V \rangle))$

The diagnoser must solve the problem of chained faults. The main effect of any subsystem fault is less material cut. Therefore, the following subsystem will observe greater currents. Nevertheless, this situation does not involve two faults. *Tf3* (table 4) solves the problem by including previous subsystem faults in the diagnosis.

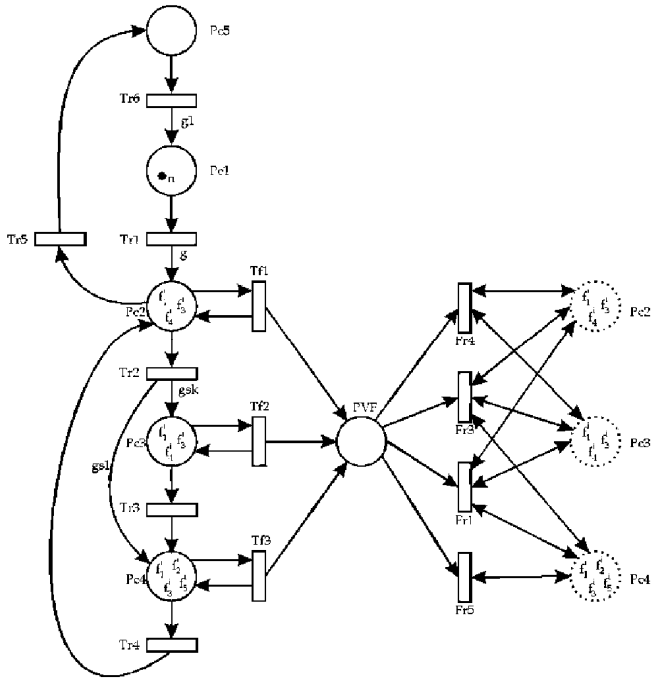


Fig. 12. Diagnoser

T	Pre	Sensors	Post	j
Tf1	$M(Pc2(\langle Sj_n\rangle,\langle fj\rangle))$	$I_1(0),*$	$M(Pc2(\langle Sj_n\rangle))\wedge M(PVF(\langle fj\rangle))$	(1,2,3,4)
	$M(Pc2(\langle Sj_n\rangle,\langle f4\rangle))$	$FI_1(1),*$	$M(Pc2(\langle Sj_n\rangle))\wedge M(PVF(\langle f4\rangle))$	(1,2,3,4)
	$M(Pc2(\langle Sj_n\rangle,\langle fs\rangle))$	$Pr_1(0),*$	$M(Pc2(\langle Sj_n\rangle))\wedge M(PVF(\langle fs\rangle))$	(1,2,3,4)
Tf2	$M(Pc3(\langle Sj_n\rangle,\langle fj\rangle))$	$I_2(0),*$	$M(Pc3(\langle Sj_n\rangle))\wedge M(PVF(\langle fj\rangle))$	(2,3,4)
	$M(Pc3(\langle Sj_n\rangle,\langle f4\rangle))$	$FI_2(1),*$	$M(Pc3(\langle Sj_n\rangle))\wedge M(PVF(\langle f4\rangle))$	(2,3,4)
	$M(Pc3(\langle Sj_n\rangle,\langle fs\rangle))$	$Pr_2(0),*$	$M(Pc3(\langle Sj_n\rangle))\wedge M(PVF(\langle fs\rangle))$	(2,3,4)
Tf3	$M(Pc4(\langle Sj_n\rangle,\langle fj\rangle))$	$I_3(0),*$	$M(Pc4(\langle Sj_n\rangle))\wedge M(PVF(\langle fj\rangle))$	(1,2,3,4)
	$M(Pc4(\langle S1_n\rangle,\langle f^1\rangle))$	$I_3(S), FI_3(1),*$	$M(Pc4(\langle S1_n\rangle))\wedge M(PVF(\langle f^1\rangle))$	
	$M(Pc4(\langle S2_n\rangle,\langle f^2\rangle,\langle f1^1\rangle,\langle f^1\rangle,\langle fs^1\rangle,\langle fs^1\rangle))$	$I_3(S), FI_3(1),*$	$M(Pc4(\langle S2_n\rangle,\langle f1^1\rangle,\langle f^1\rangle,\langle fs^1\rangle))\wedge M(PVF(\langle f^2\rangle))$	
	$M(Pc4(\langle S3_n\rangle,\langle f^3\rangle,\langle f^2\rangle,\langle f^2\rangle,\langle f^2\rangle,\langle fs^2\rangle))$	$I_3(S), FI_3(1),*$	$M(Pc4(\langle S3_n\rangle,\langle f^2\rangle,\langle f^2\rangle,\langle f^2\rangle,\langle fs^2\rangle))\wedge M(PVF(\langle f^3\rangle))$	
	$M(Pc4(\langle S4_n\rangle,\langle f^4\rangle,\langle f^3\rangle,\langle f^3\rangle,\langle f^3\rangle,\langle fs^3\rangle))$	$I_4(S), FI_4(1),*$	$M(Pc4(\langle S4_n\rangle,\langle f^3\rangle,\langle f^3\rangle,\langle f^3\rangle,\langle fs^3\rangle))\wedge M(PVF(\langle f^4\rangle))$	
	$M(Pc4(\langle Sj_n\rangle,\langle fs\rangle))$	$FI_3(0),Pr_3(1),*$	$Pc4(\langle Sj_n\rangle))\wedge PVF(\langle fs\rangle))$	(1,2,3,4)
	$M(Pc4(\langle Sj_n\rangle,\langle fs\rangle))$	$FI_3(0),Pr_3(0),*$	$M(Pc4(\langle Sj_n\rangle))\wedge M(PVF(\langle fs\rangle))$	(1,2,3,4)
Fr1	$M(PVF(\langle fj\rangle))\wedge M(Pc2(\langle Sj_n\rangle))$	$I_1(N),*$	$M(Pc2(\langle Sj_n\rangle))\wedge g2(Pc2, fj)\wedge g2(Pc3, fj)\wedge g2(Pc4, fj)$	(1,2,3,4)
	$M(PVF(\langle fj\rangle))\wedge M(Pc3(\langle Sj_n\rangle))$	$I_2(N),*$	$M(Pc3(\langle Sj_n\rangle))\wedge g2(Pc2, fj)\wedge g2(Pc3, fj)\wedge g2(Pc4, fj)$	(2,3,4)
	$M(PVF(\langle fj\rangle))\wedge M(Pc4(\langle Sj_n\rangle))$	$I_3(N),*$	$M(Pc4(\langle Sj_n\rangle))\wedge g2(Pc2, fj)\wedge g2(Pc3, fj)\wedge g2(Pc4, fj)$	(1,2,3,4)
Fr3	$M(PVF(\langle fs\rangle))\wedge M(Pc2(\langle Sj_n\rangle))$	$Pr_1(1),*$	$M(Pc2(\langle Sj_n\rangle))\wedge g2(Pc2, fs)\wedge g2(Pc3, fs)\wedge g2(Pc4, fs)$	(1,2,3,4)
	$M(PVF(\langle fs\rangle))\wedge M(Pc3(\langle Sj_n\rangle))$	$Pr_2(1),*$	$M(Pc3(\langle Sj_n\rangle))\wedge g2(Pc2, fs)\wedge g2(Pc3, fs)\wedge g2(Pc4, fs)$	(2,3,4)
	$M(PVF(\langle fs\rangle))\wedge M(Pc4(\langle Sj_n\rangle))$	$Pr_3(1),*$	$M(Pc4(\langle Sj_n\rangle))\wedge g2(Pc2, fs)\wedge g2(Pc3, fs)\wedge g2(Pc4, fs)$	(1,2,3,4)
Fr4	$M(PVF(\langle f4\rangle))\wedge M(Pc2(\langle Sj_n\rangle))$	$FI_1(0),*$	$M(Pc2(\langle Sj_n\rangle))\wedge g2(Pc2, f4)\wedge g2(Pc3, f4)$	(1,2,3,4)
	$M(PVF(\langle f4\rangle))\wedge M(Pc3(\langle Sj_n\rangle))$	$FI_2(0),*$	$M(Pc3(\langle Sj_n\rangle))\wedge g2(Pc2, f4)\wedge g2(Pc3, f4)$	(2,3,4)
Fr5	$M(PVF(\langle fs\rangle))\wedge M(Pc4(\langle Sj_n\rangle))$	$FI_3(1),Pr_3(1),*$	$M(Pc2(\langle Sj_n\rangle))\wedge g2(Pc4, fs)$	(1,2,3,4)

Table 4. Fault and recovery transition definition.

PVF place diagnoses subsystem faults. IF dynamics can also be diagnosed by collecting diagnosed faults temporal information. Table 5 shows the information required for IF dynamics diagnosis:

Fault	*CNT*	*FT*	*T*	T_A	*DF*	*Ps*	*LST*	*OTR*

Table 5. Fault information.

Where *CNT, FT, T,* T_A*, DF, Ps, LST,* and *OTR* have the same meaning as in equations (1), (2), (3), (4), (5), and (6).

The diagnoser builds a table like table 5 for each IF. Moreover, the diagnoser updates the tables each sampling time. Figure 13 shows the updating algorithm.

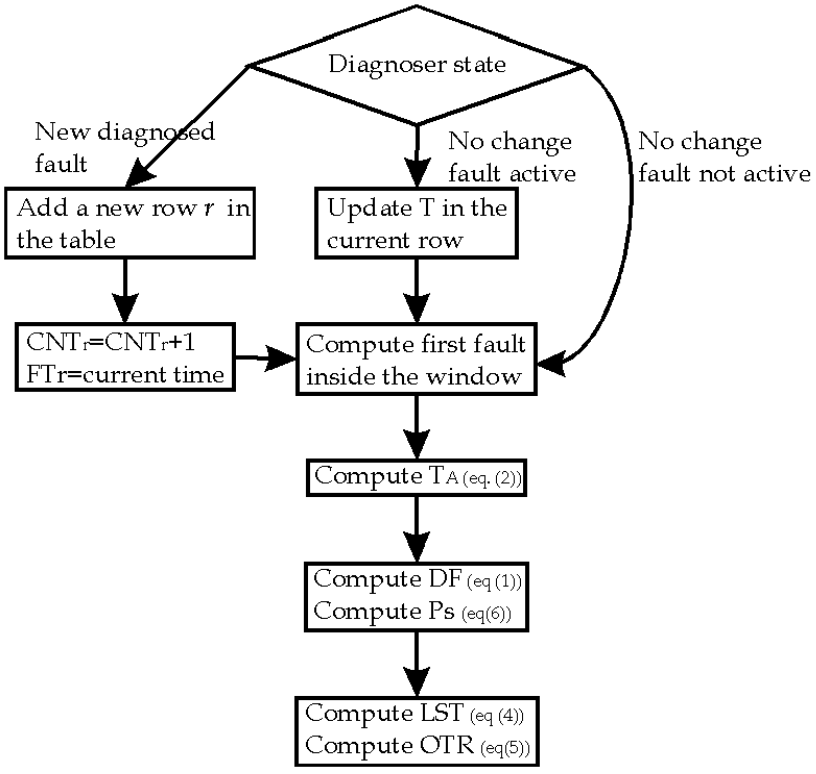


Fig. 13. Updating algorithm.

Therefore, the diagnoser computes *LST* and *OTR* each sampling time. Moreover, each table must be cleared when the faulty device is replaced with a new one.

5. Conclusions

This chapter presents the problem of diagnosing IFs dynamics. We have presented a way of modelling IF dynamics and we have tested it with real data. The density model allows us to compute the best time to repair or substitute the faulty device. This model does not need historical data.
IF dynamics diagnosis has the problem of determine a sliding window size. This problem will be treated in a future work.
We have stated a new definition for IF dynamics and we have proved that the model is able to diagnose the IF dynamics.
We have also presented the integration of IF dynamics diagnosis within a diagnosis technique for discrete event systems based on CPNs. This integration allows the acquisition of temporal information required to compute density and pseudoperiod. Therefore, the diagnosis system will be able to diagnose the IF dynamics.

6. References

Cassandrass, C.G. & Lafortune, S. (1999) *Introduction to Discrete Event Systems.* Kluwer academic publishers.

Contant, O.; Lafortune, S. & Teneketzis, D. (2004) "Diagnosis of intermittent failures". *Discrete event dynamic systems: Theory and applications.* Vol. 14. pp. 171-202. Correcher,

A.; García, E.; Morant, F.; Quiles, E. & Blasco. R. (2004) "Intermittent Failure Diagnosis based on discrete event models". *Proceeding of 7'Th Workshop On Discrete Event Systems WODES04.* pp 151-157. David, R. & Alla, H. (2005) "*Discrete, Continuous, and Hybrid Petri Nets*", Springer-Verlag, Berlin.

García., E.; Rodríguez., L.; Morant., F.; Correcher., A. & Quiles., E. (2008) "Latent Nestling Method: A new fault diagnosis methodology for complex systems" *Proceeding of The 34th Annual Conference of the IEEE Industrial Electronics Society, IECON08.*

Hensen, K. (1997). "*Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use*", Volume 1, Second Edition, Springer Editions.

Jensen, F. (2003). *Electronic component reliability.* John Wiley and Sons Ltd.

Jiang, S.; Kumar, R. & Garcia, H.E. (2003) "Diagnosis of repeated/intermittent failures in Discrete Event Systems". *IEEE transactions on robotics and automation.* Vol. 19. n°2. pp 310-323.

Leemis, L. M. (1995). *Reliability: Probabilistic Models and Statistical Methods.* Prentice Hall.

Rodríguez., L.; García., E.; Morant., F.; Correcher., A. & Quiles, E. (2008). "Application of Latent Nestling Method using Coloured Petri Nets for the Fault Diagnosis in the Wind Turbine Subsets". *Proceedings of ETFA'08,* Hamburg, Germany, 2008

Smolensky, P.; Mozer, M. & D. Rumelhart, D. *Mathematical perspectives on neural networks.* Mahwah, NJ: Lawrence Erlbaum Publishers. 1996

Sorensen, B.A.; Kelly, G.; Sajecki, A. & Sorensen, P.W. (1998) "An analyzer for detecting aging faults in electronic devices". *Proceeding of AUTOTESTCON '94. IEEE Systems Readiness Technology Conference and Proceeding of 'Cost Effective Support Into the Next Century'.* Page(s): 417 -421. Available: http://www.usynaptics.com

Easy-implementable on-line identification method for a first-order system including a time-delay

Satoshi Suzuki and Katsuhisa Furuta
Tokyo Denki University
School of Science and Technology for Future Life
Department of Robotics and Mechatronics
2-2 Kanda-Nishiki-cho Chiyoda-ku
Tokyo 101-8457
Japan

Abstract
This paper proposes a simple yet effective on-line identification method for a first-order system including a time-delay. This method is based on the Laplace transformation in a real number domain and is able to estimate both coefficients of the first-order system and the time-delay simultaneously. An accuracy of the identification was investigated through a simulation. As a result, precise estimation of the method was confirmed compared to an orthodox on-line estimation technique that utilized a bilinear-model. Moreover, a guideline for a tuning of their parameters used in the method is shown. Applying the method to an actual sensor identification, issues under the practical usage were investigated, and the countermeasure was mentioned.

1. Introduction

Many industrial processes involve input time-delays. Control of the system including time-delays is one of important issues. For controlling systems including time-delays, Smith predictor is a practical and popular method (1). Smith proposed an idea of a predictor that compensates a time-delay effect by a feedback loop having the internal time-delay model (2). As other methods, a LQG-based control design (3), a robust stabilization control using LMI (linear matrix inequalities) (4), and an iterative identification and control method (5) were reported. And, as well as control of the system including a time-delay, an identification of processes including a time-delays is also significant. Especially for a product and for a system maintenance, a sensor diagnosis is indispensable for the industrial world, and it boils down to a problem of an identification including the time-delay. As simple solution for this issue, combination of an usual identification method and an elimination of the time-delay effect using a correlation check between the input and the output are often used. However, since this approach has a limit, other various methods have been proposed. Reed *et al.* applied a least-square algorithm to locate the cross-correlation function for the estimation of time-delay from the input / output signals (6). Teng *et al.* tried to estimate the time-delay of a system using the high-order numerator polynomial function (7). Teng's method has an advantage of being able to cope with an inter-sampling behavior, but this method requires sufficient long polynomial structure that can express the unknown time-delay. For dealing with the long polynomial,

large memory and high computational power are required; hence, it is not desired for the implementation.
Additionally, the time-delay often varies with time. At chemical industrial plants, for example, a flow rate and a manipulated variable of tank reactors change by time, and these cause variations of the manipulating time-delay (8). However, from another standpoint, information of the time-delay is often useful for the comprehensive diagnosis. For instance, as the time-delay of the material flow in a chemical process can be estimated using the flow rate and length of pipes, comparison between the physically computed time-delay and the estimated time-delay from a sensor identification can raise reliability of the comprehensive system diagnosis. Therefore, several approaches that can treat unknown time-delay had been also proposed. A delay-dependent robust H_∞ filtering for an uncertain state delay system (9) is effective as well for estimation of the state of a linear system involving time-varying parameters. The amount of the computation is, however, large; this method is inadequate for an on-line estimation because the computation requires solving of a linear matrix inequality. A neural network based approach (10) and an estimation using wavelet (11) are also known; however, their computations require also high-level arithmetic compared to the ability of the embedded computer in commercial products. Product developers have been paying many efforts to implement various functions since a computer resource of a product are restricted to reduce costs. For these practical reasons, a light program using a simple model is more preferable than a precise but complex method requiring much computation power. Additionally, field engineers who have to tune parameters of the products tend not to accept advanced concepts that are difficult to understand intuitively. For instance, a diagnosis using conventionally familiar parameters, like "gain" and "time-delay", is more popular than "singular value" or "Markov parameter". These discussions are summarized to the following requests.

- applicability to a time-varying time-delay
- easy implementation (small memory, low-computation load)
- affinity to engineers in the field

Taking these requests into consideration, a simple yet accurate identification method for the first-order system including a time-delay, *real number Laplace method*, is introduced in this chapter. This method utilizes the Laplace transform in order to separate the time-delay factor from a part of the system transfer function. The benefit is that preliminary information about the time-delay is not required and both target system parameters and the time-delay can be identified simultaneously. Verification of the accuracy, a guideline of the parameter tuning, and an application example are shown in later sections.
This chapter is organized as follows. Section 2 explains the *real number Laplace method*. Section 3 evaluates an accuracy of the method by numerical check, and leads a guideline for the parameters setting through simulation tests. In Section 4, the method is evaluated using actual sensor response data. The discussion and conclusion are mentioned in Sections 5 and 6, respectively.

2. Real Numbers Laplace Identification Method

This section proposes the identification method. Below, $\mathcal{R}$ and $\mathcal{C}$ are a class of real numbers and a class of complex numbers, respectively. The first-order system with a time-delay:

$$G(s) = \frac{K}{1+Ts} \cdot e^{-sL} \tag{1}$$

is considered, where K, T and L are the gain, the time constant, and the time-delay, respectively. Defining the input and the output signals to the system $G(s)$ as $u(t)$ and $y(t)$, and describing their Laplace transformations as $U(s)$ and $Y(s)$, then

$$\frac{Y(s)}{U(s)} = \frac{K}{1+Ts} \cdot e^{-sL}. \tag{2}$$

Calculation of a natural logarithm of right- and left-hand sides of Eq. (2) yields

$$\ln\left(Y(s)/U(s)\right) = \ln K - \ln(1+Ts) - sL. \tag{3}$$

The Laplace transformation of $u(t)$ is described as

$$U(s) = \int_{-\infty}^{\infty} u(t) \cdot e^{-st} dt, \quad s \in \mathcal{C}. \tag{4}$$

As $\mathcal{R} \subset \mathcal{C}$, a real number can be chosen for s at Eq. (2). Therefore, assuming that $\sigma\ (>0, \sigma \in \mathcal{R})$ is sufficiently small number satisfying $T\sigma \simeq 0$, $\ln(1+T\sigma)$ in Eq. (3) can be approximated by the following Taylor expansion.

$$\ln(1+T\sigma) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} (T\sigma)^n \tag{5}$$

Then, Eq. (3) can be transformed as

$$\begin{aligned}
\ln\left(Y(\sigma)/U(\sigma)\right) &= \ln K - \sigma L - \ln(1+T\sigma) \\
&= \ln K - \sigma L - \left(T\sigma - \frac{T^2}{2}\sigma^2 + \frac{T^3}{3}\sigma^3 \right. \\
&\quad \left. - \frac{T^4}{4}\sigma^4 + \frac{T^5}{5}\sigma^5 \cdots \right) \\
&= \ln K - (L+T)\sigma + \frac{T^2}{2}\sigma^2 - \frac{T^3}{3}\sigma^3 \\
&\quad + \frac{T^4}{4}\sigma^4 - \frac{T^5}{5}\sigma^5 \cdots .
\end{aligned} \tag{6}$$

Using the finite numbers of the terms in Eq. (6), the parameters of the system shown in Eq. (1) are identified through the least-square method. If these terms including from the first-order to fourth-order in the Taylor expansion shown in Eq. (6) are used, a regressor vector φ and a parameter vector θ are decided as

$$\varphi(\sigma) = \begin{bmatrix} 1 & -\sigma & \sigma^2 & -\sigma^3 & \sigma^4 \end{bmatrix} \tag{7}$$

$$\theta = \begin{bmatrix} \ln K & L+T & \frac{T^2}{2} & \frac{T^3}{3} & \frac{T^4}{4} \end{bmatrix}^T. \tag{8}$$

Then, Eq. (6) is rewritten as

$$\varphi(\sigma) \cdot \theta = \ln \frac{Y(\sigma)}{U(\sigma)}. \tag{9}$$

Next, preparing M equations by substituting different real-numbers of σ_i (> 0 $i = 1, \cdots, M$) into Eq. (9), those equations are summarized into the following matrix form.

$$\begin{bmatrix} \varphi(\sigma_1) \\ \varphi(\sigma_2) \\ \vdots \\ \varphi(\sigma_M) \end{bmatrix} \cdot \theta = \begin{bmatrix} \ln \frac{Y(\sigma_1)}{U(\sigma_1)} \\ \ln \frac{Y(\sigma_2)}{U(\sigma_2)} \\ \vdots \\ \ln \frac{Y(\sigma_M)}{U(\sigma_M)} \end{bmatrix} \tag{10}$$

$$\Rightarrow \Phi \cdot \theta = \Gamma \tag{11}$$

Finally, an estimation of the parameter vector $\hat{\theta}$ can be obtained by a least-square method as

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \Gamma. \tag{12}$$

In Eq. (10), values of $Y(\sigma_i)$ and $U(\sigma_i)$ are computed using the Laplace transformation with their real numbers, and these values are approximated by summation of finite N terms from the original Laplace transformation shown in Eq. (4). For this computing, the signal $u(t)$ is assumed to satisfy $u(t) = 0$ at $t < 0$, and the Laplace transformation can be approximated as

$$\begin{aligned} U(\sigma) &= \int_{-\infty}^{\infty} u(t) \cdot e^{-\sigma t} dt \\ &\simeq \sum_{i=1}^{N} u[i] \cdot e^{-\sigma \Delta \cdot i} \cdot \Delta, \end{aligned} \tag{13}$$

where Δ is a sampling interval, and $u[i] := u(\Delta \cdot i)$ $(i = 1, \cdots)$ is the sampled data sequence. Concerning Y, the approximated value:

$$Y(\sigma) \simeq \sum_{i=1}^{N} y[i] \cdot e^{-\sigma \Delta \cdot i} \cdot \Delta \tag{14}$$

is used similarly for computation of the identification.
Estimated values of K, T and L are extracted from $\hat{\theta}$, but the elements in $\hat{\theta}$ are redundant as shown in Eq. (8). If those elements are defined as $\hat{\theta} =: [\ \hat{\theta}_1\ \hat{\theta}_2\ \hat{\theta}_3\ \hat{\theta}_4\ \hat{\theta}_5\ ]$, Eq. (8) gives several candidates of those parameters as follows.

$$\hat{K} = e^{\hat{\theta}_1} \tag{15}$$

$$\hat{T}_a = (2\hat{\theta}_3)^{\frac{1}{2}} \tag{16}$$

$$\hat{T}_b = (3\hat{\theta}_4)^{\frac{1}{3}} \tag{17}$$

$$\hat{T}_c = (4\hat{\theta}_5)^{\frac{1}{4}} \tag{18}$$

$$\hat{L}_a = \hat{\theta}_2 - \hat{T}_a \tag{19}$$

$$\hat{L}_b = \hat{\theta}_2 - \hat{T}_b \tag{20}$$

$$\hat{L}_c = \hat{\theta}_2 - \hat{T}_c. \tag{21}$$

In this case, there are three candidates for T and L as $\{\hat{T}_a, \hat{L}_a\} \sim \{\hat{T}_c, \hat{L}_c\}$. How to choose the best combination from these candidates is investigated in the next section.

3. Accuracy verification

3.1 Numerical check of computation accuracy

The *real number Laplace method* utilizes approximations for the actual computation instead of the original mathematically strict descriptions. Hence, an accuracy of the identification depends on the approximation condition. In this section, effects of four factors are investigated: σs for the Laplace transformation, the interval Δ for the integral operation, the order of the Taylor expansion, and a buffer size M for the least square method. In later discussion, a system to be identified is assumed to have intrinsic parameters as $K = 1$ and $T = 0.3$.
First, a relation between the approximation accuracy of $\ln(1 + T\sigma)$ and the decrease ratio of the envelope curve of the integrand in the Laplace transformation was investigated. Table 1 shows the error ratio of the approximated Taylor expansions against the true value $\ln(1 + T\sigma)$ for $\sigma = 0.01$, 0.1, 0.5, 1, 3 and 5, respectively. The smaller percentage is interpreted as higher accurate approximation. It indicates that smaller σ and the higher order give more precise results.

σ	true	$\sim$ 1st term	$\sim$ 2nd	$\sim$ 3rd	$\sim$ 4th
0.01	0.003	-0.150%	0.000%	-0.000%	0.000%
0.1	0.030	-1.493%	0.030%	-0.001%	0.000%
0.5	0.140	-7.325%	0.724%	-0.081%	0.010%
1	0.262	-14.345%	2.807%	-0.624%	0.148%
3	0.642	-40.219%	22.880%	-14.979%	10.575%
5	0.916	-63.704%	59.074%	-63.704%	74.421%

Table 1. Error ratios of Taylor expansions approximated using different σ

Next, an effect of the finite summation instead of the infinite integration was investigated. Table 2 shows values of the envelope function $e^{-\sigma T_F}$, that appears in the integrand shown in Eq. (4), against different combinations of σ and T_F. The smaller value shows that the rounding error at the end point of the integral computation is small; hence, the approximation is close to the true value.

σ	$T_F = 3$	$T_F = 5$	$T_F = 10$	$T_F = 20$	$T_F = 30$
0.01	0.9704	0.9512	0.9048	0.8187	0.7408
0.1	0.7408	0.6065	0.3679	0.1353	0.0498
0.5	0.2231	0.0821	0.0067	0.0000	0.0000
1	0.0498	0.0067	0.0000	0.0000	0.0000
3	0.0001	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000

Table 2. Attenuation rates of the envelope function $e^{-\sigma T_f}$

Table 1 shows that the approximation with the smaller σ is better even if the order of the Taylor expansion is small. Meanwhile, table 2 indicates that the summation computation by the small σ cannot cover the integrated area of the original integral calculation sufficiently even if the integral time is long. That is, there is a trade-off in choice of σ due to the truncation error at the finite-time approximation of the Laplace transformation. Consideration of these tables suggests the following guide-line for the selection of σ and T_F:

- $\sigma = 0.5$ is adequate for the real number Laplace transform computation.
- More than third-order approximation with small σ that is less than 0.5 is necessary for the Taylor expansion approximation[1].
- $T_F = 20$ $[s]$ appears to be adequate as the integral time[2] for the approximation.

Aforementioned investigation surmises the following remarks.

Remarks

- σ has to be chosen as small as possible in order to satisfy $T\sigma \simeq 0$ for good approximation of the logarithm function in the Taylor expansion. Furthermore, the measured data has to be long so as to attenuate an integrand of the Laplace transformation.
- For a target system including a large time-constant, it is necessary to choose small σ or to extend the integral interval by same reason of the above item.
- High order polynomial in the Taylor expansion gives more accurate approximation; however, this increases the size of the regressor and the amount of the computation.

3.2 Simulation verification using test signal

Accuracy of the *real number Laplace method* was investigated using the sample response data from a virtual target system when an input signal of a pulse wave was added to the system. The amplitude and the cyclic period of the pulse input were chosen as 0.5 and 2 seconds. It was assumed that the target system had a gain of $K = 1$, a time-constant of $T = 0.3$, and a time-delay of $L = 0.1$ $[s]$. Parameters for the identification were chosen as follows based on the remarks mentioned in Section 3.1.

- the range of real numbers for the Laplace transformation: $\sigma = 0.5 \sim 0.7$ at 0.01 interval
- the number of points for the least-square method: $M = 20$
- the number of points for numerical integral: $N = 2000$
- sampling time: $\Delta = 10$ $[ms]$

These conditions leads $T_F = \Delta t \times N = 20$ $[s]$. First, effects of the finite truncation of the Taylor expansion for $\ln(1 + T\sigma)$ was inspected by changing the "Maximum Order of truncation Terms"(MOT) in the Taylor series. Table 3 shows the results. Note that the number of candidates of pair $\{T, L\}$ increases as the truncation order increases because of a redundancy in their parameter vectors. The result shows that the identified parameters were fairly close to their true values. Not only the gain and time-constant but also the time-delay could be estimated. Table 3 shows also that the first pair $\{\hat{T}_a, \hat{L}_a\}$ of the higher order truncation leads more accurate result.
Last check is about an effect of the integral interval T_F of the Laplace transformation approximation. Results for two cases of $T_F = 10$ and $T_F = 40$ are summarized in Tables 4 and 5, respectively. The identified parameters for $T_F = 10$ (Table 4) are wholly inferior to the case of $T_F = 20$ (Table 3). This is, of course, because a finite time interval computation was used instead of an infinite interval in the integral computation. Especially, the case of MOT$= 4$ gave wrong result since pairs of $\{\hat{T}_b, \hat{L}_b\}$ and $\{\hat{T}_c, \hat{L}_c\}$ were complex numbers. On the other hand, the case of $T_F = 40$, as shown in Table 5, improved the results slightly compared to the case of $T_F = 20$. The 40 seconds appears, however, excess for T_F since twice-time computing is needed. Hence, it appears that $T_F = 20$ is adequate to the present sample system.

[1] The percentage of the approximation error is small at -0.0809 [%] in this case. (From Table 1)
[2] The remainder of the integrated area is tiny at 0.0000 [%]. (From Table 2)

MOT	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{T}_c$
4	0.9997	0.2938	0.2696	0.2103
3	0.9996	0.2903	0.2525	-
2	0.9985	0.2551	-	-
true	$K = 1$	$T = 0.3$		

MOT	$\hat{L}_a$	$\hat{L}_b$	$\hat{L}_c$
4	0.1110	0.1353	0.1946
3	0.1142	0.1519	-
2	0.1436	-	-
true	$L = 0.1$		

Table 3. Identification results using test signal with $T_F = 20$

MOT	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{T}_c$
4	0.9974	0.2080	0.1667 + 0.2888i	0.2772+0.2772i
3	0.9981	0.2612	0.1717	-
2	0.9977	0.2494	-	-

MOT	$\hat{L}_a$	$\hat{L}_b$	$\hat{L}_c$
4	0.1856	0.2268 - 0.2888i	0.1164-0.2772i
3	0.1372	0.2268	-
2	0.1473	-	-

Table 4. Identification results using test signal with $T_F = 10$ $(M = 1000)$

3.3 Comparison with a bilinear-model method

In this section, the *real number Laplace method* is compared to other conventional identification method able to be implemented on-line to show the effectiveness. Here, a recursive least-square method with a discrete model was chosen as the conventional method. This method utilizes a bilinear transformation, and is termed *bilinear-model method* simply later. Details of the *bilinear-model method* are described in Appendix. A. Same test signal that includes no time-delay was applied to the *bilinear-model method* . Figure 1 indicates five cases of transitions of the estimated values of time-constant T and gain K. Each graph was obtained by changing the DF. The DF were specified as $1, 5, 10, 20$ and 40, and these values correspond to sampling intervals 0.01, 0.05, 0.1, 0.2 and 0.4 $[s]$, respectively. From the figure, $DF = 5$ $(dt = 0.05)$ appears a best condition since the identified parameters were close to their true values $(T = 0.3, K = 1.0)$. Conversely, as shown in this analysis, the recursive type of identification methods requires selection of the adequate DF. The DF reportedly has to be chosen considering the time-constant of the target system. For the sensor diagnosis, however, the time-constant itself changes across the ages; hence, this is one of the drawbacks. In contrast, the proposed *real number Laplace method* is applicable to a change of a time-constant in the target system.

Next, the *bilinear-model method* with $DF = 5$, that was the best tune for the decimation, was applied to the test signal including a time-delay. Simulation tests were executed against different values of the time-delay. The results are illustrated in Fig. 2. The identified parameters of K and T became larger than the true values as the time-delay increased. Since the *bilinear-model method* does not have an ability of the time-delay estimation, other time-delay estimator, such as a correlation analysis between the input and output signal, is required. However, if the

MOT	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{T}_c$
4	0.9998	0.2986	0.2883	0.2544
3	0.9996	0.2911	0.2545	-
2	0.9985	0.2552	-	-

MOT	$\hat{L}_a$	$\hat{L}_b$	$\hat{L}_c$
4	0.1068	0.1172	0.1510
3	0.1135	0.1501	-
2	0.1436	-	-

Table 5. Identification results using test signal with $T_F = 40$ $(M = 4000)$

time-delay effect is removed insufficiently, the estimation becomes worse. Figure 2 highlights this weak point of the *bilinear-model method* . Conversely, from an insufficient results given by the *bilinear-model method* , it can be concluded that the proposed *real number Laplace method* is superior in terms of an estimation accuracy and of robustness against a time-delay.

4. Application-level verification

In this section, the proposed identification method was verified using an actual measured data. Here, a sensor in the engine control system was chosen for an example. For not only the performance retention but also an environmental conservation, a sensor is significant for the engine control. Response anomaly of a degraded sensor induces a change of the time-constant or the inaccurate gain; hence, it is relatively easy to determine the likelihood of the degrading by checking the step response in case of an unit testing. Rotational speed of the engine, however, varies awfully depending on various factors such as driver's demand and the load condition, and the time-delay also varies (12). Since the proposed *real number Laplace method* can be applied to unknown time-delay, this example is adequate for the verification.

4.1 O_2 sensor in an engine system

The O_2 sensor that is treated here monitors an oxygen density in combustion gas near the engine cylinder at an exhaust pipe. Air-fuel ratio (AFR) is computed based on the measured O_2 density, and the input signal for the sensor identification is assumed to be a AFR of the fuel gas. The AFR of the fuel gas varies mainly depending on an intake air mass and the amount of fuel consumption from the injector. All of the injected fuel, however, does not evaporate into air, and portion of the fuel adheres to a wall of the pipe. Further, the gas is transported via the four-stroke cycle: Intake, Compression, Combustion, and Exhaust. Because of these processing, the delay from the injection to the detection at the O_2 sensor is generated and changes dynamically.

In order to apply the proposed method to the O_2 sensor identification, variation of the fuel gas AFR was chosen for the input signal $u(t)$, and the other AFR of the exhaust gas monitored by the sensor was used as the output signal $y(t)$. The transmission lag at the pipe and the engine cyclic processing were treated as one delay element. Supposed that the dynamics can be modeled with a first-order system, the whole of the transfer function can be treated as Eq. (1). In an engine control, small additional operation is permitted during the constant speeds. So-called "active excitation", that changes the input AFR by small amplitude square wave, is executed in a product car. Referring an actual case, a square wave whose amplitude and cyclic

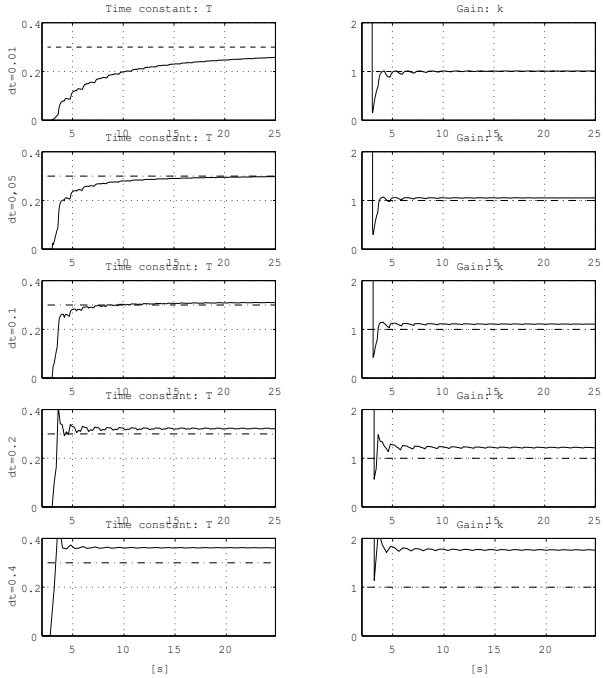


Fig. 1. Transitions of identified parameters by the bilinear-model method (dt is the sampling interval after decimation.) Dashed lines denote the true values $\bar{T} = 0.3$ and $\bar{k} = 1$.

period were $\pm 0.5[AFR]$ and $1.32[s]$ was used. Parameters for the identification were chosen as follows based on aforementioned guideline.

- the order to approximate the Taylor expansion: third order
- the range of real numbers for the Laplace transformation: $\sigma = 0.5 \sim 0.7$ at 0.01 interval
- the number of points for the least-square method: $M = 20$
- the number of points for numerical integral: $N = 2400$

Sampling time $\Delta = 8.2$ $[ms]$ was decided by the measurement condition. These conditions leads $T_F = \Delta \times N \simeq 20$ $[s]$.

4.2 Verification using actual data

The data was obtained using a bench test with a four-cycle 2.3 liter engine. At first, the data measured at a speed of constant 80 $[km/h]$ was filtered through a LPF $1/(1+10s)$ to eliminate the bias, and the step response crossing the zero-level was obtained. The identification result is shown at the case-1 in the Table 6. Contrary to the present authors' expectation, the identified values were complex numbers and the correct ones were not obtained. To find the reason, values of terms in equations were checked. As a result, it was confirmed that the integration value in the transformation (13) was smaller than other case that was computed using an ideal wave form. The zero-crossing signal tends to be affected by the noise; hence, the actual signal appears ill-conditioned involving small S/N ratio. To avoid this issue, both the input and the

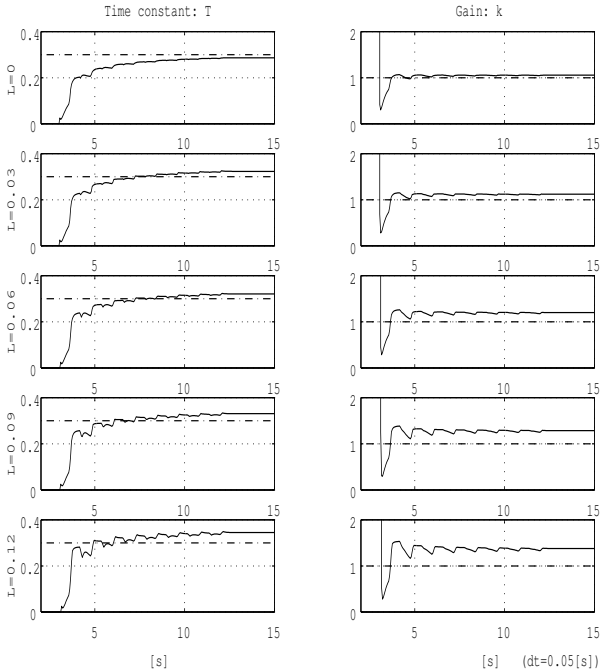


Fig. 2. Transitions of identified parameters by bilinear-model method : Comparison against different time-delay.

output signals were modified by adding 0.5 to them so as to be greater than 0, as shown in Fig. 4. The result of an identification using the modified signals is shown at the case-2 in Table 6. The identified gain $\hat{K}$ is close to the true value, and real number time-constant value $\hat{T}_a$ was obtained. Accuracy of the identification was improved, but still insufficient.

Considering of a mathematical property of Eq. (13) again, it was surmised that an initial value of the integration period affected strongly the computation since a value of the envelope curve $e^{-\sigma t}$ is large around $t \simeq 0$. Upper graph in Fig. 5 is an enlargement of the initial rising of the signals shown in Fig. 4. As indicated with an arrow in the graph, a residual vibration of a previous step-response remained in the interval of $t = 8.19 \sim 8.5$, and it was found that $y \simeq 0$ was not satisfied at the same period. To remove this adverse effect, the integration period was modified so as to be satisfied with $u \simeq 0$ and $y \simeq 0$. In short, the data was shifted in time direction by 30 sampling points to change the initial time from $t = 8.13$ to $t = 8.44$. The modified signals are shown at the lower graph in Fig. 5. Using the modified signals, the identification result was improved as written at the case-3 in Table 6. The identified gain $\hat{K}$, the time-constant $\hat{T}_a$, and the time-delay $\hat{L}_a$ were sufficiently close to their true values. Though the second candidates of the time-constant and the time-delay were complex numbers, the identification-method worked well because it was already confirmed that the proposed method lead accurate values in the first candidate term from the redundant ones as mentioned in Section 3.2.

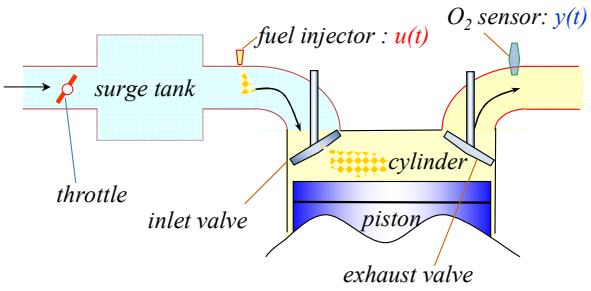


Fig. 3. O_2 sensor in the engine system

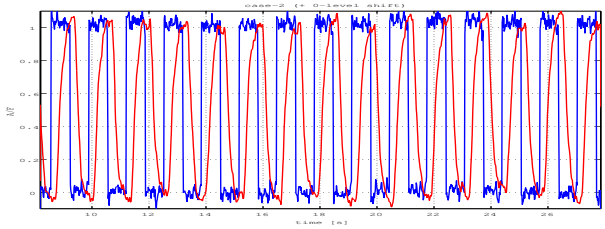


Fig. 4. Fluctuation component of the excitation (blue), and the modified sensor output (red).

For confirmation, using the original measured input signal AFR, simulation responses were computed through two models: a model with identified parameters (of the case-3 in Table 6), and a model with nominal parameters (of the "true" in the same table). The resultant wave-forms are shown in Fig. 6. The original raw experimental signal was also drawn in each graph. The graphs illustrate that the time-delay was estimated correctly by comparing those timings of the rising response. The amplitude of the response computed with the identified parameters appears slightly small. However, slant angles of the rising curve of both response data are same; hence, it can be accepted that the estimated time-constant was correct.

4.3 Investigation of effect by the excitation signal condition

Aforementioned section showed that an accuracy of the identification became low if the starting point was not chosen adequately or a shifting of the input/output signal to the positive-value zone was insufficient. Since it is better to know degree of the modification as a guideline for the implementation, the sensitivity analysis is mentioned in this section. Normative response signal was computed through the simulation by adding a rectangular excitation signal (amplitude=1, period=1.32 [s], sampling interval=8.2 [ms]) to the transfer function whose parameters were same as the former system ($K = 1.0, T = 0.18, L = 0.17$). Next, using test signals that were obtained by changing the delay of the excitation timing b $[s]$, and the vertical shift from the nominal response signal a, as shown in Fig. 7, the difference of the identified results were investigated. The a is a ratio of the part that is below a zero-level, and the signal varies from 0 to 1 when $a = 0$.

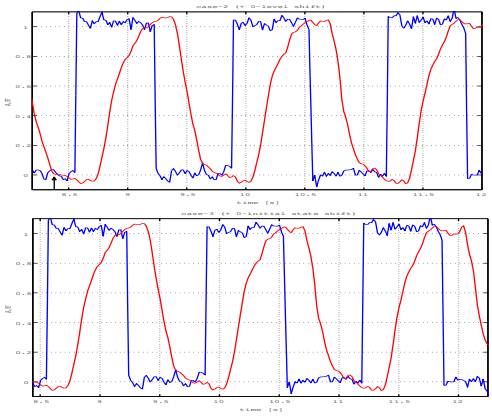


Fig. 5. Magnified graph of input and output signals (upper), and the signals modified of the initial points (lower)

case	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{L}_a$	$\hat{L}_b$	
1	c.n.	c.n.	c.n.	c.n.	c.n.	
2	0.94	0.28	c.n.	-0.0045	c.n.	c.n. = complex number
3	0.94	0.16	c.n.	0.20	c.n.	
true	$K = 1.0$	$T = 0.18$		$L = 0.17$		

Table 6. Comparison of the identified results

First, b was fixed as $b = 0$, and the identification accuracies against different ratios of positive zone in vertical direction were computed. The results are summarized in Table 7. Though the gain K was estimated correctly for $a = 0 \sim 0.5$, the time-constant T and time-delay L became worse gradually. For the a over 0.5, their results rapidly got worse. These results show that the signal including much positive values is better for the correct identification. It appears that the tolerance is $a = 0.0 \sim 0.2$.
Next, a was fixed as $a = 0$, the identification accuracies against different delays at start of the excitation were investigated similarly. The results is Table 8. Their gains were identified with similar accuracy to the former case. The accuracies of the time-constant and the time-delay became worse as b increases. The second candidate of the estimation became a complex number when $b > 0.5$. Concerning the starting delay, it appears that $b = 0.0 \sim 0.2$ is acceptable. As a conclusion, it is preferable that the signal for the *real number Laplace method* identification are modified by considering the following remarks.

- The drift and offset are removed from the output signal so as to make the signal zero at the input of zero.
- Both the input signal and the output signal include only positive values; in short, these signals do not contain zero-crossing.
- Identification is started from just after the rising up of the input signal.

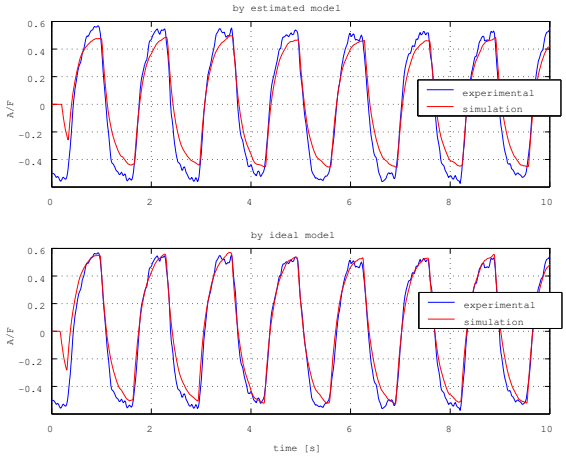


Fig. 6. Simulated responses computed by the identified model (upper) and by the ideal model (lower)

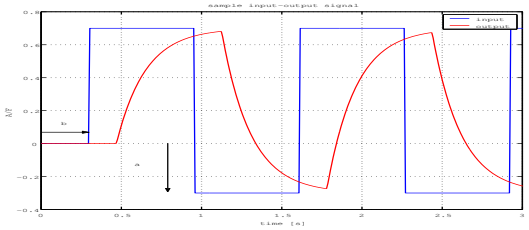


Fig. 7. Test signal for evaluation

5. Discussion

As the *real number Laplace method* includes a numerical integral computation, this method appears seemingly to require much memory at the implementation, but it is not true. This worry will be removed by the following two artifices.

The first artifice is a preliminary computation of the constant terms of the equations. Generally, in case of an on-line identification methods based on the least-square computation, the regressor vector includes time-varying variables that come from the measured signals; hence, it is necessary to compute them on-line. And as shown in Eq. (12), the inverse matrix computation of $(\Phi^T\Phi)^{-1}$ is included. Thus, the normal least-square-based method requires an on-line inverse matrix computation. This computation requires a high level of the arithmetic operation; hence, it is not welcomed for the computer device of a consumer-level product. Meanwhile, in case of the *real number Laplace method*, Φ is a constant matrix as

$\Phi := [\ \varphi(\sigma_1)^T \vdots \varphi(\sigma_M)^T\ ]^T$, where $\varphi(\sigma_i)$ is a constant vector given by Eq. (7), and this computation can be finished offline. Thus, troublesome inverse matrix computation can be replaced by simple summation and multiplication.

a	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{L}_a$	$\hat{L}_b$
0	1.000	0.176	0.155	0.174	0.195
0.1	1.000	0.180	0.170	0.170	0.181
0.2	1.000	0.187	0.188	0.165	0.163
0.3	1.000	0.197	0.215	0.155	0.138
0.4	1.000	0.219	0.257	0.136	0.099
0.5	1.000	0.288	0.353	0.078	0.014
0.6	16.91	7.217	3.559	8.042	11.70
true	$K = 1.0$	$T = 0.18$		$L = 0.17$	

Table 7. Result of identifications when a ratio of the positive-value zone of an input signal was changed

b	$\hat{K}$	$\hat{T}_a$	$\hat{T}_b$	$\hat{L}_a$	$\hat{L}_b$
0	1.000	0.176	0.155	0.174	0.195
0.1	1.000	0.175	0.154	0.174	0.195
0.2	1.000	0.174	0.147	0.176	0.202
0.3	1.000	0.170	0.130	0.179	0.219
0.4	1.000	0.163	0.080	0.185	0.268
0.5	1.000	0.151	c.n.	0.196	c.n.
0.6	1.000	0.132	c.n.	0.213	c.n.
true	$K = 1.0$	$T = 0.18$		$L = 0.17$	

Table 8. Result of identification against different starting delay

The second artifice is an on-line accumulation of the input/output data instead of the original integral computations. For the implementation to a computer, $2M$ buffer memory $\tilde{U}(\sigma_1) \sim \tilde{U}(\sigma_M)$ and $\tilde{Y}(\sigma_1) \sim \tilde{Y}(\sigma_M)$ for $\sigma_1 \sim \sigma_M$ are prepared, then recurrence equations:

$$\begin{aligned}
\tilde{U}(\sigma_1) &= \tilde{U}(\sigma_1) + u[i] \cdot e^{-\sigma_1 \Delta \cdot i} \cdot \Delta \\
&\vdots \\
\tilde{U}(\sigma_M) &= \tilde{U}(\sigma_M) + u[i] \cdot e^{-\sigma_M \Delta \cdot i} \cdot \Delta \\
\tilde{Y}(\sigma_1) &= \tilde{Y}(\sigma_1) + y[i] \cdot e^{-\sigma_1 \Delta \cdot i} \cdot \Delta \\
&\vdots \\
\tilde{Y}(\sigma_M) &= \tilde{Y}(\sigma_M) + y[i] \cdot e^{-\sigma_M \Delta \cdot i} \cdot \Delta
\end{aligned}$$

are used for computing Eqs. (13) and (14). This technique is useful to reduce memory on the computer architecture.

6. Conclusion

In this chapter, a practical, accurate yet simple identification method, termed as *real number Laplace method* , to estimate parameters of a first-order system including a time-delay was proposed. The key point is restriction of a domain of the Laplace transformation to real numbers.

The method can estimate the system parameters and the time-delay simultaneously on-line. The method does not require *a priori* information about the time-delay, and can be applied to a system including arbitrary time-delay. Comparison with other method (based on recursive least-square identification with a bilinear transformed impulse transfer model) proved that the *real number Laplace method* could identify the system parameters much accurately. And a relation between an accuracy of the identified results and the tuning conditions were investigated, and a guideline for the tuning was summarized. Moreover, the *real number Laplace method* was applied to an actual response data of an engine sensor, and the sensor dynamics and the unknown time-delay could be estimated with sufficient accuracy. Further, a guideline of a preliminary modification of the measured data was shown, and remarks for better identification on the actual system were summarized.

A. Recursive least-square identification using an impulse transfer function with a bilinear transformation

To a first-order system with a time-constant T and a gain k:

$$G(s) = \frac{k}{1+Ts}, \tag{22}$$

applying the bilinear transformation:

$$s = \frac{2}{\Delta}\frac{1-z^{-1}}{1+z^{-1}} \tag{23}$$

yields the impulse transfer function as

$$G(z) = \frac{\Delta k \cdot z^{-1} + \Delta k}{(\Delta - 2T)\cdot z^{-1} + (\Delta + 2T)} \tag{24}$$

$$=: \frac{z^{-1}+1}{b_2 z^{-1} + b_1} \tag{25}$$

$$b_2 := \frac{\Delta - 2T}{\Delta k} \tag{26}$$

$$b_1 := \frac{\Delta + 2T}{\Delta k}, \tag{27}$$

where z^{-1} is a time-shift operator. T and k can be derived from Eqs. (26) and (27) as

$$T = \frac{b_1 - b_2}{b_1 + b_2}\cdot\frac{\Delta}{2} \tag{28}$$

$$k = \frac{2}{b_1 + b_2}. \tag{29}$$

Once b_1 and b_2 are obtained, T and k can be computed from Eqs. (28) and (29).
Next, the measured input and output data are described as $\bar{u}(t)$ and $\bar{y}(t)$ respectively, and the inverse Z-transformation is applied to Eq. (25), then

$$b_2 \cdot \bar{y}(t-1) + b_1 \cdot \bar{y}(t) = \bar{u}(t-1) + \bar{u}(t) \tag{30}$$

is obtained. Furthermore, Eq. (30) is transformed as

$$\begin{bmatrix} \bar{u}(t)+\bar{u}(t-1) \\ \bar{u}(t-1)+\bar{u}(t-2) \\ \cdots \\ \bar{u}(2)+\bar{u}(1) \end{bmatrix} = \begin{bmatrix} \bar{y}(t) & \bar{y}(t-1) \\ \bar{y}(t-1) & \bar{y}(t-2) \\ \cdots & \cdots \\ \bar{y}(2) & \bar{y}(1) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{31}$$

Using replacements as

$$y(t) = \bar{u}(t)+\bar{u}(t-1) \tag{32}$$

$$\varphi(t) = [\,\bar{y}(t),\,\bar{y}(t-1)\,]^T, \tag{33}$$

Eq. (31) is transformed as

$$\begin{bmatrix} y(t) \\ y(t-1) \\ \cdots \\ y(2) \end{bmatrix} = \begin{bmatrix} \varphi^T(t) \\ \varphi^T(t-1) \\ \vdots \\ \varphi^T(2) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{34}$$

$$\Rightarrow \mathbf{y} = \Phi \cdot \theta. \tag{35}$$

Last step is an estimation of vector term Φ that includes parameters b_1 and b_2, and is executed by the recursive least-square method. Finally, $T(t)$ and $k(t)$ can be computed from Eqs. (28) and (29) on-line.

7. References

[1] R. C. Miall, D. J. Weir, D. M. Wolpert and J. F. Stein. Is the Cerebellum a Smith Predictor? *Journal of Motor Behavior*, 25(3):203–216, 1993.

[2] O. J. M. Smith. A Controller to Overcome Dead Time. *Transaction ISA*, 6(2):28–33, 1959.

[3] P. V. D. Hof and R. J. P. Schrame. Identification and Control - Closed-loop Issues. *Automatica*, 31(12):1751–1770, 1995.

[4] X. Li and C. E. de Souza. Delay-dependent Robust Stability and Stabilization of Uncertain Linear Delay Systems: A Linear Matrix Inequality Approach. *IEEE Transaction on Automatic Control*, 42(8):1144–1148, 1997.

[5] P. Albertos and A. Sala. *Iterative Identification and Control*. Springer, 2002.

[6] F. Reed, P. Feintuch, and N. Bershad. Time-delay estimation using the LMS adaptive filter-static behavior. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 29(3):561–576, 1981.

[7] F. C. Teng, and H. R. Sirisena. Self-tuning PID controllers for dead time process. *IEEE Trans. Industrial Electronics*, 35(1):119–125, 1988.

[8] A. B. Bulsari (ed.). *Neural network for Chemical Engineering*. Elsevier Science, Amsterdam, Holland, 1995.

[9] R. Yang, X. Xu, and C. Zhang. Delay-dependent robust H_∞ filtering for uncertain state delayed system. in Proc. of *the IFAC 15th Triennial World Congress*, in CD-ROM, Barcelona, Spain, 2002.

[10] Y. Tan, C.-Y. Su, and N. Karim. Neural network based time-delay estimation for nonlinear dynamic systems. in Proc. of *the IFAC 15th Triennial World Congress*, in CD-ROM, Barcelona, Spain, 2002.

[11] T. Hamada, and K. Nakano. Wavelet-based Underdetermined Blind Source Separation of Speech Mixtures. in Proc. of *the ICCAS 2007*, in CD-ROM, Seoul, Korea, 2007.

[12] L. Guzzella and C. H. Onder. Introduction to Modeling and Control of Internal Combustion Engine Systems. *Springer-Verlag*, Berlin, 2004.

[13] S. Suzuki, and K. Furuta. Real Number Laplace Transformation-based Identification for First-order System including Time-delay, in Proc. of *the 2008 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2008)*, Hamburg, Germany; 143–149, 2008.