Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions

Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions

Edited by
XiaoQi Chen, Y.Q. Chen and J.G. Chase

In-Tech
intechweb.org

Published by In-Teh

In-Teh is Croatian branch of I-Tech Education and Publishing KG, Vienna, Austria.

www.in-teh.org
Additional copies can be obtained from:
publication@ars-journal.com

First published May 2009
Printed in Croatia

Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions,
Edited by XiaoQi Chen, Y.Q. Chen and J.G. Chase
p. cm.
ISBN 978-953-307-001-8

Preface

Since the introduction of the first industrial robot UNIMATE in a General Motors automobile factory in New Jersey in 1961, robots have gained stronger and stronger foothold in the industry. In the meantime, robotics research has been expanding from fix based robots to mobile robots at a stunning pace. There have been significant milestones that are worth noting in recent decades. Examples are the octopus-like Tentacle Arm developed by Marvin Minsky in 1968, the Stanford Cart crossing a chair-filled room without human assistance in 1979, and most recently, humanoid robots developed by Honda.

Despite rapid technological developments and extensive research efforts in mobility, perception, navigation and control, mobile robots still fare badly in comparison with human abilities. For example, in physical interactions with subjects and objects in an operational environment, a human being can easily relies on his/her intuitively force-based servoing to accomplish contact tasks, handling and processing materials and interacting with people safely and precisely. The intuitiveness, learning ability and contextual knowledge, which are natural part of human instincts, are hard to come by for robots.

Hardware technologies including materials, novel actuator and mechanisms, and drives have been a severe limiting factor in mobile robots emulating human beings. The well-known Honda humanoid robot Asimo has a self weight of 52 Kg, and a lifting capacity of 1 Kg with two hands. In the 2008 Olympics held in Beijing, the women 53 Kg weightlifting gold medal went to a Thai girl who lifted 126 Kg in Clean and Jerk. Her lifting capacity to self weight ratio is about 2.4 while Asimo has the ratio of about 0.02. The ratio for a non-athletic adult would be about 0.4, much lower than an Olympic champion, but a magnitude higher than Asimo.

The above observations simply highlight the monumental works and challenges ahead when researchers aspire to turn mobile robots to greater benefits to humankinds. This book is by no means to address all the issues associated mobile robots, but reports current states of some challenging research projects in mobile robotics ranging from land, humanoid, underwater, aerial robots, to rehabilitation.

Chapter 1 provides an overview on mobile robotics. It proposes a functional model of generalized robots by drawing the parallelism between robots and humans. It also points to service robots being a disruptive technology in decades to come.

Chapter 2 introduces a field robot using the rotated-claw wheel that has strong capacity of climbing obstacles. The rotated-claw wheel overcomes the disadvantages of conventional mobile robot wheels, and provides a better solution for field and planetary robots.

Chapter 3 presents a mobile wheeled robot with step climbing capabilities using parallel individual axels. The wheel sets are designed to revolve in any direction independent of the rotation of the wheels, satisfying the primary requirements for the robot traversing rough terrains and climbing stairs.

Chapter 4 reviews some major efforts made over the past 20 years in the field of cable-climbing mechanism design to provide a basis for future developments in this field. One application highlighted is power line inspection in the power industry.

Chapter 5 proposes a multi-sensing fusion system to mimic the powerful sensing and navigation abilities of a cockroach. The paper proposes a multi-sensor fusion system, and a distributed multi-CAN bus-mastering system based on Field Programmable Gate Array (FPGA) and Advanced RISC Machine (ARM) microprocessor.

Chapter 6 highlights some characteristics observed from human abilities in performing both knowledge-centric activities and skill-centric activities. The observations can be utilised to guide the design of a humanoid robot's body, brain and mind.

Chapter 7 reports an Autonomous Underwater Vehicle (AUV) prototype that had been developed recently at the University of Canterbury. The AUV has been specially designed for shallow water tasks, such as inspecting and cleaning sea chests of ships.

Chapter 8 establishes an approach to solve the full 3D Simultaneous Localisation and Mapping (SLAM) problem, applied to an underwater environment. A new measurement system has been designed for large area's globally-consistent SLAM: buoys for long-range estimation, and camera for short-range estimation and map building.

Chapter 9 addresses flight dynamics modelling and method of model validation using an on-board instrumentation system. The validation process involves in-flight measurement of all parameters as well as wind speed detected by in-house built air-speed sensor, aiming for an accurate flight simulation system for auto-pilot development and preliminary design of Unmanned Aerial Vehicles (UAVs).

Chapter 10 describes a numerical procedure for optimal sensor-motion scheduling of diffusion systems for parameter estimation. The work introduces the optimal actuation framework for parameter identification in distributed parameter systems. This contribution lays a rigorous foundation for real-time estimation for a class of cyber-physical systems.

Chapter 11 presents some heuristics for constructing the mobile collector collection route. Control schemes for coordinating multiple collectors are designed efficiently to maximize the performance.

Chapter 12 discusses the development of the augmented reality (AR) for human-robot collaboration (HRC) system from concept and background through the design of the necessary set of interfaces required to enhance human-robot interaction. The AR-HRC system gives the user the feeling of working in a collaborative human-robot team rather than the feeling of the robot being a tool, as a typical teleoperation interface provides.

Chapter 13 details a set of classifications of indoor localization techniques. The classifications presented provide a compact form of overview on WSN-based indoor localizations. The chapter further introduces server-based and range-based localization systems that can be used for the indoor service robot.

Chapter 14 proposes a wearable soft parallel robot for ankle joint rehabilitation after carefully studying the complexities of human ankle joint and its motions. The proposed device is an improvement over existing robot in terms of simplicity, rigidity and payload performance.

The collection of the above research works is suitable for robotics researchers, engineers and academics. It can be used as case studies in advanced robotics courses. Some chapters serve as tutorial references in nature, while other chapters allow methods and results to be readily reproducible.

The editor is indebted to all authors for being forthcoming in sharing their knowledge and research work in such an open access platform. Their valuable contributions to this book are gratefully acknowledged. Special thanks to Aleksandar Lazinica the Editor-in-Chief of IN-TECH for his support in publishing this book, his passion and enthusiasm in bringing scientific knowledge to a larger public.

X.Q. Chen[1], Y.Q. Chen[2], and J.G. Chase[1], Editors
[1] Department of Mechanical Engineering
University of Canterbury, New Zealand
[2] Center for Self-Organizing and Intelligent Systems (CSOIS)
Utah State University, USA
xiaoqi.chen@canterbury.ac.nz

Contents

Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions

Mobiles Robots – Past Present and Future

X.Q. Chen[1], Y.Q. Chen[2], and J.G. Chase[1]
[1] *Department of Mechanical Engineering*
University of Canterbury, New Zealand
[2] *Center for Self-Organizing and Intelligent Systems (CSOIS)*
Utah State University, USA

1. Introduction

Since their introduction in factories in 1961, robots have evolved to achieve more and more elaborate tasks. The industrial robots now account for a 5 billion dollars market. Positioned along the assembly line, a robotic manipulator can perform tedious and repetitive tasks such as welding, painting, moving or cutting with immense speed and incredible accuracy. As an example, their use in the automotive industry has drastically cut the time it takes to assemble a vehicle.

Since the introduction of the first industrial robot UNIMATE online in a General Motors automobile factory in New Jersey in 1961, robots have gained stronger and stronger foothold in the industry. Several milestones are worth noting since then:

The first artificial robotic arm to be controlled by a computer was designed. The Rancho Arm was designed as a tool for the handicapped and its six joints gave it the flexibility of a human arm.

DENDRAL was the first expert system or program designed to execute the accumulated knowledge of subject experts.

1968 - The octopus-like Tentacle Arm was developed by Marvin Minsky.

1969 - The Stanford Arm was the first electrically powered, computer-controlled robot arm.

1970 - Shakey was introduced as the first mobile robot controlled by artificial intelligence. It was produced by SRI International.

1974 - A robotic arm (the Silver Arm) that performed small-parts assembly using feedback from touch and pressure sensors was designed.

1979 - The Stanford Cart crossed a chair-filled room without human assistance. The cart had a TV camera mounted on a rail which took pictures from multiple angles and relayed them to a computer. The computer analyzed the distance between the cart and the obstacles.

It is no surprising that in face of diverse configurations, functions, applications, and autonomy there is no agreeable universal definition of “Robot”. The well-known definition of Robot from the Robot Institute of America (RIA) is that a robot is

"A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks".

It at best describes industrial robots and applications – not all robotic applications are associated with "move things". "Programmed motions" may have to be augmented for mobile robots that often decide their motion/part based on their situational awareness. A more inspiring and general definition can be found in Webster. According to Webster a robot is:

"An automatic device that performs functions normally ascribed to humans or a machine in the form of a human."

The Webster definition broadly covers robotic tasks and functions beyond moving things as in RIA definition. Being a subset of robots, one may consider that defining mobile robot would be easier and more accurate. Indeed, Wikipedia has this definition:

"A Mobile Robot is an automatic machine that is capable of movement in a given environment."

As opposed to fix based industrial robots, a mobile robot has its movement unlimited by its physical size due to its mobility. As a result, mobile robots can operate in a large workspace and explore unknown environments and therefore are able to perform tasks wherever needed. They have been used to perform a variety functions that are normally performed by humans or a machine in the form of a human, such as surveillance, exploration, patrol, homeland security, domestics helper (e.g. lawn mower), butler, care taker, and entertainer.

The recent decade has witnessed an explosion of research activities in mobile robotics. By and large, mobile robots can be categorized into three categories according to their operating environments: i) land (based) robots, ii) aquatic/underwater robots, and iii) aerial (air, flying) robots; each of them possessing sub-categories

Because of the need to operate in unknown and/or uncertain environments, mobile robots demand much higher level intelligence than traditional industrial robots. These requirements have been met by the phenomenal advancement in silicon technology and computing power. The rapid reduction in both size and cost of integrated chips has raised a huge interest for scientists to create intelligent systems.

This chapter provides an overview on mobile robotics. Instead of attempting to cover the wide area of this subject exhaustively, it highlights some key developments. Firstly, a functional model of generalized robots is presented, and serves as a common conduit which helps the analysis and comparison of robotics technology. It then paints a global perspective of mobile robotics market, followed by historical development highlighted by some key milestones. Subsequently, the chapter presents the state-of-the-art in mobile robotics research, and summarizes the works presented in this book. Some critical reflections and challenges ahead are highlighted as far as future development is concerned. Finally conclusions are drawn.

2. Functional model of generalized robots

A generalised robot or robotic system possesses a set of functional parts similar to human beings. As shown in Fig. 1, these functional parts include intellectual, statue, motivational, actuation, sensory, communications and energy.

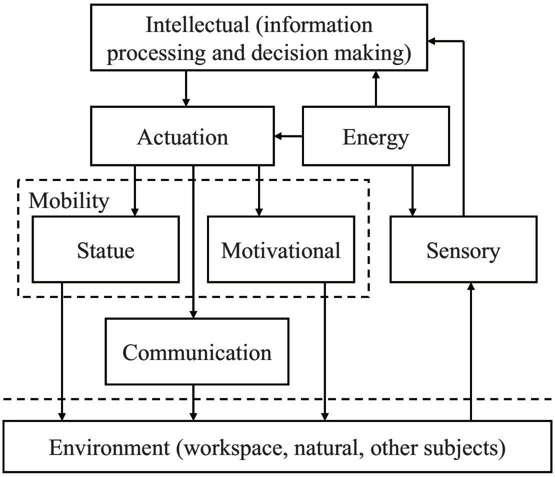

Intellectual (information processing and decision making)
Actuation
Energy
Mobility
Statue
Motivational
Sensory
Communication
Environment (workspace, natural, other subjects)

Fig. 1. Functional model of generalised robots

The functional model in Fig. 1 clearly illustrates the energy forward path and information feedback path. The intellectual part commands the actuation part, which in turn drives the mobility system - statue and motivational parts. Mobility is achieved through statue and motivational parts driven by actuators. Statue refers to body frame (skeleton) in case of a human being, and mechanical frame for a robot. In industrial robots, the mechanical frame is the mechanical links linked through joints. In mobile robots, the mechanical frame is the airframe for aerial robots, hull structure for underwater vehicle, and chassis for land vehicle, mechanical structure for humanoid robot.

Motivational parts are accessory mechanical parts to robots and limbs to human beings. They either enlarge or refine the robot mobility to execute specific tasks. Wheels, tracks, propellers increase the range of robot movement well beyond its statue. Grippers and end-effectors enhance the dexterity of the workspace and manipulation. Like a human, a robot accomplishes a delicate task like assembly, welding, and polishing by being equipped with motivational parts.

In accomplishing a defined mission, a robot, through its statue and motivational blocks, physically interacts with its operating environment. Robot operating environments can be classified into the following three categories:

Pre-defined and structured environment. The robot has all the knowledge about the environment and objects to deal with. It typically exists in factory automation

Semi structured environment. The robot has some pre-knowledge (e.g. GPS maps) about the environment, but it may change spatially and temporally. An example would be surveillance robot which tours its familiar territory, but the environment and objects inside it may change spatially and temporally.

Unstructured environment. The robot has no priori knowledge about it, with underwater being an example. The robot has to rely on its powerful sensory and navigation system to operate autonomously. A practical approach would be semi-autonomous system which accepts remote intervention from time to time.

In the energy forward path, a robot exerts energy onto the environment. In this process, the primary energy source (typically electrical) is converted to other types of energy such as kinetic, mechanical, wave, etc while a task is performed by the robot. In this process, the robot communicates with the environment through data, image, video or sound. It receives situational information through its Sensory functional block, which closes the control loop.

Although both robots and human beings possess the same set of functional blocks, hence have correspondence in each block. A parallel drawn between robots and human being in terms of realisation of these functions is shown in Table 1. It is apparent that for a robot to perform similar tasks like a human being, it indeed needs to possess the seven essential functional parts. Nevertheless, for simple applications in a more deterministic operating environment some of the functions such as sensory function are reduced to minimal.

Functional Blocks	Human Beings	Robots
Intellectual	Brain	Microprocessor (computer hardware and software)
Statue	Skeleton	Mechanical frame (airframe, chassis, hull).
Motivational	Limbs	Wheels, legs, tracks, propellers, grippers, etc.
Actuation	Muscles	Hydraulic, electric, pneumatic. piezoelectric, electrostatic actuators; artificial muscles.
Sensory (perception)	Eyes, ears, skin	Cameras, optic sensors, sonar, sound, infra-red light, magnetic fields, radiation, etc.
Communication	Speech, gesture	Data, image/video, sound
Energy	Food / energy storage	Power source / energy storage.

Table 1. Robots versus human beings: functional blocks

As a result, a mobile robot is a complex assembly of fundamental building blocks, each of which has to be carefully chosen based on specifications such as terrain, mission duration, goal and atmospheric conditions. The “optimal” design aims for the robot to accomplish a specific mission most cost effectively and efficiently. In contrast, a human being always uses the same set of “functional bocks” to accomplish any type of missions - marathon, sprinting, assembly, inspection and entertainment.

The mobility system accomplishes certain tasks by interacting with its operating environment, known or unknown. It is therefore important to take the various functions into consideration when design a mobile robot.

Mobility: Mobility in mobile robots requires mechanical frame and motivational parts. Another consideration is to give the mobile robot locomotion capabilities based on its deployment environment and missions. If the robot will only encounter smooth ground, wheels or tracks would be reasonable. Rougher terrain would require bigger wheels. In search and rescue mission in debris, leg locomotion is desired.

Sensors: A large collection of sensors are available to detect information about the environment. They can be used for monitoring purposes (chemical sensors, thermometers) or to help the robot to maintain its operations (accelerometers, GPS, etc…).

Actuators: They allow the robots to perform extra tasks besides mobility.

On-board Computation: Depending on its purpose a mobile robot will require more or less advanced on-board computation. Simple robots will just need steering computation capabilities whereas robots interacting with humans will require advanced electronics to successfully communicate.
Software: The software includes all the processes required by the robot to operate. They range from low level mobility processes up to behavioural processes.
Energy: Mobile robots usually operate in remote environments where energy is not readily available. Carrying the necessary energy to finish the assigned task is a necessary condition when creating a mobile robot.
Communication: In many applications, communication is essential to a mobile robot. It can be used to monitor the robot, to communicate with other robots or to communicate information.

3. Global perspective of mobile robotics market

They are several identified markets for mobile robots: service and military robots.

3.1 Service

Service robots are usually divided into 2 sub-categories: Professional and Domestic robots. The first one includes robots designed to serve either humans or equipment. As an example, medical robots can be used to assist for surgeries as well as help for the training of surgeons. Robotically-assisted surgery systems markets are supposed to have rapid growth. The market that was at $626.5 million in 2007 is anticipated to reach $1 billion in 2008 and is forecast to reach $14 billion by 2014. Mobile robots are also developed to work as tour guides (Toyota's "Robina" or Fujitsu's "Enon").

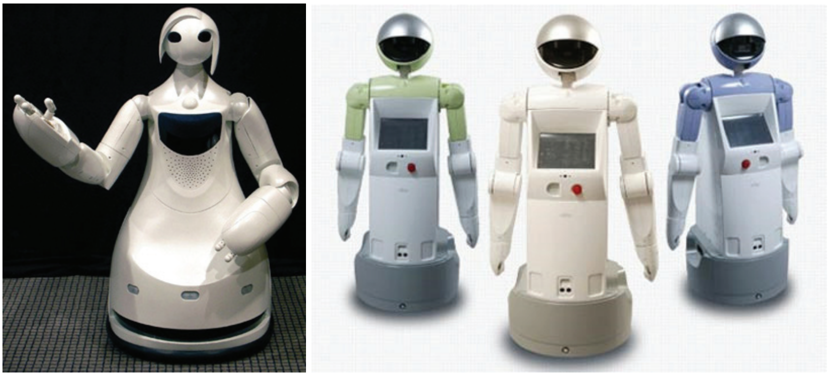


Fig. 2. Robina, Toyota's tour guide (left) and Fujitsu's Enon (right)

Service robots are also developing for equipment service such as pipe and window cleaning. They are designed to perform tasks such as inspection, maintenance and repairs.
The latter category of service robots is known as personal robots. It includes educational robots, home care, entertainment and home assistance. Educational robots are usually very versatile platforms that help students get a global experience with mobile robots. MobileRobots platforms represent a reliable base and powerful software that helps make students' robotics experiences a success. The market for educational robotic kits at $27.5 million in 2007 is forecasted to reach $1.69 billion by 2014. Home care robots introduced so

far are rather simple (vacuum robots such as Roomba® or lawn mowing robots such as the Robomawer®), but general purpose home-care robots are on the way. Consumer robot markets for house cleaning; lawn mowing, pool cleaning, and general home services reached $227 million in 2007 and are expected to attain $1.7 billion by 2014.

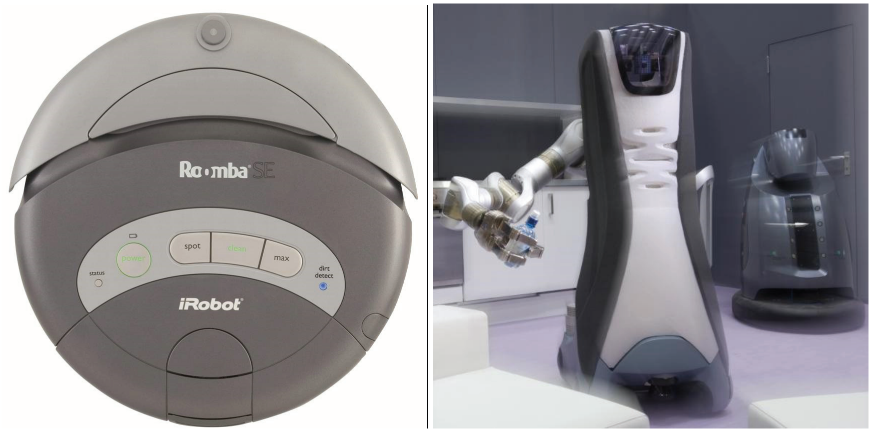


Fig. 3. The Roomba® robot and the latest Care-O-bot®

Entertainment robots such as robot toys represent a growing part of consumer electronics and toy robots are usually "Must-have" for many children during the Christmas season. A good example of the success of toy robots is Robosapiens from Wow Wee that sold over 4 million units.

3.2 Homeland Security and Military

Robots play a major part in the quest to automate military ground systems. They provide vital protection of soldiers and people in the field and will potentially reduce the number of fatalities in combat.

The use of remote-control robots in Iraq started as simple robots to investigate possible roadside bombs. Since then, a lot of robots have been developed to dispose of bombs in a combat or urban environment. Smaller and cheaper MARCBOTs and BomBots models are being engineered to provide even more help.

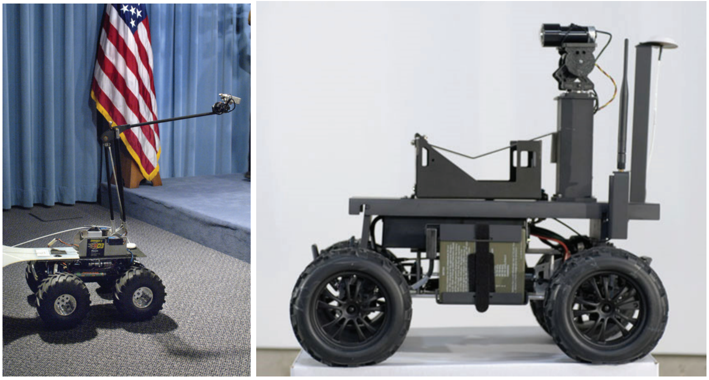


Fig. 4. Multi-Function Advanced Remote Control Robot (MARCBot - left) and BomBot (right)

The U.S. Department of Defense (DoD) released a report in 2007 that investigates at the future of military's unmanned systems for the next 25 years. This report puts into perspective the most urgent needs that are supported both technologically and operationally by various unmanned systems. These needs are listed below and should be considered by researchers as they will probably represent a large amount of funding from the DoD.

1. Reconnaissance and Surveillance.

Unmanned systems will play an important role in the field of reconnaissance, both electronic and visual. It has become essential for troops to be able to survey areas of interest while being under cover. Efforts should be made to increase the standardization and interoperability of unmanned systems as the number and diversity of users is going to increase significantly.

2. Target Identification and Designation.

Identification and localization of military targets are still to be improved as they are critical. A mistake usually results in the non-destruction of a target or worse, the death of innocent civilians. It is therefore important to reduce the latency and increase the precision of GPS guided weapons. The capability of unmanned systems to operate in high threat environments is not only safer for troops but potentially more effective than the use of current manned systems.

3. Counter-Mine Warfare.

The military authorities have pointed that sea mines have caused more damage to the fleet than all other weapons systems combined, and that Improvised Explosive Devices (IEDs) are the number one cause of casualties in Iraq. A lot of work has already been done to improve the military’s ability to find, tag, and destroy both land and sea mines, and unmanned systems naturally seems to be the solution to fulfil these missions.

4. Chemical, Biological, Radiological, Nuclear, Explosive (CBRNE) Reconnaissance.

Efforts should be made in developing unmanned systems able to identify and locate chemical and biologic agents and to survey the extent of affected areas.

The military robots markets were $145 million in 2007 and are anticipated to reach $6.9 billion by 2014.

4. Historical development of mobile robots

4.1 Evolution from fixed base robotics to mobile robotics

Interest in mobile robotics mainly comes from the need to explore areas that humans cannot explore. The reason may be that the environment is hazardous (radioactive, at deep sea level) or too far in distance and time (space exploration). Under such conditions, fixed-based robots are not sufficient.

4.2 Milestones of autonomous ground vehicle

Grey Walter's tortoises (1948-9, Bristol)

Grey Walter tortoise was an attempt to demonstrate that rich interconnection between a small amount of brain cells has the potential to create to complex behaviors. His first two robots, which were called "Machina Speculatrix", were developed between 1948 and 1949 and were named tortoises because of their shape and speed. These very early three-wheeled robots were indeed capable of photo taxis (they could find their way to a recharging station when they were low on power).

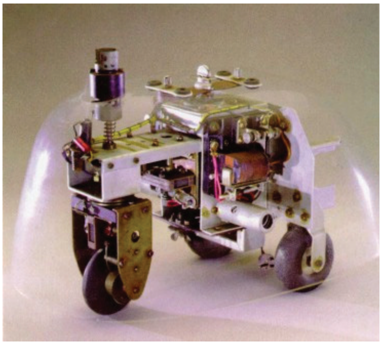


Fig. 5. Reconstitution of Grey Walter's tortoise

Shakey (1968, SRI)

Shakey was the world's first mobile robot capable of "reason" by taking actions based on its surroundings. Shakey was equipped with a TV camera, a triangulating range finder, and bump sensors, and was connected to DEC PDP-10 and PDP-15 computers via radio and video links. The robot was using programs for perception, world-modeling, and acting. Low-level action routines took care of simple moving, turning, and route planning. Intermediate level actions strung the low level ones together in ways that robustly accomplished more complex tasks. The highest level programs could make and execute plans to achieve goals given by a user. The plans could be stored for possible future use.

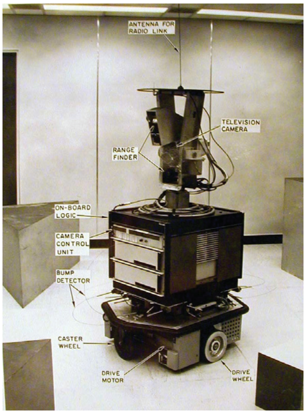


Fig. 6. Picture of Shakey

Stanford cart (1979)
In 1979, the Stanford Cart was able to cross a room full of chairs without human assistance. The cart was equipped with a TV camera mounted on a rail which took pictures from multiple angles and relayed them to a computer. The computer would then analyze the distance between the cart and the obstacles and steer accordingly.

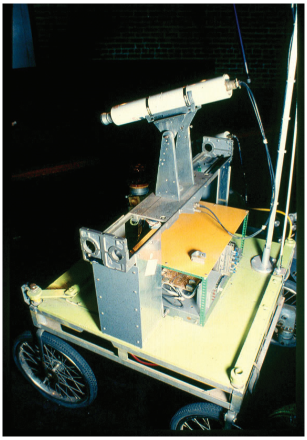


Fig. 7. Stanford Cart

Genghis (1988, MIT)
The Mobile Robots Group at MIT developed a six-legged walking robot named Genghis Khan, which was able to teach itself how to scramble over boards and other obstacles.

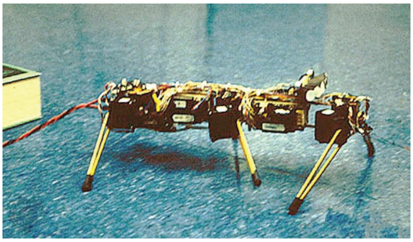


Fig. 8. MIT's Genghis Walking Robot

Khepera (1994, EPFL)
The Khepera is a small (5.5 cm) differential wheeled mobile robot that was developed at the LAMI laboratory of Prof. Jean-Daniel Nicoud at EPFL (Lausanne, Switzerland) in the mid '90s.

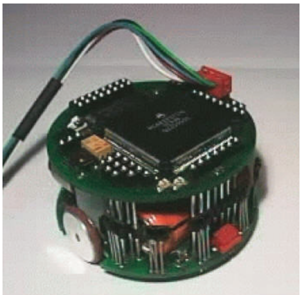


Fig. 9. The Khepera Robot

Stanley (2005, Stanford)
Stanley is an autonomous car, winner of the 2005 DARPA Grand Challenge (>200km in desert paths in ~7h). The vehicle incorporates measurements from GPS, a 6DOF inertial measurement unit, and wheel speed for pose estimation. The environment is perceived through four laser range finders, and a monocular vision system. Map and pose information are incorporated at 10 Hz, enabling Stanley to avoid collisions with obstacles in real-time.

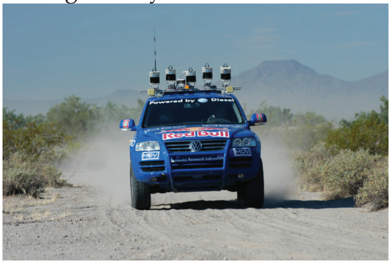


Fig. 10. Stanford Autonomous Car "Stanley"

4.3 Milestones of unmanned aerial vehicles

1916: aerial torpedo - Hewitt-Sperry Automatic Airplane
The Hewitt-Sperry Automatic Airplane project was started during World War I. Its purpose was to develop an aerial torpedo, carrying onboard components capable of sustained flight over a long period of time without the need of human manipulation. The "brain" of this

unmanned vehicle consisted of gyroscopes, mechanically coupled to the aircraft control surfaces so as to maintain its stability.

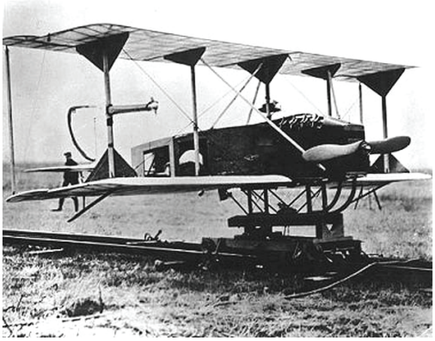


Fig. 11. The original Sperry Aerial Torpedo, 1918

1923: Pilotless Airplane
In 1923, the Army Air Service announced that a pilotless airplane, equipped with an automatic control device had been developed to a point where it has made successful flights of more than ninety miles. At the time, it was said to be more accurate and dependable than any human pilot.
1935: Drones and Radio Controlled Aerial Target
In 1935, a large number of RC targets were produced, the "DH.82B Queen Bee", derived from the de Havilland Tiger Moth biplane trainer. The prototype was first flown on January 5, 1935. The name of "Queen Bee" is said to have led to the use of the term "drone" for remote-controlled aircraft. These were used as targets for military anti-aircraft gunnery practice.

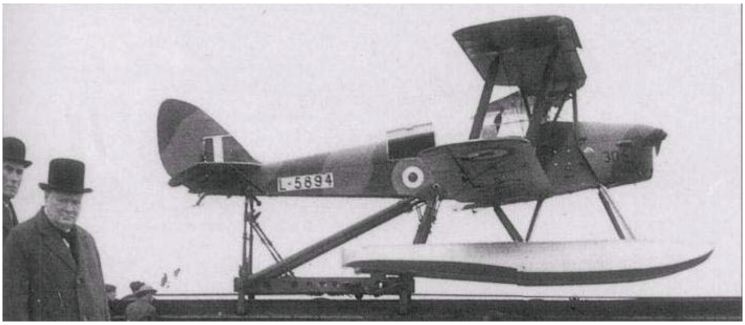


Fig. 12. The De Havilland DH.82B Queen Bee

1956 : Surveillance Drones
Drones used as target were successful and this led to it being used for other purposes. The Ryan Firebee was a good platform for test experiments, and tests to use it for reconnaissance missions were successful. A series of reconnaissance drones derived from the Firebee, known generally as "Lightning Bugs", were used by the US to spy on Vietnam, China, and North Korea in the 1960s and early 1970s.

Fig. 13. Firebee Lightning Bugs

1964: Unmanned Combat Air Vehicle

In 1964, "Project CeeBee" was created to experiment using a Firebee, fitted with underwing pylons, to carry two 115 kilogram (250 pound) bombs. The Firebee was launched from a ground station, using a booster for initial propulsion. However, the CeeBee experiments were not successful enough; the cause being that shooting at a target proved much more complex than flying over an area and taking pictures. The very first launch of a missile, a Maverick, from a UAV occurred on 14 December 1971, thanks to more advanced guided munitions.

1992: Miniature UAVs (MAVs)

During the early 1990s, the idea that small UAVs could be of interest was introduced. DARPA started a 35 million dollar project to develop and test a UAV. The specifications were that the MAV needed to carry a night / day imaging device, and operate for at least 2 hours. This study came to an end in 2001, and DARPA then investigated commercial vendors capable of producing MAVs to the initial design specification.

The majority of modern MAVs are designed to be smaller than conventional UAVs, but not as small as originally envisioned. Most UAVs used by the military and government organizations are hand launched units, able to be transported, deployed and operated by a single user. Smaller MAVs are still under investigation, but currently the majority of used MAVs are on the larger side.

5. The state of the art in mobile robots

5.1 Design methodology

The development of any type of mobile robot is challenging and takes a lot of time and considerations. It would be possible to improve the productivity of the development work by optimizing the design methods and tools.

The design process of a mobile robot can be divided into three logical parts based on its architecture: software, hardware and mechanical.

The software is usually divided hierarchically into two parts:

High level software that allows the mobile robot to function autonomously and fulfil its designated mission.

Low level software includes the basic motor functions such as the steering algorithm as well as reflex functions like communication routines or collision avoidance programs.

To create the software architecture, the most common approach is to replace the sequential programs into a collection of simple, distributed, and decentralized processes with capability for each to access information from sensors as well as to send commands to the actuators of the robot. This approach was first introduced under the name "subsumption architecture" and was proposed by Rodney Brooks at MIT. Local behavioural routines run constantly in parallel using only local available information. The emergent overall behaviour turns out to be flexible and robust. All these emergent approaches are based on biological systems. They try to mimic mechanisms from biological systems, especially adaptive behaviours. Adaptation, combined with a decentralized bottom-up approach, is often seen as a solution to the problem of generating and maintaining stable behaviours in partially unknown and dynamic environments.

The hardware part is twofold. Electronics parts, such as digital, analog and power electronics components, are used to convert software requests into actuator control signals and to scale and digitize sensor signals. Actuators provides mobility to the robot by transforming its input signal, likely electrical, into motion. The types of actuators used for mobility have to careful chosen depending on the environment where the mobile robot will ambulate (air, land, sea). Sensors are used to measure a physical quantity from the environment and convert it into a signal used either for locomotion or monitoring a variable of interest.

The mechanical part can be divided in two. Mechanisms can be used to transform the movement of the actuators; for example change the rotational movement of a motor into a translational one. The body is designed to protect the main parts of the robot from the environment; it gives the robot its integrity.

5.2 Perception and situational awareness

The terms such as perception and awareness have been replaced by the more general term "cognition". In robotics, cognition includes the creation of high level information from the combination of low level pieces of information and memory. Such high level information can be a "mental" map of the surroundings as well as a prediction of the future evolution of that environment. Cognition also requires different types of behaviour such as planning, sensing, recognition, learning, coordination and other "human-like" tasks.

There exist several possible states of cognition. To illustrate these different states, we introduce a cognition model found in (Patnaik, 2007). This model includes seven different mental states (sensing & acquisition, reasoning, attention, recognition, learning, planning, action & coordination) as well as two separate types of memory (short-term and long-term). The cognition is this model is realized by three cycles. These three cycles are the "acquisition cycle", the "perception cycle" and the "learning and coordination cycle".

The purpose of the acquisition cycle is to memorize the information from the sensors into a short-term memory. The data are then compared to what is known (i.e. stored in the long term memory) for validation. The perception cycle uses information stored in the long term memory to interpret data gathered during the acquisition cycle. Finally, the learning and coordination cycle plans the future actions of the robot based on stored information of the surroundings.

In robotics, cognitive behaviours are achieved with the help of computing techniques. The most common pattern is that sensor readings are used by a micro-controller to steer the robots actuators so as to achieve a specific task.

Two of the most researched areas in perception are vision and recognition. To operate a mobile robot in an unknown environment, it is necessary to be able to analyze that environment to maintain the safety and integrity of the robot.

Whereas vision mostly depends on hardware, recognition is mostly based on computational techniques. Here is a list of recognition techniques:

Template matching: this technique is based on image processing where the image is scrutinized to find a match of a template image.

Feature-based model: instead of looking for templates, the image is analyzed to find some features or patterns. Such features are usually geometric shapes or specific colours.

5.3 Control of mobile robots

5.3.1 Formation Control

A large number of different strategies for formation control of a group of mobile robots can be found in the literature. Several frameworks stand out by the number of strategies that have been developed including the leader- follower schemes, behaviour-based methods, and virtual structure techniques. Between these three techniques, the leader- follower approach is the most acknowledged: one or more mobile robots are designated as leaders and are in charge of leading the formation, the other robots have no information about their headings and simply follow the leader(s): they are called followers. The followers usually try to maintain a set distance between them and the formation leader.

In the behaviour-based methods, the robots maintain formation thanks to two complementary processes: the first one , called detect-formation-position, computes the robot's ideal location within the formation based on sensor readings; the second one, called maintain-formation, generates the control commands to steer the robot toward the position determined by the first process. The virtual structure method uses the idea that points in space maintaining a fixed geometric relationship can be observed as behaving in the same way as points on a rigid body moving through space. When robots behave in this way, they are moving inside a virtual structure.

5.3.2 Control of non-holonomic mobile ground robots

Nonholonomic robots dynamics are characterized by equations involving the time derivatives of the system's variables and constrains. These dynamic equations are non integrable. Nonholonomy is usually encountered when the system has less control inputs than controlled variables. As an example consider a wheeled mobile robot that has two controls (linear and angular velocities) while the domain it evolves in is 3-dimensional. Therefore, every feasible control signal does not necessarily correspond to a feasible path for the system. This is the reason why the geometry-based techniques developed in motion planning for holonomic systems cannot be directly applied to nonholonomic ones.

The purpose of motion planning is to obtain open-loop controls to steer a nonholonomic mobile robot from an initial state to a desired final state along a feasible trajectory that is possibly optimal. The principal motion planning techniques can be classified as differential geometric, geometric phase, and optimal control. Differential geometric techniques are based on an extended set of equations for the system in conjunction with the original one. A first control sequence is computed from the extended system and is used to create the final control input used on the real system. Geometric phase techniques use the concept of holonomy and path integral along an m-polygon to transform the differential constraints of the system to algebraic geometric phase equations. The optimal control path planning problem is usually formulated as a two point boundary value problem with various problem constraints. The optimization criterion being minimized is a control performance index which can include energy saving, formation, collision avoidance or bang-bang conditions. This technique can easily be extended to robots moving in a three dimensional environment.

Obstacle avoidance can also be considered and increases the motion planning's difficulty. The methodology then needs to take into account both the constraints due to the obstacles and the nonholonomic constrains. It appears necessary to combine geometric techniques addressing the obstacle avoidance together with control theory techniques addressing the special structure of the nonholonomic motions.

Readers can refer to (Li and Canny, 1993) for a more complete review of motion planning techniques for nonholonomic systems.

5.3.3 Control of unmanned air vehicles

Unmanned air vehicles are usually controlled by an autopilot, i.e. a system allowing the UAV to fly autonomously. Nowadays, autopilot systems are automatically implemented in modern aircrafts. The purpose of the UAV autopilot system is to steer the UAV so as to follow either a predefined path or fly between waypoints. Advanced UAV autopilot systems are able to fly the UAV during all flight phases such as take-off, ascent, descent, trajectory following, and landing.

To our best knowledge, all commercially available autopilots are simple PID controllers. It is the best solution for most users as they are simple and require little knowledge to tune on a small UAV. However, PID controllers are also well known in control system theory to lack in optimality and robustness. Researchers have developed several advanced control techniques that can be used in autopilot systems for improved flight performances. We can cite fuzzy logic, neural networks, LQG and H∞ as successful examples of such techniques. Fuzzy logic and neural network techniques are not model-based (knowledge-based for fuzzy logic and data-driven for neural networks) and don't require advanced control knowledge from the user. They represent a good alternative to PID controllers as they usually perform better for multivariable flight control. However, in terms of guaranteed stability and performance, optimal and robust control techniques such as LQG and H∞ sounds more suitable. A combination of Linear Quadratic Gaussian controller and Kalman filter can be used to achieve better altitude control performance. H∞ loop shaping techniques can also be used on small fixed wing UAVs for improvements in noisy or even payload changing circumstances.

5.4 Biomimetic robots

Mobile robotics researchers have been inspired by the trans-disciplinary development in bionics, also known as biomimetics, or biomimicry. Bionics applies biological methods and systems found in nature to the study and design of engineering systems and modern technology. The exceptional natural abilities in many animals and insects have drawn much attention from biorobotists. A common approach is to build animal-like features into robots, and such robots are called biomimentic robots or simply biorobots.

It is debateable whether biologically inspired robotics should be simply emulation of some general feature like legs or wings of an animal, or a more considered approach in which specific structural or functional elements of particular animals is emulated in hardware or software (Delcomyn, 2007). It is difficult to draw a line between the two, although the latter may rely on biological aspect more.

The intensive research effort in searching for hardware and software solutions to emulate specific features of a real animal would expose their efficiency and deficiency, and improve our understanding of those animal features and engineering capabilities and limitations. Whether a design solution comes from engineering or a biological perspective, it is generally agreed that certain degree of fusion and integration between engineering and biology takes place. Despite minute differences in interpretation and emphasis, bionics, biorobotics, biomimetics, or biologically inspired robotics is emerging as a discipline in its own right. It has witnessed an explosion of research interests and efforts in the past few decades worldwide. Researchers working in this field rightfully claim their own identity – biorobotists, or bionicists. It is fitting to recognise that engineers apply biological principles to construct robots, and biorobotics in turn can advance biologists' knowledge and understanding of those same biological principles

Rapidly growing interests in biorobotics were confirmed by the statistics shown in (Delcomyn, 2007). There are more than 1.5 million hits one obtained by conducting a Google search on the phrase "walking robot". In terms of research literature included in the ISI Web of Knowledge database, the number of papers on mobile robotic machines with biological inspiration or variants as a key phrase has increased from an average of 9.2 papers per year between 2000 and 2004, to 16 in 2005 (an increase of over 70%), and 30 in 2006 (another increment of more than 85%). Though not large, this is nevertheless a field that is attracting much attention.

Biorobotics research has covered many types of animals to be emulated - fish and eel underwater; dog, cockroach, gecko on land; and black flies, wasps, bumblebees and other flying insects in air. These robots are built to swim, walk, climb a wall or a cable, or fly. Wall climbing robots have been considered to replace human beings to perform dangerous operations on vertical surfaces like cleaning high-rise buildings, inspecting bridges and structures, or carrying out welding on a tank. The locomotion of a wall climbing robot has become a key research, which is achieved through some kind of attachment mechanism.

Generally speaking, three main types of attachment mechanisms are used: suction, magnetic and dry adhesion mechanisms. The suction method creates vacuum inside cups through vacuum a pump, the cups are pressed against the wall or ceiling so that adhesion force is generated between the cups and the surface. This effect is dependent on a smooth impermeable surface to create enough force to hold the robot.

A wall-climbing robot with a single suction cup has been studied in (Zhao et al., 2004). It consists of three parts: a vacuum pump, a sealing mechanism with an air spring and regulating springs, and a driving mechanism. Two application examples were considered: i) ultrasonic inspection of cylindrical stainless steel nuclear storage tanks, and ii) cleaning high-rise buildings.

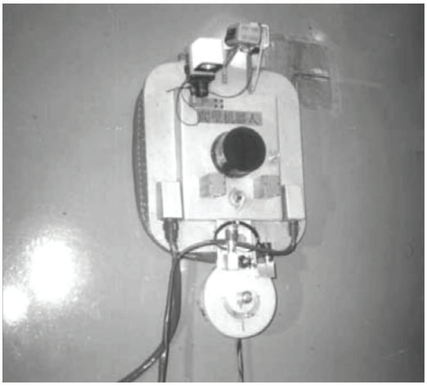
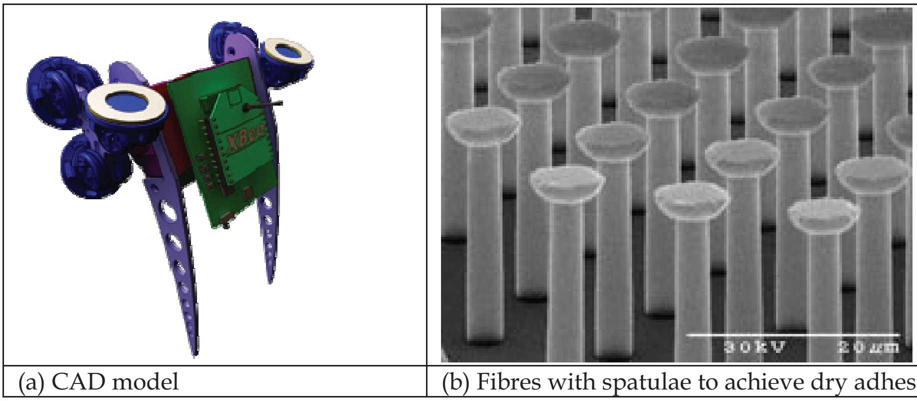


Fig. 14. A wall-climbing robot with a single suction cup

Magnetic adhesion has been implemented in wall climbing robots for specific applications such as nuclear facilities or oil and gas tanks inspection (Shen, 2005). In specific cases where the surface allows, magnetic attachment can be highly desirable for its inherent reliability.

Recently, researchers have developed and applied synthetic fibrillar adhesives to emulate bio-inspired dry adhesion found in Gecko's foot. An example is Waalbot using synthetic dry adhesives developed by Carnegie Mellon University, shown in Fig. 15. Fibres with spatulae were attached to the feet of the robot, and dry adhesion is achieved between the robot feet and the surface.

Also based on the dry adhesion principles is a bioinspired robot "Stickybot" (Kim et al., 2008). It is claimed that the robot climbs smooth vertical surfaces such as glass (shown in Fig. 16), plastic, and ceramic tile at 4 cm/s. The undersides of Stickybot's toes are covered with arrays of small, angled polymer stalks. In emulating the directional adhesive structures used by geckos, they readily adhere when pulled tangentially from the tips of the toes toward the ankles; when pulled in the opposite direction, they release.



(a) CAD model

(b) Fibres with spatulae to achieve dry adhesion

Fig. 15. Tri-leg Waalbot (http://nanolab.me.cmu.edu/projects/geckohair/)

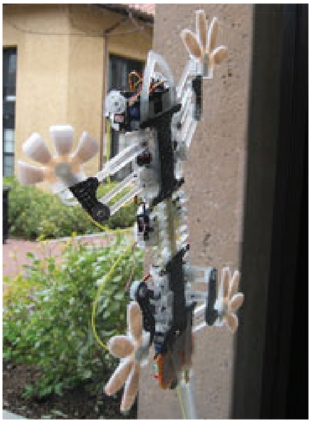


Fig. 16. Stickbot http://bdml.stanford.edu/twiki/bin/view/Main/StickyBot

Existing wall climbing robots are often limited to selected surfaces. Magnetic adhesion only works on ferromagnetic metals. Suction pads may encounter problems on the surface with high permeability. A crack in a wall would cause unreliable functioning of the attachment mechanisms, and cause the robot to fall off the wall. Dry adhesion methods are very sensitive to contaminants on wall surface.

For this reasons, a wall climbing robot independent of wall materials and surface conditions is desirable. The University of Canterbury has develop a novel wall climbing robot which offer reliable adhesion, manoeuvrability, high payload/weight ratio, and adaptability on a variety of wall materials and surface conditions (Wagner et al, 2008). Their approach is based on the Bernoulli Effect which has been applied in lifting device. It is believe that for the first time the Bernoulli pads have been successfully developed as a reliable attachment for wall climbing robots, as shown in Fig. 17.

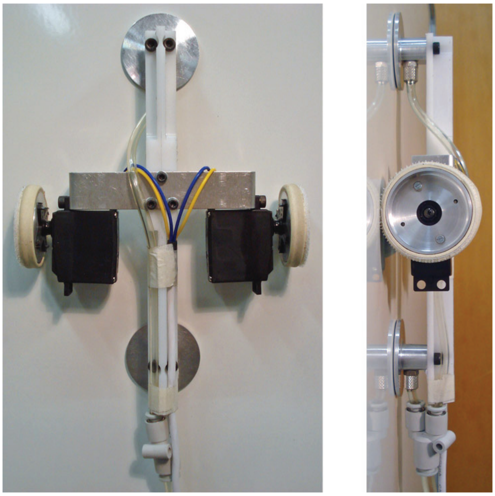


Fig. 17. An innovative wall climbing robot based on Bernoulli Effect.

As a standard Bernoulli device only offers a small attraction force, special attachment mechanisms have to be designed to enhance effectiveness of mechanical force generation. The mechanisms are designed to create the force without any contact to the surface. They literally float on an air cushion close to the wall. The contact between the robot and the wall lies in wheels with tires made of a high friction material which avoids sliding. The non-contact mechanisms provide a continuous and relatively constant suction force as the robot manoeuvres. The locomotion through the motorised wheels ensures smooth motion of the robot, which is paramount for continuous 3D curvature surface operation.

The advantage of the novel approach is that the adhesion force is largely independent of the type of materials and surface conditions. High attraction forces can be achieved on a broad range of surface materials with varying roughness. The experimental results show that the robot weighing 234 grams can carry an additional weight of 12 N, with the force/weight ratio being as high as 5. The device accommodates wall permeability to air to a certain degree, which means that gaps and cracks, which would pose a hazard to conventional suction methods, can be tolerated by the novel device. Furthermore, the robot is easy to setup using a standard pressure supply readily available industry wide.

5.5 State-of-art reported in this book

This book reports current states of some challenging research projects in mobile robotics ranging from land, humanoid, underwater, aerial robots, to rehabilitation. The book also covers some generic technological issues such as optimal sensor-motion scheduling, mobile data collector, augmented virtual presence, and indoor localization techniques. Some of the research works are directly related to demanding task and collaborative missions.

Chapter 2 introduces a field robot using the rotated-claw wheel that has strong capacity of climbing obstacles. The experimental results demonstrate that Rabbit can move in different terrain smoothly and climb over step of 8.1cm and slop of 40°. The Rabbit can adopt different moving modes on different terrains. Because the rotated-claw wheel overcomes the disadvantages of conventional mobile robot wheels, it provides a better solution for field and planetary robots.

Chapter 3 presents a mobile wheeled robot with step climbing capabilities using parallel individual axels. Each axel offset a given radius from the wheel set axis of rotation. In this way, the wheels could revolve and also be powered from a source of angular speed and torque. The wheel sets could also revolve in any direction independent of the rotation of the wheels. This design seemed to satisfy the primary requirements for the robot for both rough terrain and stair climbing.

Chapter 4 reviews some of the main efforts made over the past 20 years in the field of cable-climbing mechanism design to provide a basis for future developments in this field. History of the research in this field shows that due to the huge benefit of early detection of likely damage to the line, even the cable-climbing robots capable of only climbing on part of the line between two obstacles are in use, and further researches in this field will definitely benefit the power companies to efficiently manage their assets. In addition, based on the reviewed works, a flying-climbing platform which is a commercially available UAV modified with a cable-climbing mechanism would enormously benefit the line inspection quality and the design universality.

Chapter 5 proposes a multi-sensing fusion system to mimic the powerful sensing and navigation abilities of a cockroach. It consists of binocular vision system based on infrared imaging, and tactile sensors using fibre optic sensors and position sensitive detectors. The paper further proposes a distributed multi-CAN bus-mastering system based on FPGA (Field Programmable Gate Array) and ARM (Advanced RISC Machine) microprocessor. The system architecture provides stage treatment for control information and real-time servo control. The control system consists of there core modules: (1) node part of CAN bus servo drive; (2) distributed multi-CAN bus-mastering system composed by FPGA; (3) software system based on ARM and RTAI.

Chapter 6 highlights some characteristics observed from human abilities in performing both knowledge-centric activities and skill-centric activities. Then, the observations related to a human being's body, brain and mind guide the design of a humanoid robot's body, brain and mind. After the discussions of some important considerations of design, the results obtained during the process of designing the LOCH humanoid robot are shown. It is hoped that these results will be inspiring to others.

Chapter 7 reports an AUV prototype that had been developed recently at the University of Canterbury. The AUV was specially designed and prototyped for shallow water tasks, such as inspecting and cleaning sea chests of ships. It features low cost and wide potential use for normal shallow water tasks with a working depth up to 20 m, and a forward/backward speed up to 1.4 m/s. Each part of the AUV is deliberately chosen based on a comparison of readily available low cost options when possible. The prototype has a complete set of components including vehicle hull, propulsion, depth control, sensors and electronics, batteries, and communications. The total cost for a one-off prototype is less than US $10,000. With these elements, a full range of horizontal, vertical and rotational control of the AUV is possible including computer vision sensing.

Chapter 8 establishes an approach to solve the full 3D SLAM problem, applied to an underwater environment. First, a general approach to the 3D SLAM problem was presented, which included the models in 3D case, data association and estimation algorithm. For an underwater mobile robot, a new measurement system was designed for large area's globally-consistent SLAM: buoys for long-range estimation, and camera for short-range estimation and map building. Globally-consistent results could be obtained by a complementary sensor fusion mechanism.

Chapter 9 addresses flight dynamics modelling and method of model validation using on-board instrumentation system. It was found that the aerodynamics coefficients determined by software packages do not accurately represent the actual values. The experimental drag coefficients are higher than those predicted by the software model and this has a large affect on the accuracy of the flight dynamic model. The validation process involves in-flight measure of all parameters as well as wind speed detected by in-house build air-speed sensor. The sensor hardware allowed the collection of flight data which was used to assess the accuracy of the flight dynamics model. The presented validation process and hardware makes a step towards completing an accurate flight simulation system for auto-pilot development and preliminary design of UAVs.

Chapter 10 describes a numerical procedure for optimal sensor-motion scheduling of diffusion systems for parameter estimation. The state of the art problem formulation was presented so as to understand the contribution of the work. The problem was formulated as an optimization problem using the concept of the Fisher information matrix. The work further introduces the optimal actuation framework for parameter identification in distributed parameter systems. The problem was reformulated into an optimal control one.

It solved parameter identification problem in an interlaced manner successfully, and successfully obtained the optimal solutions of all the introduced methods for illustrative examples. It is believed that this work has for the first time laid the rigorous foundation for real-time estimation for a class of cyber-physical systems (CPS).

Chapter 11 presents some heuristics for constructing the mobile collector collection route. The algorithm's performances are shown and their impact on the data collection operation is presented. There are many directions in which this work may be pursued further. Statistical measures are required to measure the buffer filling rate and thus the sensor can send its collection request before its buffer is full, which gives an extra advantage for the mobile collector. Applying multiple mobile collectors can enhance the performance. Control schemes for coordinating multiple collectors need to be designed efficiently to maximize the performance.

Chapter 12 discusses the development of the AR-HRC system from concept and background through the design of the necessary set of interfaces required to enhance human-robot interaction. It has shown that the AR-HRC system does enable natural and effective communication to take place. The use of AR affords the integration of a multi-modal interface combining speech and gesture interaction, as well as providing the means for enhanced situational awareness. The AR-HRC system gives the user the feeling of working in a collaborative human-robot team rather than the feeling of the robot being a tool, as a typical teleoperation interface provides. Therefore, the development of the AR-HRC system brings closer the day when humans and robots can truly interact in a collaborative manner.

Chapter 13 details a set of classifications of indoor localization techniques. The classifications presented in this chapter provide a compact form of overview on WSN-based indoor localizations. The chapter further introduces server-based and range-based localization systems that can be used for the indoor service robot. Specifically, it presents UWB, Wi-Fi, ZigBee, and CSS-based localization systems. Since the methods introduced in this chapter are RSSI-based method, the system is very simple and the implementation cost is much cheaper than TOA and TDOA-based methods, such as Ubisense systems and CSS systems.

Chapter 14 proposes a wearable soft parallel robot for ankle joint rehabilitation after carefully studying the complexities of human ankle joint and its motions. The proposed device is an improvement over existing robots in terms of simplicity, rigidity and payload performance. The proposed device is very light in weight (total weight is less than 2 Kg excluding the weight of support mechanism) and is inexpensive. The kinematic and workspace study is carried out and the performance indices to evaluate the robot design are discussed in detail. It attempts to use an algorithm that maximizes a fitness function using weighted formula approach and at the same time obtain Pareto optimal solutions.

6. Challenges ahead

Despite rapid development of robotics technologies in the past decades, there still exist many technical issues and challenges ahead in realising the full potential of mobile robots. These challenges include standardization, software, hardware and control. In face of ever increasing aging population and human augmented functions, service robots will have significant impact on the society as well as individuals.

6.1 Standards and architecture

In the last century, the manufacturing industry has benefited enormously from the rapid advancement and maturity of computer numerical controlled (CNC) machines. Mechanical parts are automatically produced from a computer model. CNC machines have become a common tool widely accepted by the manufacturers, large or small. The same story cannot be said of robots. Even for a simple task, different robots will have different ways of programming and execution. The lack of standards in robot programming has become a serious limiting factor in promoting robots in the industry. There is a need of a combined effort from the industry, research community and professional bodies to standardise robot programming language. This will be a significant step forward using robots as a common versatile tool that can be easily deployed, mastered and re-programmed.

6.2 Intuitive learning and control

As researchers aspire to create more mobile robots for health care, domestic work, or automating tasks that too dangerous for human beings, the intuitiveness of robots in almost non-existent at present. The industry still feels much more comfortable about hiring a new worker who understands instructions, does a job effectively, and can be easily retrained than employing a mobile robot who is not humane in terms of learning. A human operator learns how to correctly carry out a job through observation and iterative learning by practicing. These processes are simple and intuitive to a human being, but it is still impractical for a mobile robot, indeed any types of robots.

An illustrative example would be polishing of 3D high-pressure turbine (HTP) vanes (Chen 2000a). The manual operation is depicted in Fig. 18. The procedure of the operation is as follows.

Manipulate the part correctly in relation to the tool head, with two-arm coordination.

Exert correct force (up to 15 kg) and compliance between the part and the tool through wrists, and control the force interaction based on process knowledge.

Adapt to part-to-part variations and observe the amount of material removed through visual observation and force feedback.

Check the final dimension with gages.

Repeating step 1 to 5 until the final dimension is achieved. It takes about 10 minutes to finish one piece.

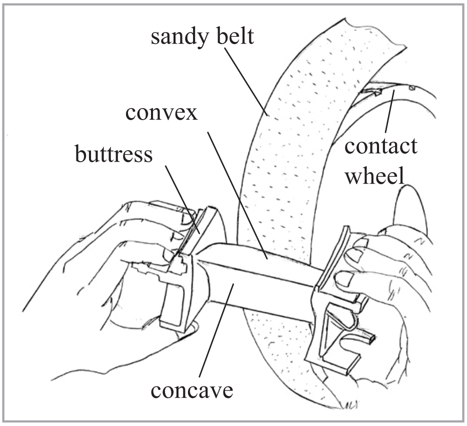


Fig. 18. Manual polishing of aero engine turbine vanes

A robotic system was developed to automate the operation depicted in Fig. 19. The system (Chen, 2000b) has a self-compliant mechanism that grips the part as an operator does with two hands.

In its appearance, the robot does look like an operator in manipulating the part to achieve desired contact states between the workpiece and the polishing tool, remove the right amount of materials through force feedback control. It shortens the cycle time from 10 minutes for manual polishing to an average of 5.75 minutes, resulting in an improvement of 42.5% (Chen, 200b). Such an improvement mainly comes from two advantages that the robot has over a human operator. Firstly the robot can exert a large force constantly while the operator is unable to exert a large force for a long period. Secondly the robot is more deterministic in planning the polishing paths after obtaining the part measurement (which is part of 5.73 minutes cycle time), hence removes the iterations of inspect-then-polish in manual operation and optimises and reduces the number of polishing passes.

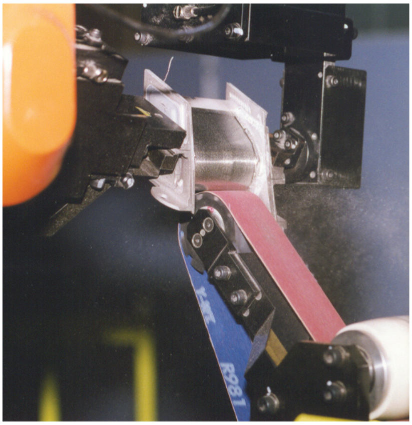


Fig. 19. Robot grips the part with a compliant robot end-effector grips part.

However this system's force regulation during the polishing is still based on simple robot positioning control. To increase or decrease the contact force which is detected by a spring mechanism mounted inside the polishing tool, the robot moves the part in or out in reference to the polishing tool. For a robot to be more adaptable to contact tasks, and intrinsically safe, combined force and visual servoing is highly desired as if the operator exerts a muscular force based on his/her tactile and force sensing intuitively.

As mobile robots make inroad into service and healthcare sectors where physical interactions between robots and human beings take place, controlling robot movement, particularly the movement of motivational parts (legs, feet, arms, fingers), intuitively based force servoing is paramount. From engineering point of view, such intuitive control coupled with flexible and compliant manipulator will enable a robot to execute a contact task, e.g. help a patient lay down or use toilet, more efficiently and safely.

Another important aspect of intuitiveness lies in the way to learn skills and re-apply the skills. Certainly, with the skills learnt, the operator can easily handle other types of parts, and the skills are reusable. If a robot can emulate human intuitiveness, it should be able to take simple instructions, observe the manual operation, practice under supervision, and eventually master the skills to polish a 3D surface. Like a human operator, the robot can take the skills learnt and be readily transferrable to another product line. There is still a long way for robots to gain such intuitiveness.

6.3 Software designs

Software designs rely on a software platform to achieve desired cognition and intelligence.

Practical Robot Software Platforms. Various robot software platforms are already available (e.g. Evolution Robotics). These systems can provide a cost-effective way of producing and operating home-security robots, and will continue to increase in functionalities.

Robot Cognition and Artificial Intelligence. Advances robots with cognitive abilities, artificial intelligence and associated technologies are vital for the development of intelligent, autonomous robots for domestic applications.

In the future, mobile robots will require increased flexibility and robustness to the uncertainties of the environment. Their predicted increased presence in daily life means that they will have more tasks to perform and that these tasks will be diverse. An envisioned approach to fulfil these requirements is to engineer the robots so that some of the processes, inherent to the multiple functions to be performed, can be adapted based on contextual knowledge. In other words, information from the robot's surroundings gathered by multiple sensors could be used to help the robot to achieve its tasks and even determine future tasks. We can imagine a household robot deciding to clean the room when it "feels" that the room is dirty.

Robotics is not the only field of research where contextual knowledge plays an important role. In the literature, five specific tasks stand out as important for future research: Behaviour, Navigation, Localization and Mapping, Perception.

Decision making based on contextual knowledge can easily be foreseen as useful in robotic scenarios, the common scenario being to adapt the robot's behaviour to different situations which the robot may encounter in operation. This is usually dealt with via plan selection, hierarchical approaches to planning and meta-rules.

In the context of motion planning, the goal would be to find general solutions that can easily be adapted in case of a change of context. Typically, however, to obtain effective implementations, specific algorithms and optimal solutions for specific cases are required. The use of contextual knowledge can provide the necessary information to modify a general technique so as to solve the problem at hand. In many cases, the navigation process is included into a more complex process (for example exploration of the environment) where the robot needs to find a target to reach and meet that target. It can easily be envisioned that contextual knowledge can help set priorities when the robot has different missions to fulfil.

Contextual knowledge can also be useful for a robot to map its environment in an abstract manner. Introducing language-based information (for example objects names, colors or shapes) in addition to precise information about the environment can help the decision making process as well as provide improved human-robot interactions. Contextual knowledge may be used for selecting routines. The use of contextual knowledge can be enlarged, for example, to decide when the robot can halt the mapping process and switch to another function.

The use of contextual knowledge has a long tradition in Vision, both from a cognitive perspective, and from an engineering perspective. Indeed, also robot perception can benefit significantly from contextual knowledge. Moreover, it is through the sensing capabilities of the robot that environmental knowledge can be acquired. In robot perception, normally, iterative knowledge processing occurs: a top-down analysis, in which the contribution given by the environmental and mission related knowledge helps the perception of features and objects in the scene; a bottom-up analysis, in which scene understanding increases the environmental knowledge

6.4 Hardware technologies

Affordable robots will continue to be built using fairly conventional hardware—off-the-shelf electronic components, batteries, motors, sensors, and actuators. Materials and designs for statue and motivational parts of the robots have not changed fundamentally, which has been a significant limiting factor in advancing robotics technology and robot performances. Table 2 compares the lifting capacity and lifting-to-weight ratio. For a typical articulated industrial robot weighing 359 Kg, the lifting-to-weight ratio is about 0.03. The well-known Honda humanoid robot lifts 1 Kg with two hands while a person of a similar body mass can lift 20 Kg. Weighting lifting athletes can have lifting-to-weight capacity as high as 2.4. In this regards, the robot construction is very inefficient compared to human build. Novel materials and actuators are a key to building lighter robots for higher handling capacity.

	Self weight (Kg)	Lifting capacity (Kg)	Lifting-to-weight ratio
ABB IRB 2000	350	10	~0.03
Honda Asimo	52	1 (for two hands)	~0.02
2008 Olympic Women 53 Kg Weightlifting Gold	53	126 (clean & jerk)	~2.4
A person having similar weight to Asimo	52	~ 20	~0.4

Table 2. Robot versus human: lifting capacity

Table 3 compares the motion range of robot wrist and human wrist. Generally speaking, robots have better positioning repeatability and consistency than human beings. But human beings accomplish precision tasks through powerful perception and intelligence.

	Human	Robot
Positioning repeatability	~ mm	~ μm
Consistency	Fair	good
Wrist movement (roll, pitch, yaw)	±90°, ± 40°, ± 20°	±200°, ±120°, ±200° (ABB IRB 2000)
Finger flexibility	Dexterous (14 dof)	Primitive
Mobility	Excellent	Poor
Dynamics & compliance	Excellent	Poor
Process knowledge	Excellent	Primitive
Observatory control	Excellent	Primitive

Table 3. Robot versus human: manoeuvrability and Intelligence

Robots generally have larger range of wrist movement. As illustrated in Table 3, the industrial robot ABB IRB200 has ±200°, ±120°, and ±200° movement for roll, pitch and yaw respectively, as opposed to ±90°, ± 40°, and ± 20° for human wrists. However the robot's superiority does not translate to better dexterity. In fact a human operator is much more dexterous in manipulating objects as in the case of polishing 3D surface. This arises from: 1) natural coordination of two hands, and more importantly our fingers, a total of 14 degrees of freedom, are separately actuated. In the case of Asimo, five fingers are driven by the same motor, which can only achieve limited handling dexterity and coordination.

As robots will interact more and more with human, they will need improved mobility and movement capabilities. To achieve human-like movements, robots will have to become much more complex than they are today. The development of these enhanced robots represents an excellent challenge to researchers. However, increasing the complexity of robots' hardware and structure should not be done by neglecting reliability of those same robots. To avoid these problems, the simplification of robots mechatronics will be necessary. Although everyone working on robots acknowledges that, in reality, the design of the mechanical structure greatly affects the performance and controllability, general investigations of the relationship between a robot's mechanical structures and its controllability and reliability have been relatively scarce. This is a fundamental omission in the field of robotics. The research direction would be to find a unified method to create suitable mechanics for autonomous mobile robots that provide good dynamic performance, as well as simplicity and reliability.

6.5 Service robots – a disruptive technology in decades to come

Fig. 20 shows the technology road map of autonomous systems (SRI, 2008). Mobile robots are evolving from unmanned, remote controlled, semi-autonomous, to full autonomous systems. In this evolution, mobile robots require greater mobility, and higher intelligence.

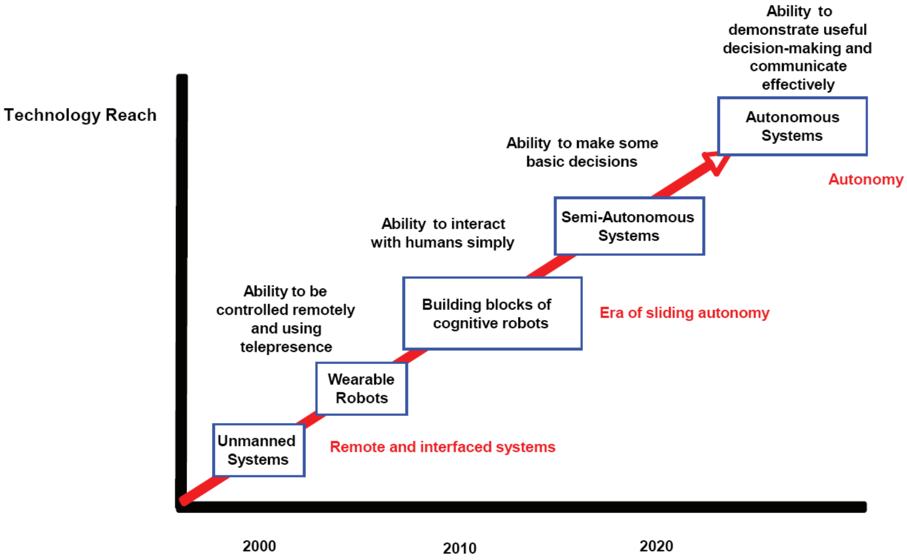

Technology Reach
Ability to demonstrate useful decision-making and communicate effectively
Autonomous Systems
Autonomy
Ability to make some basic decisions
Semi-Autonomous Systems
Ability to interact with humans simply
Building blocks of cognitive robots
Era of sliding autonomy
Ability to be controlled remotely and using telepresence
Wearable Robots
Unmanned Systems
Remote and interfaced systems
2000
2010
2020

Fig. 20. Technology Roadmap: Service Robotics (SRI, 2008)

The highly integrated systems comprising machines, sensors, computers, and software that have action and reasoning capabilities may relieve human from working in hazardous operations such as welding, plant dismantlement, to operate cutting tools from a safe distance. Robots could offer human-machine interfaces as easy to operate as the current personal computer. Fig. 21 shows a human augmented robotics welding system. It allows the operator to carry out welding remotely.

As the elderly population increases, there are greater and more pressing societal needs of health care and personal assistance at the society and family level. As more people live to the oldest ages, there may also be more who face chronic, limiting illnesses or conditions, such as arthritis, diabetes, osteoporosis, and senile dementia. According to OECD statistics (EURON, 2004), one third to one-half of health spending is for elderly people. Elderly people with varying limiting conditions become dependent on others for help in performing the activities of daily living. These needs call for much faster advancement of service robots that can assist elderly people to perform everyday activities such as bathing, getting around inside the home, and preparing meals.

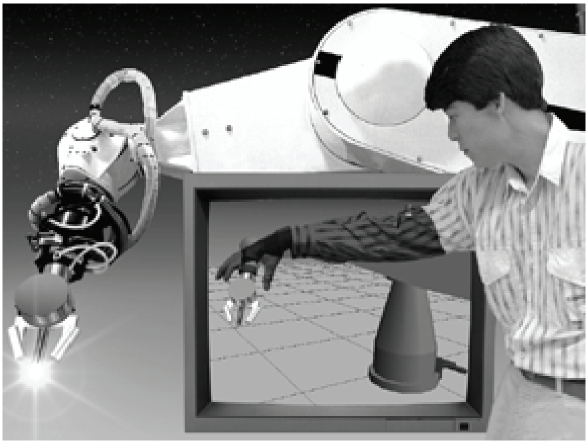


Fig. 21. Human augmented welding robot (http://www.sandia.gov/LabNews/LN03-12-99/robot_story.htm)

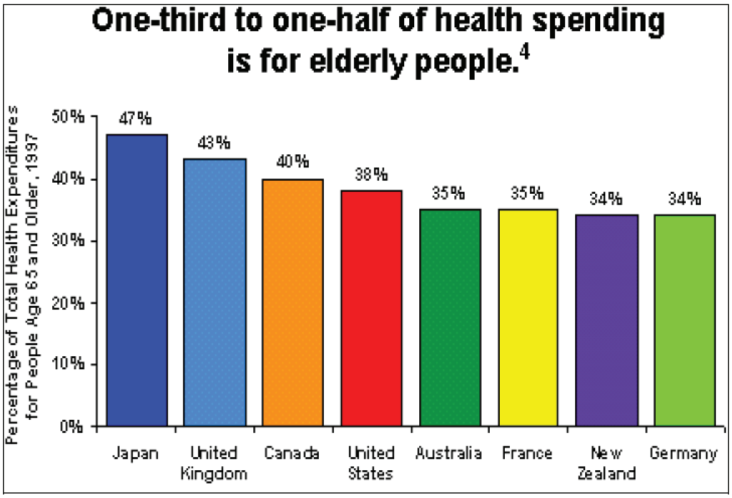


Fig. 22. Health spending quantities (Source: OECD)

According to the IFR (International Federation of Robotics), a service robot is a robot which operates semi or fully autonomously to perform services useful to the well being of humans and equipment, excluding manufacturing operations. Apart from caring for elderly people, service robots will expend into many aspects of the society. A robot nanny may look after children, providing more interactive learning environment (Figure 23.).

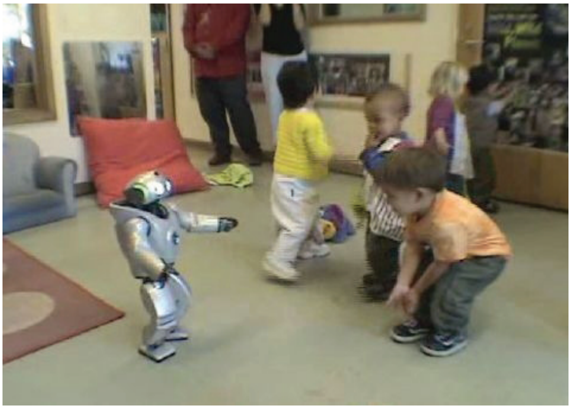


Fig. 23. Robot nannies look after children (blog.bioethics.net.)

Robots may replace nurses by performing jobs like dispensing drugs, taking temperatures and cleaning up wards (Figure 24.). The humanoid will make caring for patients cheaper and safer.

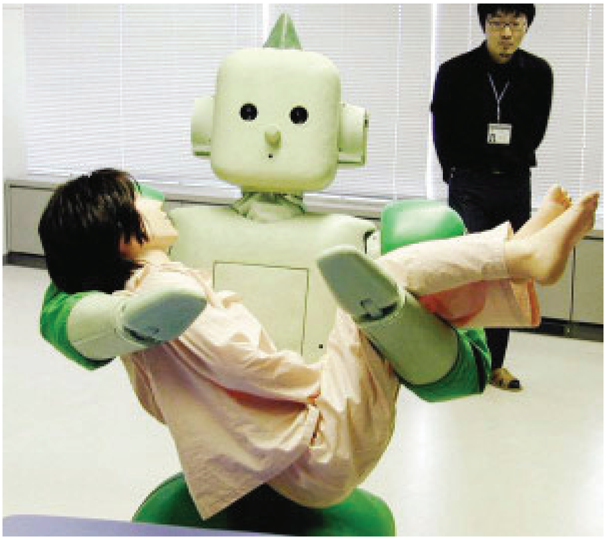


Fig. 24. A robot nurse carries a patient (bp2.blogger.com)

In developing the National Intelligence Council's Global Trends 2025, SRI Consulting Business Intelligence (SRIC-BI) was commission to identify six potentially disruptive civil technologies that could emerge in the coming fifteen years. The six disruptive technologies were identified through a process carried out by technology analysts from SRIC-BI's headquarters in Menlo Park, California, and its European office in Croydon, England. The full finds are published in the recent conference report (SRI, 2008). From 102 potentially disruptive technologies, the following six disruptive technologies have been identified:

Biogerontechnology
Energy Storage Materials
Biofuels and Bio-Based Chemicals
Clean Coal Technologies
Service Robotics
The Internet of Things.

Apart from military applications, the report noted the significant impact of service reports on elder-care applications. It is envisioned that the development of human-augmentation technologies will allow robots to work alongside humans in looking after and rehabilitating people. What is interesting is that some of technologies have dual uses. As opposed to human beings who convert food to energy, robots rely on external power supply. Energy storage, which is also identified as a disruptive technology, has high relevance to service robots as well.

7. Conclusions

Humankind has dreamt of robots being faithful slaves to carry various mundane or intelligent operations since the ancient times. Although there is no universally accepted definition for "robot", the key elements of a "robot" lie in "its ability to perform functions" and "its ability to move". To possess the two basic abilities, a robot requires intellect, actuation, mobility (statue and motivational), sensors, and communication. There is correspondence between robots and human in terms of these functional blocks. Robotics and robot intelligence researchers benefit from understanding of biological systems in developing biologically inspired robots that can emulate the naturally gifted abilities of these subjects.

Mobile robots, as opposed to fix based industrial robots, have huge potential to impact the society. The market for mobile robots is increasing. As such, there has been an explosion of research activities in mobile robots since the emergence of Shakey and Stanford Cart in 1970s. The field has witnessed a variety of attempts to develop mobile robots for critical missions, being on land, underwater or in the air. The applications range from military, to civil and service. Great progress has been achieved in design, perception and control. In terms of bio robots, many techniques have been developed to equip robots with efficient mobility and locomotion. Nevertheless, because of technological limitations, these systems mostly are still confined to laboratory exploitations. The deficiencies in current robotics technology have been key limiting factors in wider acceptance and adoption of mobile robots. Further research and development efforts are required to tackle many challenges and close the gap between proof-of-concept and actualisation.

Wide applications of robots require standardisation and rationalisation of programming languages, system architecture, mechanical/electrical/control interfaces for plug-and-play. The key issues are connectivity, modularity, portability and interchangeability.

The existing robots have limited intuitive learning ability. They have no or little ability to "grow", a natural gift in human. Once programmed to perform one function, robots have little self adaptation to a new function or a variation of the same function. A breakthrough in the theories of intuitive learning and observatory control would make the robots more human-like intellectually, not just physically. This is tied to the ability for a robot to gain contextual knowledge from interacting with environments. Information from the robot's surroundings gathered by multiple sensors could be used to help the robot to achieve its tasks and even determine future tasks. Furthermore robots grow intellectually so to operate in new environments.

At present, robots are still far more inferior to biological systems in terms of dexterity, coordination, lifting capacity, etc. Hardware technology, such as novel materials, actuation, and mechanical design holds the key to enhance the robot capabilities.

As the society is aging, there are pressing needs for service robots to help elderly people gain greater independence for activities encountered in daily life. In addition to military applications and exploitations, service robots will fulfil different roles in the society, being robot security guard, robot nannies, robot tour guide, robot nurse, etc. In decades to come, they may well emerge as a disruptive technology to impact the society greatly.

8. Acknowledgement

The authors wish to thank Mr. John Fleming for express ordering of newest reference books, and Christophe Tricaud and Mervin Chandrapal for their assistance in preparing this chapter.

9. References

Bekey G.; Ambrose R.; Kumar V.; Lavery D.; Sanderson A.; Wilcox B.; Yuh J. & Zheng Y. (2008). *Robotics: State of the Art and Future Challenges*, Imperial College Press, ISBN 978-1-84816-006-4.

Calisia D.; Iocchi L.; Nardia D.; Scalzoa C. M. & Ziparoa V. A. (2008). Context-based design of robotic systems, *Robotics and Autonomous Systems*, Vol. 56, No. 11, (November 2008), pp. 992-1003

Chao, H.; Cao, Y. & Chen Y. Q. (2007). Autopilots for Small Fixed-Wing Unmanned Air Vehicles: A Survey. *Proceedings of International Conference on Mechatronics and Automation, 2007*, pp.3144-3149, 5-8 Aug. 2007

Chen, X.Q., Gong, Z.M., Huang, H., Ge S.Z., and Zhou L.B. (2000a) “Process Development and Approach for 3D Profile Grinding and Polishing” in Advanced Automation Techniques in Adaptive Material Processing, Eds: X.Q. Chen, R. Devanathan & A.M. Fong, ISBN 981-02-4902-0, World Scientific, Singapore, pp. 19-54.

Chen, X.Q., Gong, Z.M., Huang, H., Ge S.Z., and Zhou L.B. (2002b) “Adaptive Robotic System for 3D Profile Grinding and Polishing” in *Advanced Automation Techniques in Adaptive Material Processing*, Book edited by: Chen, X.Q., Devanathan R., and Fong, A.M., ISBN 981-02-4902-0, World Scientific, Singapore, pp. 55-90.

Department of Defense (2007). Unmanned systems soadmap, 2007-2032, http://www.acq.osd.mil/usd/Unmanned%20Systems%20Roadmap.2007-2032.pdf

Delcomyn F. (2007) “Biologically Inspired Robots”, *Bioinspiration and Robotics: Walking and Climbing Robots*, Book edited by: Maki K. Habib, ISBN 978-3-902613-15-8, I-Tech, Vienna, Austria, EU, September 2007, pp. 279 – 300.

Ecole Polytechnique Federale de Lausanne (2008). Course on mobile robots, http://moodle.epfl.ch/course/view.php?id=261

EURON, European Robotics Research Network (2004), *EURON Research Roadmaps*, 23 April 2004.

Floreano D.; Godjevac J.; Martinoli A.; Mondada F. & Nicoud J. D. (1998). Design, Control, and Applications of Autonomous Mobile Robots, *Advances in Intelligent Autonomous Agents,* Boston : Kluwer Academic Publishers, 1998

Laumond, J. P. (1999). Robot Motion Planning and Control. http://www.laas.fr/~jpl/book.html

Li, Z. & Canny, J. F. (1993). Nonholonomic Motion Planning, *The Springer International Series in Engineering and Computer Science*, Vol. 192, ISBN: 978-0-7923-9275-0

Liu S. C.; Tan D. L.; & Liu G. J. (2007). Formation Control of Mobile Robots with Active Obstacle Avoidance, Acta Automatica Sinica, Vol. 33, No. 5, (May 2007), pp. 529-535,

Kara, D. (2006). Global trends in the consumer robotics market, http://www.roboticstrends.com/

Kaikkonen J.; Mikelainen T. & Hakala H. (1991). Advanced Design Methods and Tools for Mobile Robot Development, *In Proceedings of IEEE/RSJ International Workshop on Intelligent, Robots and Systems* IROS '91. Nov. 3-5, 1991

Kim, S., Spenko, M., Trujillo, S., Heyneman, B., Santos, D., Cutkosky, M.R. (2008) "Smooth Vertical Surface Climbing With Directional Adhesion", IEEE Transactions on Robotics, Volume: 24, Issue: 1, pp. 65-74

Nassiraei A. & Ishiia K. (2007). Concept of Intelligent Mechanical Design for Autonomous Mobile Robots, *Journal of Bionic Engineering,* Vol. 4, No. 4, December 2007, pp. 217-226

Ohio State University, College of Engineering (1923) "Pilotless Plane", Ohio State Engineer, vol. 6, no. 3 (April, 1923), page 19. https://kb.osu.edu/dspace/handle/1811/32748

Patnaik, S. (2007). *Robot Cognition and Navigation – An Experiment with Mobile Robots,* Springer, ISBN: 978-3-540-23446-3, Berlin Heidelberg New York.

Shen, W., Gu, J., and Shen, Y. (2005) "Proposed wall climbing robot with permanent magnetic tracks for inspecting oil tanks," in Proc. IEEE Int. Conf. Mechatronics and Automation, Niagara Falls, Canada, July 2005, pp. 2072-2077.

SRI Consulting Business Intelligence, under the auspices of the National Intelligence Council (2008), *Disruptive Civil Technologies - Six Technologies with Potential Impacts on US Interests out to 2025*, Conference Report CR 2008-07, April 2008.

Wagner, M., Chen, X.Q., Wang, W.H. and Chase J.G. (2008), "A novel wall climbing device based on Bernoulli effect", Proc 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA08), ISBN: 978-1-4244-2368-2, Beijing, China, October 12-15, pp. 210-215.

WinterGreen Research, Inc. (2007), http://wintergreenresearch.com/

Zhao, Y., Fu, Z., Cao, Q., and Wang, Y. (2004) "Development and applications of wall-climbing robots with a single suction cup", Robotica, Cambridge University Press, Volume 22, pp. 643–648.

2

A Field Robot with Rotated-claw Wheels

Ronggang Yue, Jizhong Xiao[2]*, Kai Li, Jun Du, Shaoping Wang
[1]*Beihang University*
China
[2]*City University of New York*
United States of American

1. Introduction

With the development of space science and technology, some countries have launched unmanned planetary exploration programs. Because planet surfaces are rough and formidable, planetary exploration rovers were used as a versatile and safe alternative to manned space missions to explore the planets. In recent years, many types of planetary rovers were developed including tracked type, legged type, and wheeled type, etc. (Li et al., 2005). Because wheeled rovers are well suited to provide smooth motion and offer a high payload capacity, they become the most popular and the mainstream in current rovers (Salemo et al., 2002).

In order to deal with the rough terrains of planetary surfaces, researchers put most of the efforts in designing new structure of rover body, but give less attention to new Types of wheels. Currently, three main classes of wheels are used in planetary rovers as follows:

(1) Representations are the wheels of Mars exploration rover Spirit (Lindemann et al., 2006) developed by JPL and Japanese rover Micro5 (Takashi et al., 2003). This type of wheel has a circular shape with a lot of transverse striae spaced evenly around the wheel to increase the friction between ground and the wheel. The wheels are commonly used in most of the rovers that can move smoothly and turn around agilely. But its fatal disadvantage is that it can not scale the step whose height is larger than the radius of the wheel, also it can not cross the ditch whose width is larger than the diameter of the wheel.

(2) Flexible flake wheel. The representation is the wheel of Mars rover made by Toshiba in Japan (Liu et al., 2002). This kind of wheel is circular too, but the difference is that there are lots of flexible flake spaced evenly around the wheel. Compared with the first type wheel, this wheel increases buffering, and it can traverse over the step whose height is larger than the radius of the wheel. However, it can not cross the ditch whose width is larger than the diameter of the wheel.

(3) Multiple planetary wheel. A representative example is the wheel of Lunar rover developed by Harbin Institute of Technology (HIT) in China. The lunar rover uses planetary wheel which consists of three small circular wheels (Deng et al., 2003; Deng et al., 2004). When the lunar rover moves on flat terrain, only two small wheels touch ground. Once it encounters a big obstacle or step, the three small wheels revolve around the public center of the wheels to traverse over the obstacle. This type of wheel overcomes the shortcomings of

conventional circular wheels, and can climb over the step that is higher than diameter of the small wheel. However, this type of wheel makes the robot hard to make turns.

In this paper, we introduce a field robot (named as Rabbit) using the rotated-claw wheel have demonstrated the superiority over the conventional wheels. The analysis and testing results show that the innovative rotated-claw wheel can move steadily, turn around agilely, and climb the step whose height is larger than the radius of the wheel which has demonstrated.

2. Design of the Rabbit

The basic requirements for an excellent field robot are motion stability, and strong capability of climbing obstacles or slopes. We designed a field robot prototype named as Rabbit whose motion is based on 4 rotated-claw wheels attached by "bogie levers" to the chassis. Figure 1 shows its electrical diagram and Figure 2 shows the picture of the robot whose weight is 10.5Kg and outline dimension is 57cm×43cm×31cm. The body of Rabbit is mounted to the rocker through a differential mechanism. Rabbit is driven by 8 DC brushed motors that are controlled by a DSP controller. Each motor drives a series of stages of gearing that result in the final torque-speed relationship of the actuator. The total gear reduction rate for each actuator is 246:1. An encoder on each motor is utilized for sensing the rotation speed and determining steering angle. Under average load conditions, the wheel rotates at 24.4 rounds per minute, which results in a nominal vehicle speed of 15.3 cm/s with the 12 cm diameter wheels. In order to study the motion performance, an accelerator and an inclinometer are installed in the robot prototype.

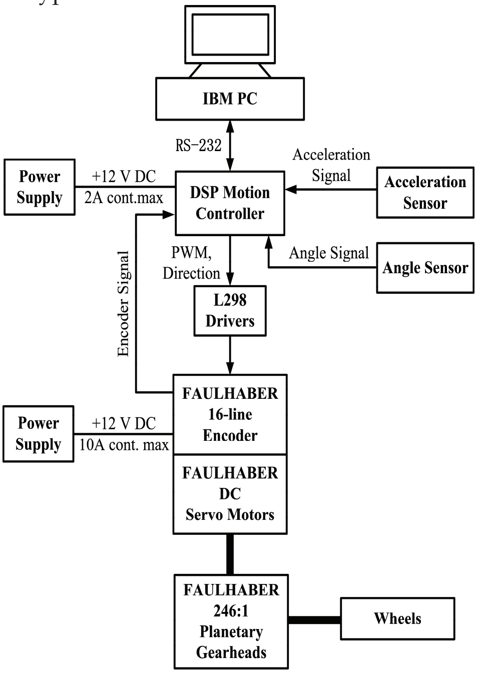


Fig. 1. Schematic of Rabbit's electrical diagram

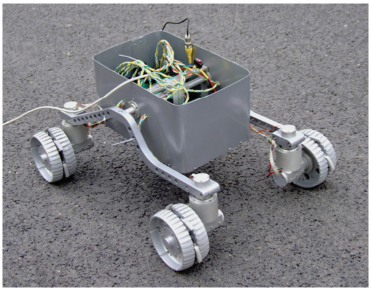


Fig. 2. The field robot prototype named as Rabbit

Due to the rotated-claw wheel is an innovative design. We would like to introduce principle and design of the wheel in detail.

3. Design of rotated-claw wheel

3.1 Principle of the rotated-claw wheel

The schematic diagram of the rotated-claw wheel is shown in Figure 3. In order to ensure the motion stability, turning flexibility, and obstacle-climbing capability, the wheel is designed as traditional circular structure with six sets of claw mechanism evenly installed around the wheel's outer skirt. Each set of claw mechanism consists of claw, claw axle, gag lever post and draught spring. Each claw can swing round its axle. The claw is in open state under the effect of draught spring and gap lever post when it doesn't touch the ground, and the claw swings into the wheel body under the effect of rover's weight when it touches the ground, which improves bumpy motion caused by hexagon effect.

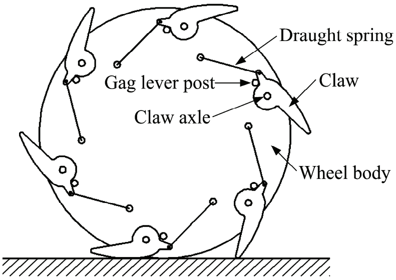
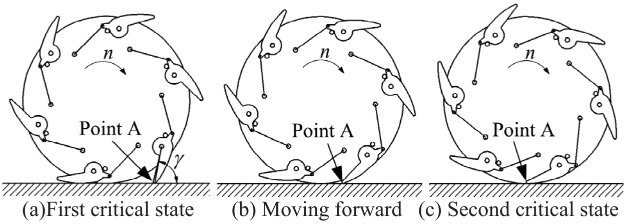


Fig. 3. Schematic diagram of the rotated-claw wheel

3.2 Motion stability analysis

As shown in figure 3, the claw can swing into the wheel body smoothly when the wheel rotates in anticlockwise direction, and the hexagonal effect of the wheel gives little influence on the stability of the rover. However, when the wheel rotates in clockwise rotation, the claw may disturb the stability of the wheel and cause the rover jounce. Therefore, we must analyze the stress of the wheel and select design parameters carefully to avoid this problem in the case of clockwise rotation.

When design the wheel, we first determine the claw dimensions and the distance between each claw axle and wheel's center, then adjust the position of gag lever post. Figure 4 shows the process of a claw touching the ground when the wheel rotates in clockwise direction on flat terrain. There are two cases about angle γ in Figure 4. One is the case when angle γ is larger than 90°, the other is the case when angle γ is less than 90°. We will discuss the two cases respectively.



(a)First critical state (b) Moving forward (c) Second critical state

Fig. 4. Rotation in clockwise direction on flat terrain of rotated-claw wheel

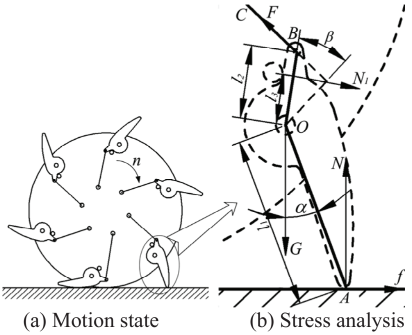
3.2.1 When angle γ is larger than 90°

Figure 5(a) shows the critical state when a claw tip touches the ground while the wheel leaving the ground. At this time instance only point *A* supports the wheel, i.e., only point *A* gets supporting force from the ground. Taking the claw touching the ground as an object, we can get the stress analysis shown in Figure 5(b), in which the wheel and the claw are expressed in dashed lines and the claw is simplified as line segments *AOB* and the draught spring is simplified as line *BC*.

Stress analysis in Figure 5(b) shows that the claw satisfies the following equation:

$$Nl_1 \sin\alpha + fl_1 \cos\alpha + Fl_2 \cos\beta = N_1 l_3 \tag{1}$$

where, *N* is the acting force of ground to wheel at point *A*; N_1 is the acting force of gag lever post to claw; l_1 is the length of line *OA*; l_2 is the length of line *OB*; l_3 is the level distance of force N_1; *a* is the angle between line *OA* and plumb line; β is the angle between line *OB* and the line perpendicular to line *BC*; *f* is the friction between point *A* and ground; *F* is the pre-setting pulling force of spring *BC*.



(a) Motion state (b) Stress analysis

Fig. 5. Stress analysis when angle γ is larger than 90°

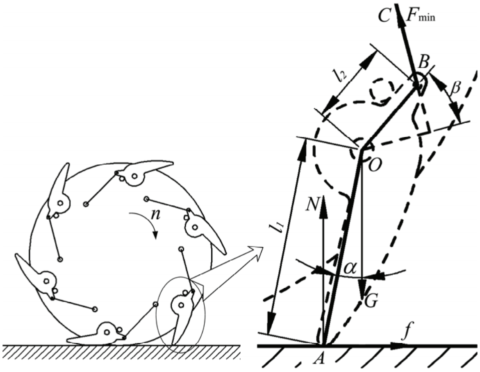
Because it is in a balance state, the claw cannot swing around point O should angle γ is larger than 90°, and thus the claw cannot swing into the wheel body. Now we get the first condition ensuring the claw swing into the wheel body smoothly is that angle γ must be less than 90°. We can adjust numerical values of l_1, l_2, l_3, a, β to make γ be less than 90°.

3.2.2 When angle γ is less than 90°

There are two critical states as illustrated in Figure 4(a) and Figure 4(c).

(1) First critical state

Figure 6 shows the First critical state-when the wheel is on the point of leaving the ground, in which a claw tip touches the ground, and its weight acts on point A. At this time instance, only point A supports the wheel. Taking the claw touching the ground as an object, we can get the stress analysis shown in Figure 6(b), in which the wheel and the claw are expressed in dashed lines, the claw is simplified as line segments AOB and the draught spring is simplified as line BC.



(a) Motion state (b) Stress analysis

Fig. 6. Stress analysis at the first critical state

In Figure 6, the claw touching the ground must swing around the point O clockwise if the wheel is to rotate clockwise smoothly. In order to satisfy this requirement, every claw must satisfy the following equations:

$$\begin{cases} Nl_1 \sin\alpha > F_{\min} l_2 \cos\beta + fl_1 \cos\alpha \\ N \approx G \\ F_{\min} = k\Delta x_{\min} \\ f = \mu N \end{cases} \tag{2}$$

where, N is the acting force of ground to wheel at point A; l1 is the length of line OA; l_2 is the length of line OB; a is the angle between line OA and plumb line; β is the angle between line OB and the line perpendicular to line BC; $F_{\min}$ is the pre-setting pulling force of spring BC; f is the friction between point A and ground; G is the gravity force acting on the single wheel from the rover that is approximate to N; k is the elastic coefficient of spring BC; $\triangle x_{\min}$ is the extension of spring BC; μ is the coefficient of sliding friction between point A and ground.

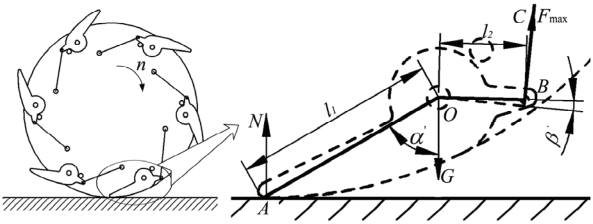
Then we can get the following inequation:

$$Gl_1 \sin\alpha > k\Delta x_{\min} l_2 \cos\beta + \mu G l_1 \cos\alpha \tag{3}$$

Inequation (3) is the first criterion that makes the claw swing into the wheel body freely and ensures the wheel rotate clockwise smoothly.

(2) Second critical state

The second critical state is shown in Figure 7(b), when the excircle of the claw is tangent to the road surface, in which its weight acts on point *A* and the other part of the wheel doesn't touch the ground.



(a) Motion state (b) Stress analysis

Fig. 7. Stress analysis at the second critical state

In Figure 7, the claw must be inside of the profile of the wheel body if the wheel is to rotate clockwise smoothly. Every claw must satisfy the following equations:

$$\begin{cases} F_{\max} l_2 \cos\beta' < N l_1 \sin\alpha' \\ N \approx G \\ F_{\max} = k\Delta x_{\max} \end{cases} \tag{4}$$

where, $F_{\max}$ is the pulling force of spring *BC*; l_1 is the length of line *OA*; l_2 is the length of line *OB*; α' is the angle between line *OA* and plumb line; β' is the angle between line *OB* and the line perpendicular to line *BC*; *N* is the acting force of ground to wheel at point *A*; *G* is the gravity force of the single wheel; *k* is the elastic coefficient of spring *BC*; $\triangle x_{\max}$ is the extension of spring *BC*.

Then we can get the following inequation:

$$k\Delta x_{\max} l_2 \cos\beta' < G l_1 \sin\alpha' \tag{5}$$

Inequation (5) is the second criterion that makes the claw swing into the wheel body freely and ensures the wheel rotate clockwise smoothly.

In summary, when designing the rotated-claw wheel, the distance between point *O* and the center of the wheel should be chosen appropriately. When the claw touches the ground, the angle γ should be less than 90° as shown in Figure 5(a). At the same time, appropriate numerical values of *G*, l_1, l_2, α, β, α', β', $\triangle x_{\min}$, $\triangle x_{\max}$ should be chosen to satisfy inequations (3) and (5). These criteria ensure the wheel rotate in clockwise direction steadily and reduce jounce (Yue et al., 2007).

3.3 Obstacle-climbing analysis

Conventional circular wheels cannot scale the step whose height is larger than the radius of the wheel. In Figure 8, the step height is H_1, and the radius of the conventional circular wheel is R. If the wheel wants to traverse over the step, H_1 must be less than R. The rotated-claw wheel overcomes this shortcoming. In Figure 9, the step height is H_2, and the radius of the rotated-claw wheel is still R. However, due to the existence of the claw, wheel can scale the step whose height is larger than the radius of the wheel by h, and h can be designed larger than zero by choosing suitable numerical values of l_1, l_2, α, β, and distance between point O and wheel's center. From the analysis, we can claim with confidence that the obstacle-climbing capability of the rotated-claw wheel is improved in comparison with the conventional circular wheels. Real experimental results will verify the performance.

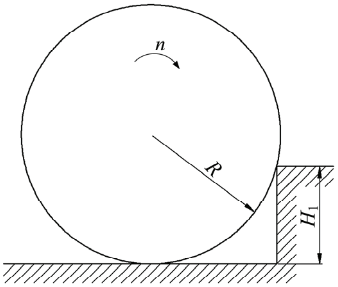


Fig. 8. Sketch drawing of circular wheel climbing a step

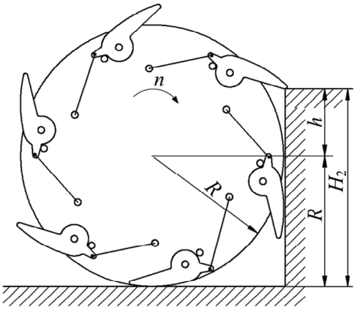


Fig. 9. Sketch drawing of rotated-claw wheel climbing a step

3.4 Structure design of rotated- claw wheel

Based on the criteria mentioned above, this paper designed the three-dimensional model of the rotated-claw wheel shown in Figure 10 that includes a steering mechanism and the obstacle-climbing mechanism (rotated-claw). In order to enhance the friction between the wheel and ground, grooves are constructed around the wheel in even space. The deigned diameter of the wheel is 120 mm and its width is 60 mm. Dimensions of the claw is shown in Figure 11, which make γ be less than 90°.

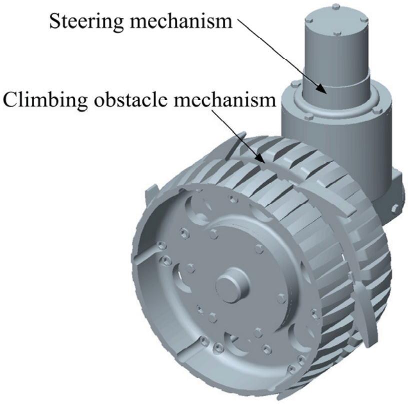


Fig. 10. Three-dimensional model of the rotated-claw wheel

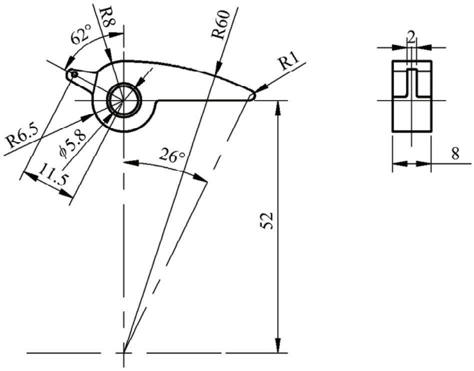


Fig. 11. Dimensions of the claw

4. Performance test of Rabbit

In order to test environmental adaptability of Rabbit, the experiments are conducted on various terrain, including surfaces of bituminous macadam, dry soil, step, slope, and simulated lunar soil. All of the data were collected when Rabbit moves at full speed..

4.1 Motion stability performance

The motion stability analysis reveals that the six claws may cause a little jounce when the rotated-claw wheel rotates in clockwise direction, while smooth motion is guaranteed in anticlockwise rotation. We conducted experiments to evaluate the motion stability performance by commanding the Rabbit robot move on flat ground of bituminous macadam. Figure 12 shows the acceleration curve of Rabbit in plumb direction when the rotated-claw wheels rotate in clockwise direction. The acceleration varies from -0.39g to 0.46g. Figure 13 gives the acceleration curve of Rabbit in plumb direction when the wheels make anticlockwise rotation, it shows that the acceleration changes from -0.13g to 0.16g. Figure 14 gives the acceleration curve in plumb direction when Rabbit rotates with the claws retracted inside the wheel body (Now the rotated-claw wheel is the same as conventional circular wheel). It shows that the acceleration varies from -0.13g to 0.13g. Compare Figure 13 with Figure 14, we can see that stability of the rotated-claw wheel under the condition of retracted claws is similar to that of conventional circular wheel.

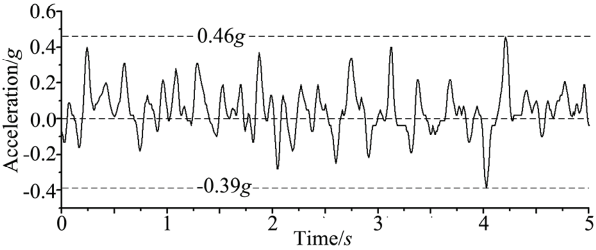


Fig. 12. Plumb direction acceleration curve of Rabbit when the rotated-claw wheels make clockwise rotation on bituminous macadam ground

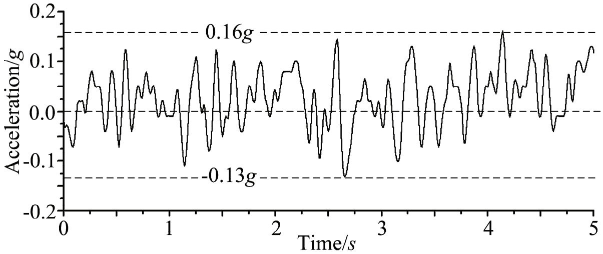


Fig. 13. Plumb direction acceleration curve of Rabbit when the rotated-claw wheels make anticlockwise rotation on bituminous macadam ground

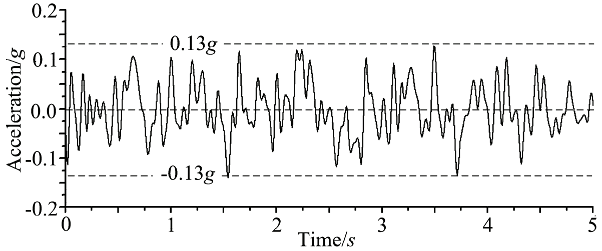


Fig. 14. Plumb direction acceleration curve of Rabbit when the rotated-claw wheels rotate with retracted claws on bituminous macadam ground

It is obvious that the motion stability under anticlockwise rotation is more stable than that under clockwise rotation. The reason is that the claw can swing into the wheel body under anticlockwise rotation while the hexagon effect causes the bumpiness under clockwise rotation. So the Rabbit should be commanded to move in a backward mode (i.e., all the wheels rotate in anticlockwise direction) on flat hard ground.

4.2 Performance of climbing obstacles

4.2.1 Dry soil terrain

In order to test Rabbit's motion performance on dry soil terrain with multi-obstacle, we did another experiment as shown in Figure 15. Figure 16 shows the acceleration curve of Rabbit in plumb direction that denotes the acceleration varying from -0.125g to 0.125g. Figure 17 gives the acceleration curve of Rabbit in plumb direction on dry soil when the wheels rotate in clockwise direction. It shows that the acceleration varies from -0.10g to 0.10g. Figure 18 gives the acceleration curve of Rabbit in plumb direction on dry soil when the Rabbit moves under the condition of retracted claws, which shows the acceleration varies from -0.10g to 0.10g. Compare Figure 17 with Figure 18, we can see that stability of the wheel is as good as conventional circular wheel under the condition of retracted claws.

It is obvious that the backward mode is smoother than forward mode (i.e., all the wheels rotate in clockwise direction) when Rabbit operates on dry soil. But the two results are approximative. The reason is that the claw can sink into soil and the obstacle-climbing capability is enhanced. So Rabbit should move in a forward mode when operates on dry soil terrain with multi-obstacle. The highest obstacle on dry soil terrain that the robot can climb over is 13cm. The experiments also show that Rabbit can step over the clod or stone whose dimension is equivalent to the diameter of the wheel.

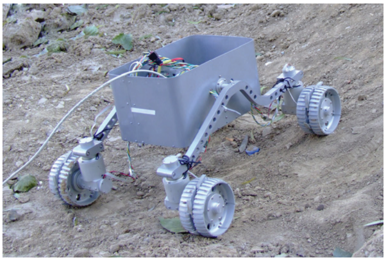


Fig. 15. Rabbit moves on dry soil terrain

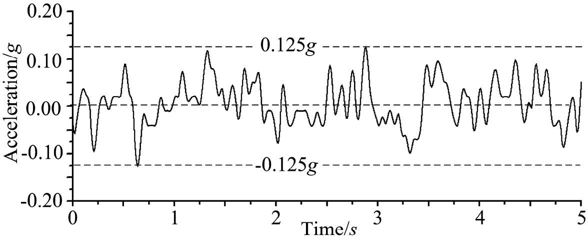


Fig.16. Plumb direction acceleration curve of Rabbit while the robot moves forward on dry soil terrain

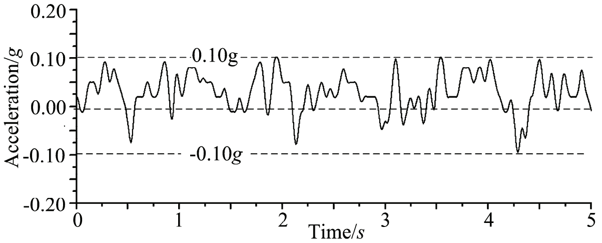


Fig. 17. Plumb direction acceleration curve of Rabbit while the robot moves backward on dry soil terrain

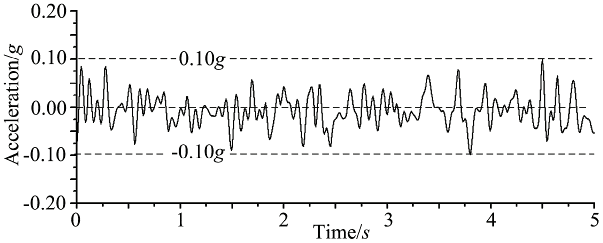


Fig. 18. Plumb direction acceleration curve of Rabbit while wheels rotates under the condition of retracting claws on dry soil terrain

4.2.2 Step terrain

When the robot moves on steps terrain, Rabbit should move in a forward mode (i.e., all the wheels rotate in clockwise direction), because the claw can catch step in front of the wheel and help the robot to climb over it easily in the forward mode. Table 1 shows the experimental results in different step height.

Step height/cm	2.2	3.9	6.3	8.1	9.0
Result	Success	Success	Success	Success	Fail

Table 1. Experimental results on different height step

It is obvious that the rotated-claw wheel can climb over the 8.1cm step that is almost 1.35 times of wheel's radius as shown in Figure 19. This verifies that the rotated-claw wheel can improve the obstacle-climbing capacity.

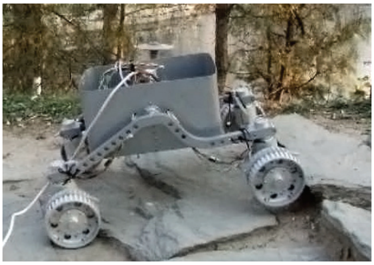


Fig. 19. Climbing step in a forward mode

4.2.3 Slope terrain

In order to test Rabbit's motion performance on slope terrain, we did other experiments as shown in Figure 20, in which the Rabbit climbs over slope terrain in a forward mode.

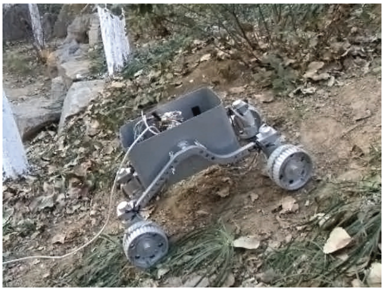


Fig. 20. Rabbit climbs over slope terrain in a forward mode

Figure 21 and Figure 22 show the angle curve when Rabbit climbs slope terrain in forward mode and backward mode respectively. We can see that Rabbit can climb a slope up to 40° in the forward mode, in contrast, Rabbit is able to climb a slope just up to 31° in backward mode.

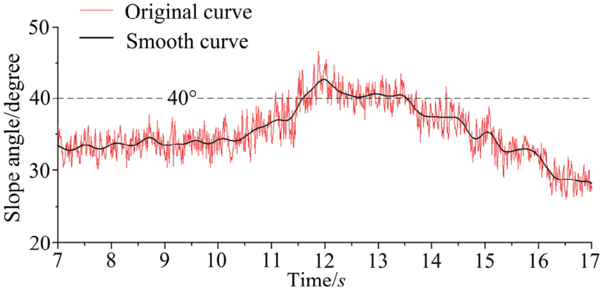


Fig. 21. Angle curve when Rabbit climbs slope terrain in forward mode

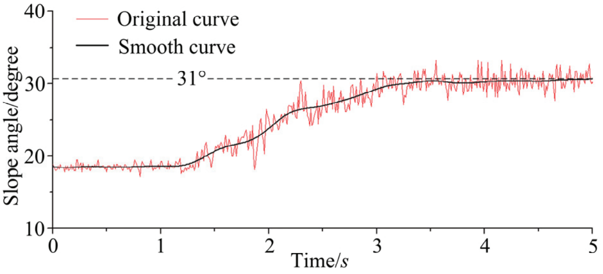


Fig. 22. Angle curve when Rabbit climbs slope terrain in backward mode

Comparing Figure 21 and Figure 22, the rotated-claw wheel increases the climbing slop angle up to 9 degree. The reason is that the claw can sink into soil in motion, which enhances physical attraction between the wheel and ground. So Rabbit should move in a forward mode when it moves on slope terrain.

4.2.4 Lunar soil simulation

In order to adapt to the utilization in planetary, we did experiments on simulated terrain of lunar soil whose material is pozzuolana. The lunar soil is loaded in a trough which has dimensions of 300cm×80cm×60cm as shown in Figure 23. Void ratio (It is defined as the ratio of the volume of all the pores in a material to the volume of all the grain) of the lunar soil is approximately from 0.8 to 1.0, and density of the grain is 2.77g/cm3.

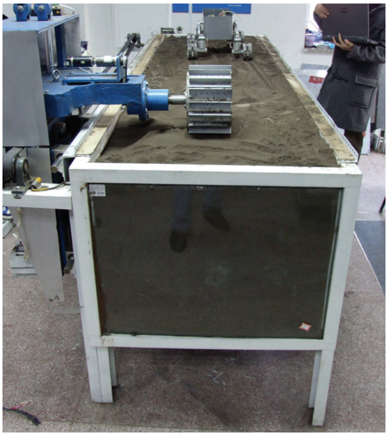


Fig. 23. Trough for lunar soil simulation

We tested Rabbit's motion performance on rough terrain and multi-obstacle terrain made up of lunar soil as shown in Figure 24 and Figure 25. The result shows that Rabbit can move freely on simulated lunar soil.

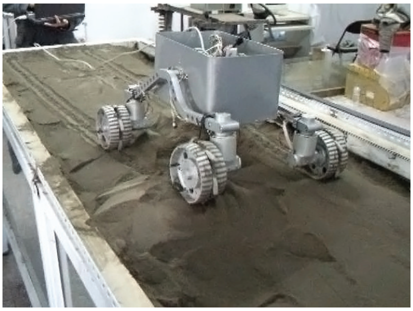


Fig. 24. Experiment on rough terrain

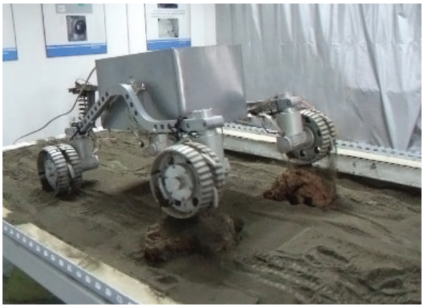


Fig. 25. Experiment on multi-obstacle terrain

In addition, we tested Rabbit’s horizontal pulling capacity on simulated lunar soil in both forward and backward modes (Figure 26). The experimental results show that Rabbit can generate maximum pulling forces of 26.5N in forward mode, and 25.1N in backward mode.

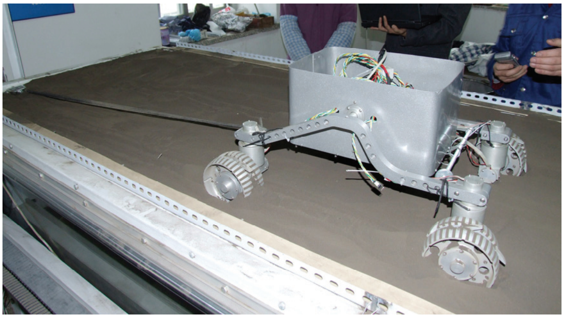


Fig. 26. Rabbit’s horizontal pull testing

5. Performance comparison

According to the available data from literature, we compare the performance of Rabbit with MFEX and Spirit robots. MFEX (Microrover Flight Experiment) was a small rover designed by JPL (Jet Propulsion Laboratory) in 1990s. It was launched to Mars in December 1996 [9]. Spirit is one of the latest Mars rovers designed by JPL. It landed on Mars on January 4, 2004 [10], and finished exploration mission with flying colors in the following years (and still alive). Table 2 lists the data comparison among Rabbit, MFEX, and Spirit. From the table, we can see that maximum slope the Rabbit can climb is larger than that of the other two rovers although Rabbit only equipped with 4 rotated-claw wheels (2 wheels less than the other rovers). In addition, Rabbit can climb over step which is higher than the radius of wheel.

Robot name	Rabbit	MFEX	Spirit
Mass	10.5Kg	9Kg	176.5Kg
Dimensions	57 cm×43 cm×30.9cm	63 cm×48 cm×28cm	140 cm×120 cm×150cm
Chassis type	Body mounted to rocker through a differential	Body mounted to rocker through a differential	Body mounted to rocker through a differential

Suspension system	Springless suspension	Springless suspension	Rocker-bogie suspension
Locomotion system	4 wheels (four steerable)	6 wheels (outer four steerable)	6 wheels (outer four steerable)
Maximum speed	0.153m/s	0.02m/s	0.046m/s
Operational range	1Km	10m	1Km
Layout of wheels	Claw-wheel with 120 mm diameter 60 mm width	wheels with 130 mm diameter 60 mm width	250 mm diameter
Motion control processors	One 2407A DSP	One Intel 80C85	
Max. step height	13cm (Can climb over the step whose height is 1.35 times higher than the radius of the wheel)	Less than 6.5cm	Less than 12.5cm
Maximum slope	40 ° in forward mode, 31 ° in back mode (in soft soil)	32 ° (dry sand) 17 ° (lunar soil simulant)	16 ° at least 30 ° in the nature of the Mars soil and terrain

Table 2. Performance comparison of Rabbit, MFEX, and Spirit

6. Conclusion

In this paper, we introduce a field robot using the rotated-claw wheel that has strong capacity of climbing obstacles. The experimental results demonstrate that Rabbit can move in different terrain smoothly and climb over step of 8.1cm and slop of 40°. The Rabbit can adopt different moving modes on different terrains.

(1) Rabbit should move in backward mode on flat hard ground.

(2) Rabbit should move in forward mode on rough, slop, and step terrains.

Because the rotated-claw wheel overcomes the disadvantages of conventional mobile robot wheels, it provides a better solution for field and planetary robots.

7. Acknowledgment

We thank Wen Li, Gang Sun, and Peng Sun of Beihang University for their valuable help in the experiments of lunar soil simulation.

8. References

Cuilan Li; Peisun Ma; Xueguan Gao & Zhikui Cao. (2005). A new six-wheel lunar robot for uneven surface. *Drive System Technique*, Vol. 19, No. 1, (Mar. 2005) page numbers(9-13), 1006-8244 (in Chinese)

Alessio Salemo; Svetlana Ostrovskaya & Jorge Angeles. (2002). The Development of Quasiholonomic Wheeled Robots, *Proceedings of the 2002 IEEE international Conference on Robotics and Automation*, Vol.4 , pp. 3514 – 3520, Washington, DC, May. 2002.

Randel A. Lindemann; Donald B. Bickler; Briand. Harrington; Gary M. Ortiz & Christopher J. Voorhees. (2006). Mars exploration rover mobility development. *Robotics & Automation Magazine*, IEEE, Vol. 13, No. 2, (Jun. 2006) page numbers (19-26), 1070-9932.

Takashi Kubota; Yoji Kuroda; Yasuharu Kunii & Ichiro Nakatani. (2003). Small, light-weight rover Micro5 for lunar exploration. *Acta Astronautica*, Vol. 52, No. 2-6, (Jan.-Mar. 2003) page numbers (447-453), 0094-5765.

Fanghu Liu; Jianping Chen; Peisun Ma & Zhikui Cao. (2002). RESEARCH STATUS AND DEVELOPMENT TREND TOWARDS PLANETARY EXPLORATION ROBOTS. *Robot*, Vol. 24, No. 3, (May. 2002) page numbers (268-275), 1002-0446. (in Chinese)

Zongquan Deng; Haibo Gao; Ming Hu & Shaochun Wang. (2003). Design of lunar rover with planetary wheel for surmount obstacle. *Journal of Harbin Institute of Technology*, Vol. 35, No. 2, (Feb. 2003) page numbers (203-213), 0367-6234. (in Chinese)

Zongquan Deng; Haibo Gao; Shaochun Wang & Ming Hu. (2004). Analysis of climbing obstacle capability of lunar rover with planetary wheel. *Journal of Beijing University of Aeronautics and Astronautics*, Vol. 30, No. 13, (Mar. 2004) page numbers (197-201), 1001-5965. (in Chinese)

Ronggang Yue; Shaoping Wang; Zongxia Jiao & Rongjie Kang. (2007). Design and performance simulation of a new type wheel with claws. *Journal of Beijing University of Aeronautics and Astronautics,* Vol. 33, No. 12, (Dec. 2007) page numbers (1408-1411), 1001-5965. (in Chinese)

K. Schilling & C. Jungius. Mobile robots for planetary exploration. (1996). *Control Engineering Practice*, Vol. 4, No. 4, (Apr. 1996) page numbers (513–524), 0967-0661. (in Chinese)

Glenn Reeves & Tracy Neilson. (2005). The Mars Rover Spirit FLASH Anomaly. *Aerospace Conference*, 2005 IEEE, pp. 4186-4199, Mar. 2005.

3

Mobile Wheeled Robot with Step Climbing Capabilities

Gary Boucher, Luz Maria Sanchez
Louisiana State University, Department of Chemistry-Physics
Shreveport LA, USA

1. Introduction

The field of robotics continues to advance towards the ultimate goal of achieving fully autonomous machines to supplement and/or expand human-performed tasks. These tasks range from robotic manipulators that replace the repetitious and less precise movements of humans in factories and special operations to complex tasks which are too difficult or dangerous for humans. Thus an important and ever-evolving area is that of mobile robots.

Extensive research has been done in the area of stair-climbing for mobile robotics platforms. Humanoid, wheeled, and tracked robots have all been made to climb stairs, however in most of these cases robots where designed for two dimensional operations and then later utilized or modified for stair climbing. (Herbert, 2008) Although strides have been made into exotic forms of legged robots, the conventional methods, such as wheels or tracks still form the basis for robotic locomotion.

The wheeled mobile systems are useful for practical application compared with the legged systems because of the simplicity of the mechanisms and control systems and efficiency in energy consumption (Masayoshi Wada 2006). To better understand the problems faced by mobile ground based robots one must understand the expected terrain that the machine must negotiate. This can range from un-level ground to rocky and irregular terrain and in some cases man-made obstacles such as steps or stairs must be climbed. Each of these applications has unique challenges and solutions.

In 2003, Louisiana State University-Shreveport took on the task to create an alternate approach to a rugged terrain robot capable of traversing not only rough terrain, but also man-made obstacles, such as steps and stairs, with the intent to meet the requirement to ascend and descend between levels in a building as in the case of security robots performing their tasks. The project further addressed the issue of observation capabilities to handle obstacles in the robot's path.

In conjunction with our Computer Science CSC 410 course in robotics, the LSUS Department of Chemistry-Physics took up the challenge to develop a robotic design that would meet these requirements. The criteria that factored into the initial concept phase of the project were the following: First the robot must be robust, capable of extended service in rugged environments and carry its own power source. Secondly, the robot must also have versatile vision systems which can relay the video information back to the operator via radio signals or a fibre optic link. Thirdly, the device should have the ability to climb steps and stairs for changing floors in a building. The challenge was then handed to the students

A robot using more than four wheels could compete with some tracked devices if the wheels are driven simultaneously. One approach considered to meet this requirement was through the use of a hydraulic motor on each wheel. This would allow all wheels to derive their rotation from one single power source. A central hydraulic pump generating a constant flow of fluid could provide the source to power the device. This concept was first patented by Joseph Joy in 1946 (Joy, 1946). Joy described a 16 wheel automobile capable of being driven by 8 hydraulic motors powered by a single engine and hydraulic pump.

Such a scheme for driving a robot would require two hydraulic pumps and two sets of motors, one set for the left and one for the right side of the robot. The differential drive would then allow turning much the same way as tank treads. The motors could be in series on each side and therefore produce the same rotation for the volume of fluid pumped. Hydraulic pumps and motors were ruled out in the LSUS robot due to cost and the shear bulk of two hydraulic systems with proportional rate of flow control.

The concept of wheel sets that can rotate is also not new. As far back as 1932 Raphael Porcello patented their use in numerous mobile devices from baby carriages to landing gear for airplanes (Porcello, 1932). Although not driven, these wheel sets demonstrated the versatility of allowing wheels to be grouped together and have their individual axels fixed at a certain common radius from the axes of wheel set rotation.

The LSUS design consensus centered on using sets of two wheels that used parallel individual axels each offset a given radius from the wheel set axis of rotation. In this way, the wheels could revolve and also be powered from a source of angular speed and torque. The wheels sets could also revolve in any direction independent of the rotation of the wheels. This design seemed to satisfy the primary requirements for the robot for both rough terrain and stair climbing.

2. Related Work

In 1991 King et al patented a method of stair climbing using a robot with rotating wheel sets (King et al, 1991). This device used two sets of two wheels each for stepping and used a larger front wheel to ride up and over oncoming steps. This larger wheel was forced by the rotating rear wheel sets. This novel approach used counter rotation between the rear wheel sets and the individual wheels in the sets. Thus, if properly geared, each wheel set would “step” motionless on each stair step without rotating relative to the stairs. This requires the proper ratio of wheel set speed and rotational speed for the tires.

The early work by King et al was followed by several unique approaches to rotating wheel sets for stair climbing robots. Andrew Poulter set forth the concept of a robotic all-terrain device that consisted of two elliptical halves or “clam shells” that supported the drive mechanism for two wheels (Poulter, 2006). These clam shells were articulated as connected together with a common shaft. In this way, the robot could almost continuously have all four wheels in touch with the surface. Although not intended for stair climbing this device demonstrated articulated wheel sets.

Poulter also used a long boom situated between the two clam shells that could be rotated to right the vehicle should it topple over or need to raise the forward or rear wheel sets. This device also incorporated the concept of having no front or rear, handling either direction desired as forward.

In 1998 Yasuhiko Eguchi from Heyagawa, Japan was issued a patent on a system of eight wheels driven in sets of two wheels each (Eguchi, 1998). This system could both rotate wheel sets and drive the wheels individually and separately. This vehicle had its individual wheel drive and wheel set drive linked using gears. As the wheel sets were driven, the gears would transfer torque to the individual wheels. Having the power transferred in this way caused the wheel sets to rotate opposite to the wheels, much the same as King et al.

The LSUS robot design paralleled the Eguchi concept set forth in his 1998 patent. As far as the authors are concerned, the LSUS design is the first prototype of its kind that applies the Eguchi concept and combines stair climbing with rough terrain negotiation capabilities. The LSUS adaptation of this wheel set concept for robotics limited the rotation of the wheel sets to approximately 35 degrees in either direction from level using pneumatic cylinders affixed to each of the wheel sets. The type of pneumatic control valves allowed a step up or down of the wheel sets and also a "neutral" position where the air valves allow full and free motion as will be discussed later in this chapter. Also, the use of chain drive rather than gears was incorporated in the LSUS robot. This less expensive alternative to gears requires lower maintenance and is easily replaced should failure occur. Other works that apply the Eguchi concept for stair climbing is that of Minoru et al, 1995 although with Figure 1 shows WHEELMA (Wheeled Hybrid Electronically Engineered Linear Motion Apparatus), the LSUS designed robot that uses the Eguchi concept.

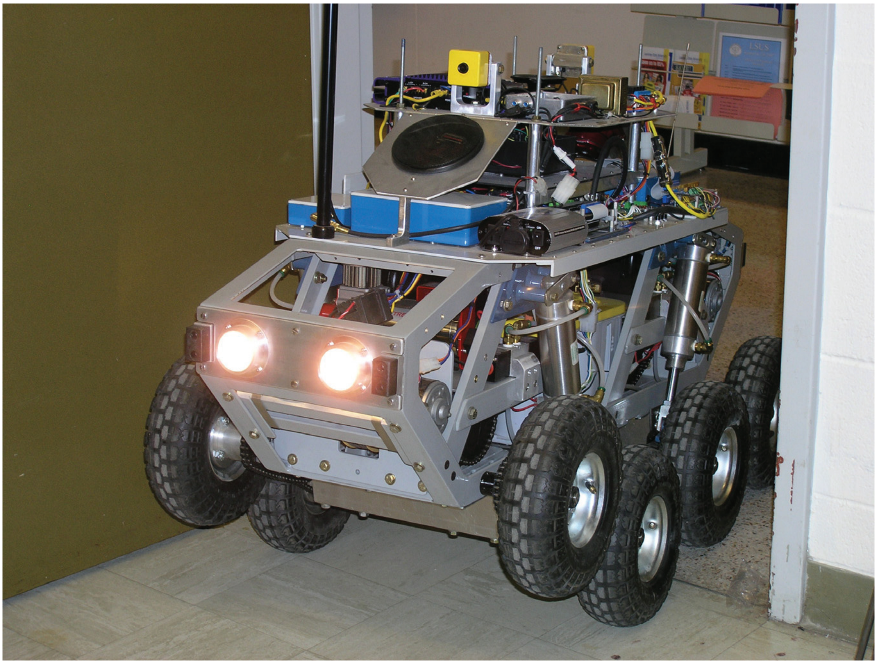


Fig. 1. Wheelma Robot based on Eguchi system

Extreme examples of wheeled robots use multiple wheel drive that is articulated not in a rotary manner but in a vertical manner. A vertically wheel articulated system is seen in a design by the Intelligent Robotics Research Centre in Clayton Victoria Australia (Jarvis, 1997). This robot, the size of a small car, uses six wheels that can move vertically to negotiate rough terrain. This robot was inspired by a Russian model of a Marsokhod Mars Rover M96. This robot was been located at the Intelligent Robotics Research Centre at Monash University since 1997.

Another unique example of articulated wheeled robots is the Octopus developed by the Swiss Federal Institute of Technology Zurich (Lauria et al, 2002). This wheeled design uses tactile sensing in each wheel to identify and negotiate obstacles. This robot's instrumentation can identify the height of obstacles and the system can decide how to handle the obstacle such as total avoidance or decide a strategy to overcome the obstacle. This eight-wheeled robot is small and can be configured to a variety of wheel configurations.

3. WHEELMA

A priority of the LSUS design was for it to be articulated so as to conform to un-level terrain as needed and continue to drive the robot in forward and reverse directions. Articulation requires a method of suspension with a certain amount of slack for the wheels to adjust to varying contours as they roll over terrain. Articulation combined with all-wheel drive has been used to handle rough terrain negotiation.

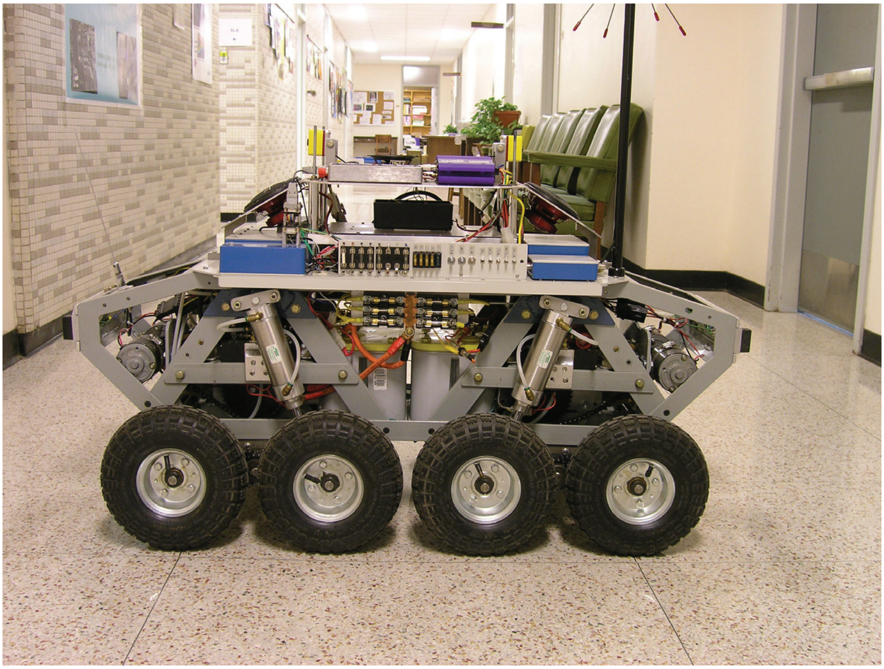


Fig. 2. Wheelma Resting on Eight Wheels

WHEELMA uses soft rubber inflatable wheels. All eight wheels of the robot use economical 10-inch tires and hubs that rotate around a unique hub design attached to a rigid frame. Figure 2 shows WHEELMA with both wheel sets level as would be the case typically on a flat floored surface. Four internal motors effectively drive the wheels sets in both forward and reverse directions through appropriate speed reduction. The wheel-set rotation around a central axel delivers the articulation needed to conform to terrain contour. Figure 3(a) and 3(b) show the side and top profile of the wheel sets used on WHEELMA.

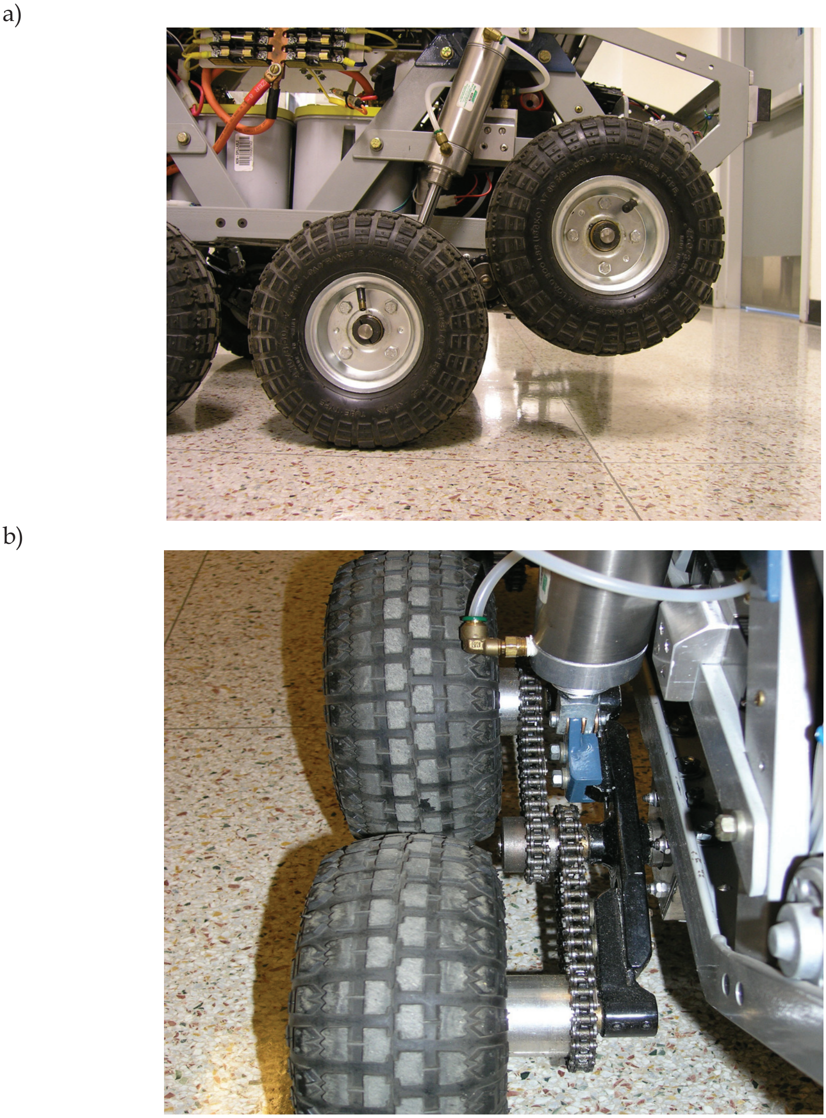


Fig. 3.(a) Side View of Wheel Set (b) Top View of Wheel Set

This design further allows fulfillment of the above stated design criterion with the use of a simple left-right differential control system, similar to the classic arcade game "Tank." Each control device or lever controls one side of the robot. The left control drives the two left wheel sets, while the right control drives the right side wheel sets. This allows forward and reverse motion and also rotation around an axis central to the robot in a fashion similar to a tracked vehicle.

Each wheel set has two wheel axels that are attached to a cross bar. Figure 3(b) shows the crossbar. The 1.25-inch axel that passes through two ball bearings (not shown) held in place by the robot's frame allows full articulation of the wheel sets. Inside each of these 1.25-inch axels is set of two needle bearings that support a second 0.5-inch axel used for driving the two wheels. Thus, an axel inside an axel enables rotation and horizontal translation for the wheel-sets. To allow for the tank-like operation of the robot, all wheel sets are allowed to "float" at times creating a contour following approach similar to a treaded vehicle. The wheel-sets also must encompass a control mode where they can both be locked or driven.

With each wheel set in either of these two modes, it must be driven through the central drive shaft for locomotion of the robot. The initial approach utilized in WHEELMA used electric clutches that allowed floating operation where the terrain requires such and locked rotation where needed for confronting obstacles. The clutches would be activated to thus be driven by a high torque source to rotate each wheel-set. This approach could deliver stair climbing capabilities.

Full rotational capabilities for each wheel set would demand a method of monitoring the position of each wheel-set. This task could be accomplished by means of an optical encoder. The optical encoder would digitally measure the angle of rotation for each wheel set shaft. As the shaft moves, digital pulses would be counted and at any given time the counter could be read to determine the position of the encoder and thus the position of the wheel-sets for this application.

This method of sensing would require an index mark where the electronics used for counting rotation pulses could find the level position and zero the counters that measured angular displacement. Otherwise, the encoder and digital circuit would not know when the sets were level. The clutch based system posed one daunting problem, in case of a power failure the clutches would disengage. If this occurred while the robot was climbing stairs the clutches would in turn release the wheel-sets causing a catastrophic loss of control. An alternate approach to the electric clutches was devised.

The new approach would limit the wheel set movement to approximately 35 degrees clockwise and counterclockwise. Although agility would be compromised, this approach would allow upward rotation for oncoming obstacles such as curbs and steps as well as rotation in the opposite direction for possible clearing of the robot from stuck positions. While continuous rotation may have been more beneficial to certain climbing operations the simpler approach was found to be more adequate for most operations.

A method to lower and raise each set of wheels was the next step in the design. The chosen approach replaces the clutches and drive mechanisms with four 2-inch diameter bore pneumatic cylinders each with its own air control valve operated electrically. This system solves two issues. It provides the wheel sets rotation and also afforded a method of achieving the "float" condition, which allows the wheel-set's unhampered movement as the robot negotiates terrain.

The float condition is achieved with the use of pneumatic control valves that have an off or neutral position which allows for the exhausting of both ports of the double acting cylinders to the outside air. Thus, when the wheel-sets rotate in one direction outside air is drawn into one double acting cylinder on one end and exhausted from the other. The use of exhaust filters addressed the resulting drawback of dirty air entering the system that could cause a buildup of debris in the air valves and cylinders.

Electronic control for each wheel-set is initiated by the ground based control unit and once this control information reaches the robot, power Darlington transistors control the current to actuate the pneumatic control valves. This separate control over each wheel-set was found to have great utility in ways not initially conceived.

This independent wheel-set control further facilitates the turning of the robot on surfaces that afforded a high coefficient of friction between the robot wheels and the surface. This was first noted on concrete where the coefficient of friction can be as high as 1. Turning with all eight wheels in the floating position requires a great deal of wheel torque due to the extended front and back wheels having to slide laterally to some extent due to the general nature of all differentially operated locomotion systems.

Since the vehicle can lift the outer ends of its tracks, or outer wheels in the case of WHEELMA, the device can turn on the inner portions of the tracks or the four inner wheels. This fact proved invaluable for turning with lower levels of power especially when the robot had been working and the batteries were supplying lower voltage and the electronics and motors were reaching higher temperatures all of which reduce the drive torque.

Raising the outer wheels places the same weight loading totally on the inner wheels maintaining approximately the same amount of friction between the surface and drive system. Much of the turning friction is reduced by the shortened moment arms of the inner wheels which allow for much lower resistance when turning since the resulting wheel drag will be decreased. Effectively shortening the length of the robot allows for far less lateral or non-rotational component yielding a greater turning ability.

The control portion of WHEELMA consists of a custom built base station which provides switches for defining options on the robot along with the standard two levers for the differential driving of the device in forward and reverse directions much the same as a tank. Figure 5 shows the base station along with associated switches, receiver controls, and TV for monitoring returning video from the robot.

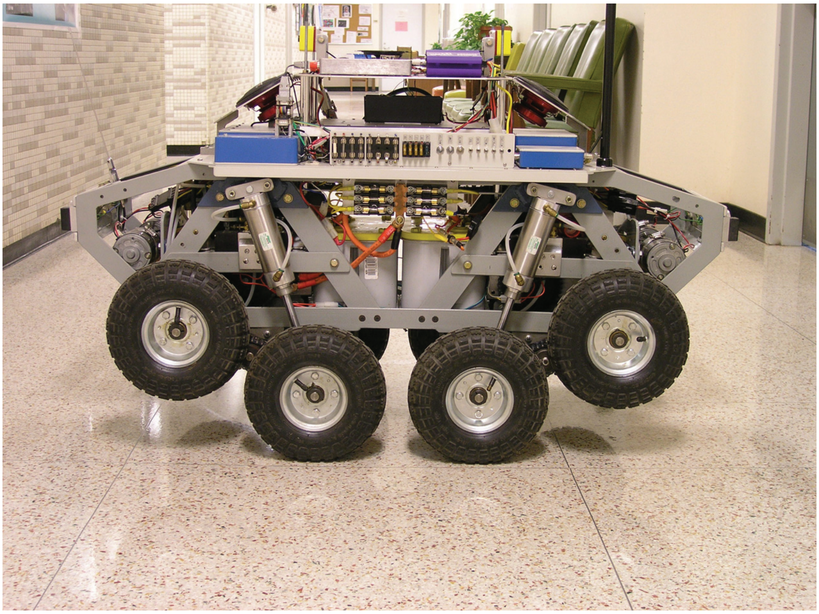


Fig. 4. Wheelma Resting on Inner Wheels

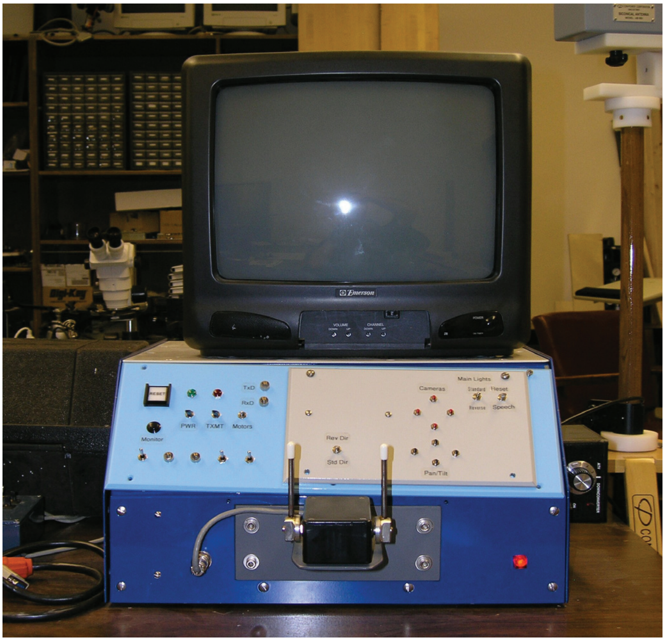


Fig. 5. Controls Station

4. Vision System and Directional Orientation

Many robotic designs rely on either physically turning the robot around 180 degrees to exit tight situations or else attempt to back out of restrictive spaces the same way they were entered. Many robots have the ability to reverse their viewing direction either by rotation of the camera or by using a secondary camera mounted to look aft.

It is also difficult to convey to an observer the lack of depth perception that occurs while using only one robotic camera at a time. Although two cameras could afford a stereo view of the operating area to the operator wearing a head-mounted viewing device, the difficulty in transmitting two simultaneous video signals precludes this practice for more applications. Such transmissions are often plagued by the need for excess transmission bandwidth for transmission or inter-modulation between two closely spaced video transmitters operating at nearly the same carrier frequency.

Using one camera for forward and one for reverse makes driving the robot easier in close quarters as the operator can always be looking in the direction of travel. This approach can, in some cases, eliminate having to perform a 180 degree turn by simply switching cameras and backing out of a given situation. Reversing the view however does not in itself reverse the control action necessary to drive the robot. This fact makes the operation extremely disorienting to the operator. Also the drive characteristics of the robot may change for certain robots used in the reverse mode of travel.

Because WHEELMA is basically symmetrical with no difference in handling characteristics in forward or reverse, the ability to reverse the robot’s direction of control was incorporated. The video circuit has provisions for up to eight cameras multiplexed through a specially designed video multiplexer controlled by the base station. This multiplexer feeds a single video output from one camera at any given time to the video transmitter on the robot. This allows for easy installation of two cameras, one at each end of the robot. Figure 6 shows the video multiplexer.

This multiplexer board uses two DG540DN video multiplexers which are under the control of the robot logic. This video multiplexer has the feature of using two analog CMOS switches in series to pass the video signal. When the channel is not selected the switches both go to the open mode and a third switch grounds the connection between the two. This lowers crosstalk to a minimum. Although there are eight possible connections to this multiplexer, only two are used for WHEELMA’s cameras. Refer to Figure 1 and 2 for a view of the cameras used. These cameras in the figures are in small square enclosures. Each looks toward its respective end of the robot.

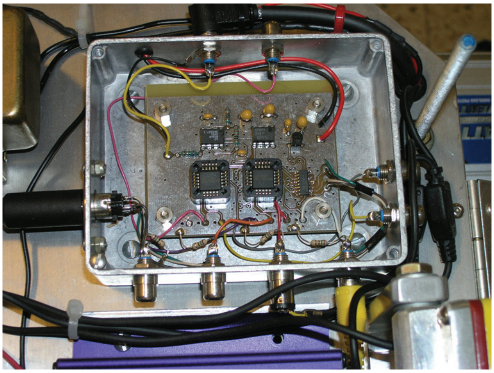


Fig. 6. Video Multiplexer

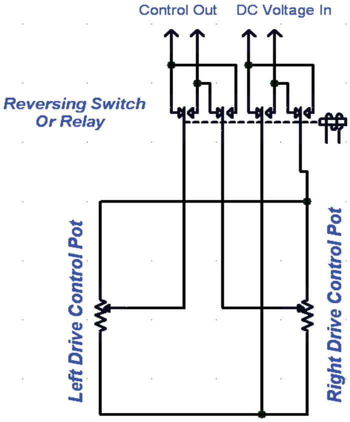


Fig. 7. DPDT Relay

Using a reversing strategy allows the operator to literally switch ends with the robot with the change of a single toggle switch at the base station. This toggle will switch the cameras, the left and right controllers, and even the lighting on the robot. In this way if WHEELMA enters a room and the remote operator desires to exit the room, one flip of a switch causes the view to change along with all necessary controls and lighting to simply drive out. Thus, WHEELMA has no front-back orientation and relies solely on the remote operator's manipulation of the direction switch.

Figure 7 shows the simple strategy for accomplishing a total reversal. A DPDT relay is shown that is driven by an SPST reversing switch. Lighting control is not shown, but can be added with using the microprocessor's logic located in the base station. Note that power is DC voltage and is supplied to two potentiometers connected to the left and right drive control levers (tank controls). When the relay is activated the polarity of the signal going to the two pots is reversed thus changing the directional nature of the controls Secondarily the left-right input from the pots are also switched to compensate for the left-right reversal effect. One switch can thus control a total reversal of control operation.

It should not be construed that the robot cannot reverse and back out of a tight location maintaining the same front-back orientation that carried the device into the room. The robot can also switch cameras and back out without reversal. However, using this system is seamless in operation and solves many problems encountered with some robots.

When setting up such a reversing control system, once should endeavor to place the zero thrust position in the exact vertical center position of the levers. Thus, when the left and right control levers are centered there is no thrust to the robot wheels. This is extremely important when doing a reversal as any residual thrust at the time of reversal will become an opposite thrust upon reversal. Several times WHEELMA operators have become disoriented through a combination of slight forward or reverse thrust set by the levers at the time of reversal.

It would be good design practice to either not allow reversal with thrust applied or to have a centering method to be assured that no residual thrust is present in the control lever settings when a reversal is undertaken. Allowing a spring loaded return to center for the thrust levers along with an indicator for denoting zero thrust could be used to prevent this effect.

5. Control Link

A simple system using the 1.25-meter amateur radio band was used to transmit all of the control signals from the base station to WHEELMA. This band of frequencies available to all amateur radio operators for sending ASCII packets was used in this and other previous robotic designs. (CGS Network)

The motor control information begins at the two-lever input for the tank-like control of the robot. These two levers are attached to two high quality potentiometers that divide the applied 5-volt potential. With each lever in the center position (vertical) the voltage is measured as 2.5 volts. As the levers are moved forward and backward the voltages rise and fall above and below this 2.5-volt center value.

The voltages are conditioned via an operational amplifier interface before being presented to a Motorola MC68HC711 microcontroller where the 8-bit A to D converters sample the voltage levels. Each voltage level is then converted into a 2's compliment signed number that is transmitted to the robot's receiver. A continuous stream of these sampled lever voltages are sent using ASCII transmission techniques. Each complete motor control sample is sent as one line of ASCII text followed with Return and Line Feed characters.

Each text string transmitted contains a starting character. A colon (:) was chosen to begin each string. Following the colon two ASCII bytes are transmitted representing the 2's compliment signed motor control sample position of the left lever for the left motor sets. Then the right motor set sample is sent using another two ASCII characters. At this point in the stream a simple checksum is transmitted followed by a Return and Line Feed. These two control characters in ASCII have provided a good way of showing all control information using a simple computer terminal for debugging purposes.

This stream of serial information at 2400 baud is sent to a modulator where it is converted to a standard FSK audio signal. This signal is then transferred to an amateur FM modulated transmitter operating on the 1.25-meter band. This 1-watt unit is the legal limit for amateur band transmissions for radio control purposes as authorized by the FCC. An Omni-directional whip antenna radiates the signal and has a range of over one mile line of sight.

Reception is done with a 1.25-meter crystal controlled receiver feeding a demodulator. The resulting 2400 baud serial stream is then sent to a second MC68HC711 microcontroller through the Serial Communications Interface (SCI) where the values are parsed and the checksum calculated and compared against the embedded checksum. If the checksum and framing characters match what is expected, the string is passed to a motor controller circuit via two parallel cables, one for each channel (Left and right).

A duty-cycle type motor control is utilized. This is implemented in WHEELMA using a specially designed duty-cycle board that takes parallel information in from the microcontrollers and converts it to controlled rectangular waveforms that drive the motor controllers. The motors are controlled using power MOSFET circuits, one for each of the four motors that drive the four wheel sets.

5. Voice System

To complete the work started by our computer science group that initiated the design of WHEELMA, it was decided to give the robot a voice. For this capability, the RC Systems DoubleTalk© RC8650 integrated circuit chip set was used. This set comes complete on a small module called the V-Stamp and is interfaceable to a standard microcontroller [8]. For this we chose the same Motorola MC68HC711 microcontroller used for other functions and interfaced it with the V-Stamp using the SCI port on the microcontroller.

Figure 8 shows the voice system consisting of two circuit boards (PCBs). The top board shows a standard microcontroller board connected to a board housing the V-Stamp. The V-Stamp board has a Maxim RS-232 driver IC for connection of the device to both the outside programming computer and also the microcontroller.

The V-Stamp uses a 3.3 volt power supply and has a 0 to 3.3V serial connection that is relatively easy to connect to any standard microcontroller. The V-Stamp operates in several modes. First, you can operate this device in the text-to-speech mode where the microcontroller can simply send text to the device and it will speak it with a number of different voices that are software selectable. Secondarily, you can record up to 33 minutes of sound into the RC86L60F4I version of the V-Stamp and play it back on defined boundaries. The user can use Indexes or Tags which allow the user to utilize either ASCII text labels or serial numbers for each word or phrase recorded.

In designing the voice system provisions were made to connect an RS-232 cable directly to the V-Stamp for entering both, the recorded words and messages as well as the tag information. Software is provided by RC Systems to upload converted .wav type files for the unit. The process is straightforward and can be mastered in short order.

Provisions were made to send tags for the V-Stamp via the same serial connection from the base unit. This technique worked well, but requires a separate computer to handle the voice tags. It was found that voice is extremely difficult to coordinate with robot motions for one single operator. This additional unit for generation of voice proved valuable during demonstrations of WHEELMA at locations such as schools and science museums.

6. Future Work

One preferred embodiment of a WHEELMA-type robot would be to return to the 360 degree wheel set rotation concept of Eguchi but replace the two-wheel sets with three-wheel sets. The three wheel sets would rotate into position easier in the process of climbing stairs and should have extremely good functionality in the negotiation of rough terrain. This should prove highly effective as the wheels sets can rotate for propulsion as well as being driven in the conventional manner. Similar devices have been developed for carrying heavy loads up stairs using a three-wheel drive mechanism.

7. References

Sam D. Herbert, Andrew Drenner, and Nikolaos Papanikolopoulos "Loper: A Quadruped-Hybrid Stair Climbing Robot" 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23, 2008

Masayoshi Wada "Studies on 4WD Mobile Robots Climbing Up a Step" Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics December 17 - 20, 2006, Kunming, China

Joseph Joy. "Automotive Vehicle" Patent 2393324 Application September 18,1982, Serial No. 458,886 I1 Claims. (a.18 0-17)

Raphael Porcello "Wheeled Device" Patent 1887427 April 6, 1932

Edward G. King, Baltimore; H. Shackelord, Jr., Finksburg; Leo M. Kahl, Baltimore, all of Md. Patent 4993912 February 19,1991.

Andrew R. Poulter 80128 United States "Rugged Terrain Robot"Patent (10) Patent NO.: US 7,011,171-BI Poulter (45) Mar. 14,2006

Yasuhiko Eguchi "Stairway Ascending/Descending Vehicle Having an Arm Member with a Torque Transmitting Configuration" Patent 5833248 Nov 10,1998.

Ray A. Jarvis, "Autonomous Navigation of a Martian Rover in Very Rough Terrain" Proc. International Symposum on Experimental Robotics, March 26-28 1999, Sydney University, pp.225-234

Lauria, Piguet and Siegwart, R "Octopus – An Autonomous Wheeled Climbing Robot" Proceedings of the Fifth International Conference on Climbing and Walking Robots, 2002.

CSG Network (2008). http://www.csgnetwork.com/hamfreqtable.html :accessed Aug 2008

4

Cable-Climbing Robots for Power Transmission Lines Inspection

Mostafa Nayyerloo, XiaoQi Chen, Wenhui Wang, and J Geoffrey Chase
Mechanical Engineering Department, University of Canterbury
Private Bag 4800, Christchurch 8140, New Zealand

1. Introduction

Power transmission line inspection is of utmost importance for power companies towards having sustainable electricity supply to vast number of customers in major industries as well as households in a city. Inspection provides valuable data from status of the line, thus helps line engineers to plan for necessary repair or replacement works before any major damages which may result in outage.

Constant energy supply to the customers requires performing all the inspection tasks without de-energizing the line, so live line inspection methods are of the most interest to power companies. These companies perform patrol inspection mainly using helicopters equipped with infrared and corona cameras to detect observable physical damages as well as some internal deterioration to the line and line equipment. However, aerial inspection is costly and always there is a risk of contact with live lines and loss of life. Moreover, there are some critical specifications of the line such as internal corrosion of steel reinforced aluminium conductors that should be inspected precisely from close distances to the line that are not accessible by a mobile platform such as a helicopter or even an unmanned aerial vehicle (UAV). Hence, power companies have endeavored to make especial cable-climbing robots to accomplish inspection tasks from close distances to the hot line.

Thanks to technological advances, utilizing robots as reliable substitutes for human beings in hazardous environments such as live lines has become possible. For many tasks requiring high precision over a long period of time, robots even do their job better than human operators. However, power companies have mainly focused on automating inspection tasks more willingly than making autonomous systems to perform repair works on the live line due to the fact that repair works are often complex to be accomplished by a robot.

In the past two decades, researchers have endeavored to make fully autonomous and intelligent cable-climbing robots equipped with necessary sensors for hot line inspection, aiming at making a cable-climbing mechanism with obstacle avoidance capability to pass the line equipment and the tower. Also some research has been done to devise a durable power supply method for the hot line inspection robots to make them sufficiently durable to perform inspection over long distances of live lines without interruptions for recharging the power source. Inspection data quality enhancement has been another challenging issue in this field due to the fact that swinging of the inspection robot in windy climates and even sometimes during the navigation makes the captured images of the line, as the main inspection data for line status evaluation, blurry. These undesirable vibrations also make some problems in the robot's navigation, which mainly relies on a vision system, in most of the proposed designs.

The robot's mechanical mechanism as main part of the robot design may significantly affect other issues in the whole design process such as energy consumption and inspection data quality. Hence, this chapter aims to review some of the main efforts made over the past 20 years in cable-climbing mechanism design for power lines inspection to provide a basis for future designs and developments in this field.

The chapter has organized as follows. Section two of the chapter briefly reviews different kinds of faults, which may occur in power lines, and origins of these faults. In the third section, which is the main part of this chapter, the focus will be on reviewing different types of mechanisms proposed for navigation and obstacle avoidance on power lines, advantages and disadvantages of each proposed mechanism over others, and adaptability of these mechanisms to power line environment. We then conclude the chapter in the last section.

2. Problems of deterioration in transmission lines and their symptoms

Transmission lines are exposed to variety of factors, such as corrosion and wind induced vibrations, which cause different problems and limit life time of the lines. Damage to the transmission lines can be categorized into two main groups: damage to the insulators and damage to the conductors.

2.1 Damage to the insulators

The insulators are affected by impact, weathering, cyclic mechanical and thermal loading, electro-thermal causes, flexure and torsion, ionic motion, cement growth, and corrosion (Aggarwal et al., 2000). Temperature difference between hot sunny days and freezing cold nights as well as the heat generated by fault current arcs cause thermal cycling, which produce micro-cracks and allows water to penetrate into material. The amount of imposed stress depends on relative expansibility of dielectric, metal fittings, and the cement used to fix the metal fittings of the line to the dielectric.

Cement growth, which is mainly caused by delayed hydration of periclase (MgO) as well as sulphate related expansion, generates radial cracks in the porcelain insulators' shell and makes them faulty (Aggarwal et al., 2000). Contaminants in the atmosphere, such as sea or road salts, can attack both Portland cement itself, or if penetrate into metal parts, can corrode galvanizing surface. Ionic motion caused by electric field makes this situation worse.

2.2 Damage to the conductors

The steel reinforced aluminium conductors (ACSR) are one of the most popular conductor types. The most important phenomenon that degrades such conductors is corrosion of aluminium strands. Pollutants and moisture, in the form of aqueous solutions containing chloride ions, ingress into the interface between the steel and the aluminium strands and attack galvanizing protection of the steel. Corrosion of the galvanizing coat exposes steel and aluminium to each other and leads to galvanic corrosion between iron and aluminium.

As an anode, aluminium corrodes rapidly and white powder aluminium hydroxide is produced. Loss of aluminium strands decreases current carrying capacity and mechanical strength of the line (Cormon Ltd, 1998; Aggarwal et al., 2000).

In addition to corrosion, wind induced vibrations can make severe mechanical damage to the conductors due to generating cyclic mechanical load. The wind flow creates vortices downstream when it passes the line. These vortices produce fluctuating lift and drag forces causing aeolian vibrations with frequencies from 10-30 Hz and amplitudes of the order of diameter of the conductor. In bundled conductors, the wind also induces sub-conductor oscillations, which can cause fretting of the aluminium strands near the clamps. The fretting reduces the fatigue strength of the line and speeds up the failure process.

2.3 Symptoms of the transmission line damage and detection methods

Damage to the line can be detected through investigation of their symptoms. Most of the line problems produce unusual partial discharges. Whenever the electric field intensity on the line surface exceeds the breakdown strength of air, electrons in the air around the conductor ionize the gas molecules and partial discharges, namely corona effects, occur. High frequency partial discharges produce radio noise in ultra-high frequency range, as well as audible noise in ultra-sonic range. In addition to noise, discharges send a current to the line. This current can also be used to detect faults. Depending on the weather, age of the line, problem conditions, and other factors, the level of discharge can also be different. Abnormal temperature is another symptom, which can be used to identify defects on the transmission lines.

Based on aforementioned major symptoms, following techniques are mainly used for detecting faults in the transmission lines (Aggarwal et al., 2000):

1. Ultrasonic detection
2. Measurement of corona pulse current inconsistency
3. Partial discharge detector
4. Infrared inspection of overhead transmission lines
5. Radio noise detection system
6. Solar-blind power line inspection system (through detecting UV)
7. Corona current monitor for high voltage power lines
8. Fiber optic application to transmission line inspection
9. Audible noise meters
10. Field testing of insulators

3. Cable-Climbing mechanisms for power line inspection

3.1 The design environment

Power lines are a dangerous environment with intensive electric and magnetic fields. Power lines are also a complex environment, and difficult for robots to navigate. The simplest power lines have one conductor per phase hung on insulator strings, which can be either suspension or strain insulators. Besides insulators, there are other obstacles on the conductors, such as dampers, aircraft warning lights, and clamps. In bundle power lines, which have more than one conductor per phase, there are even more obstacles such as spacers and spacer dampers (Figure 1)[1].

The robot travels suspended from the conductor and has to cross obstacles along the power line that requires complex robotic mechanisms including conductor grasping systems and robot driving mechanisms. Moreover, an obstacle detection and recognition system, robot control system, communication, inspection platform equipped with necessary sensors and measurement devices, power supply and electromagnetic shielding have to be considered in robot mechanism design and construction.

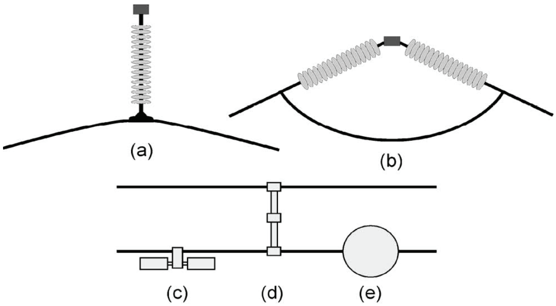


Fig. 1. Different obstacles on power line conductors: (a) suspension insulator, (b) strain insulator, (c) damper, (d) spacer/spacer damper (e) aircraft warning sphere (Katrasnik et al., 2008).

3.2 Cable-climbing mechanisms designed over the past 20 years

One of the first cable-climbing mechanisms presented in (Aoshima et al., 1989). The proposed mechanism, which was designed for telephone lines inspection, compared to its previous works is able to transfer to branch wires, and thus provides more flexibility in cable-climbing. The robot structure, as shown in Figure 2, consists of a multi-unit of six identical modules with three degrees of freedom: longitudinal movement on the cable, horizontal rotation about robot's longitudinal axis parallel to the power line, and vertical elongation of robot's arms.

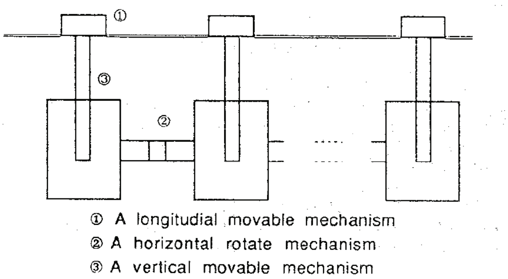

①
③
②
① A longitudial movable mechanism
② A horizontal rotate mechanism
③ A vertical movable mechanism

Fig. 2. Proposed cable-climbing mechanism in (Aoshima et. al., 1989).

[1] All of the figures and tables used in this literature survey have been taken from the original works presented by their authors.

Using different configuration of these six units, the robot will be able to adapt itself to different geometrical environments and avoid obstacles with different shapes and sizes, transfer to a branch wire, and even transfer to a parallel line. As an example to show the flexibility of the proposed mechanism by Aoshima et al., Figure 3 describes how the robot transfers to a branch wire. The proposed robot has good maneuverability over different obstructions and variety of different geometrical environments on the power lines, but as Figure 3 shows, the robot is complex in control.

One of the first efforts towards designing a more simple cable-climbing mechanism carried out by (Sawada et al., 1991) to inspect fibre-optic overhead ground wires (OPGWs). The proposed robot, as shown in Figure 4, consists of a vehicle assembly to navigate on the power line, an arc shape guide rail, a guide rail manipulator, and a balancer with controller to pass the obstacles. It can travel on slopes of up to 30º. When the robot comes across an obstacle, it opens its rail and hangs it on the line on both sides of the obstacle. Then the drive mechanism detaches from the conductor and travels on the rail to the other side of the obstacle. Utilizing such an obstacle avoidance mechanism, the robot is able to negotiate towers as well and transfer to the next span to inspect rest of the power line.

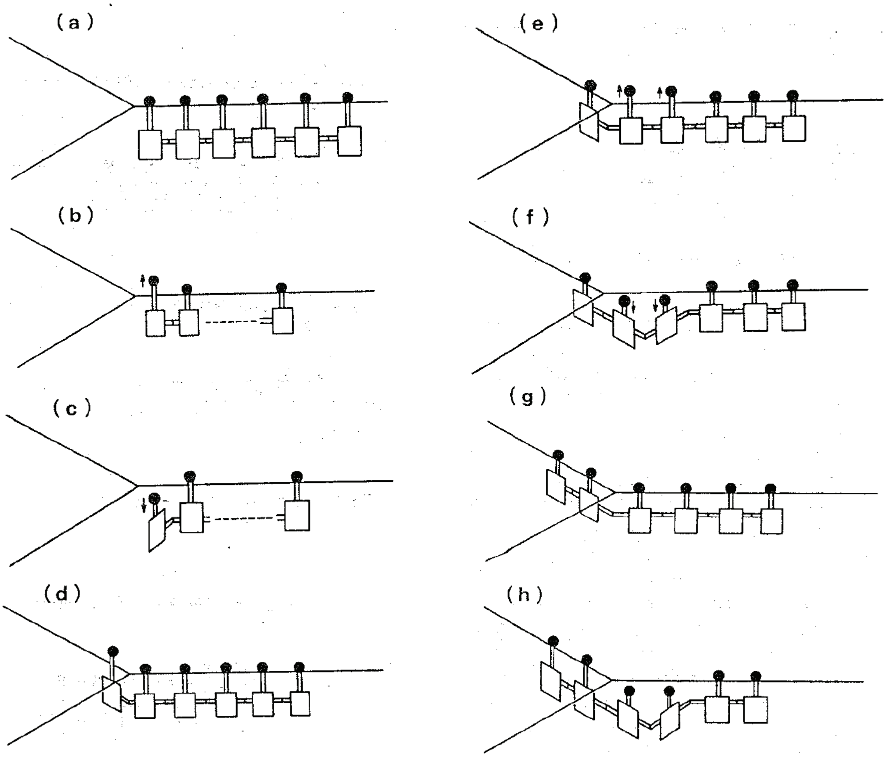


Fig. 3. How the multi-unit robot transfers to a branch wire (Aoshima et al., 1989)

The proposed robot did not have proper shielding for live line inspection and could not travel on phase conductors. Moreover, stability issues in windy climates and slow obstacle overcoming mechanism, for example, spending 15 minutes to overcome each tower, were unresolved issues in the proposed robot.

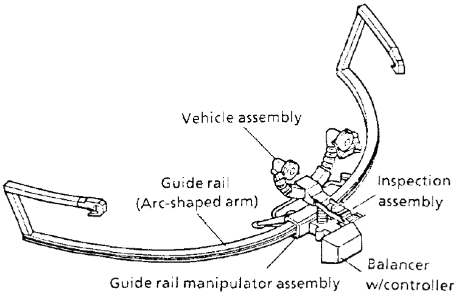


Fig. 4. Basic configuration of proposed mechanism in (Sawada et al., 1991)

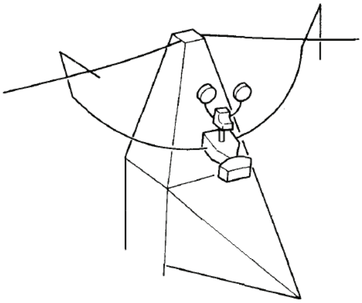


Fig. 5. How Sawada and his colleagues' robot passes the towers (Sawada et al., 1991).

In the same year, a project with same purpose was done by Higuchi et al. The proposed stride type robot, as shown in Figure 6, can move on a ground wire stretched on top of the towers and is also able to pass the towers (Higuchi et al., 1991). The robot can navigate on the overhead ground lines using two arms to take steps and a crawling mechanism to move on top of the towers (Figure 7).

The presented work has the stability problems in windy climates and is also more complex to control than the work in (Sawada et al. 1991). In addition, as Figs. 6 and 7 show, the design is specific for inspection of the overhead ground wires stretched on top of the towers with a flat area on top and cannot be used as a general inspection robot for all types of the power lines. For instance, the robot cannot pass the towers when it is traveling on phase conductors as the robot is not able to overcome the insulators.

To achieve both stability in movement and simplicity in control, another project ran by Tsujimura and Morimitsu in 1997 to make a cable-climbing robot for the telecommunication lines inspection. The proposed robot in (Tsujimura & Morimitsu, 1997) and the locomotion principle have been shown in Figs. 8 and 9. A linkage mechanism creates a gait kinematically and causes the arms to hang on the cable at intervals. The robot walks parallel to the cable and as it moves, due to the nature of the movements, can avoid obstacles as well.

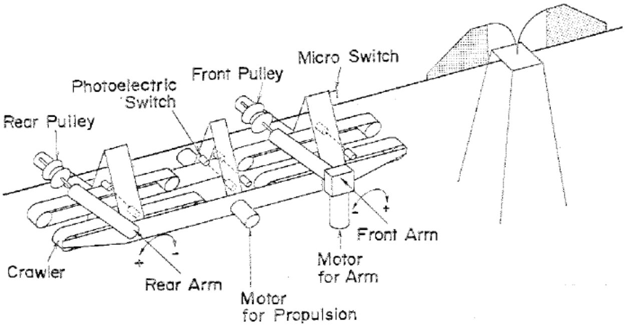


Fig. 6. Architecture of the robot proposed by (Higuchi et al., 1991)

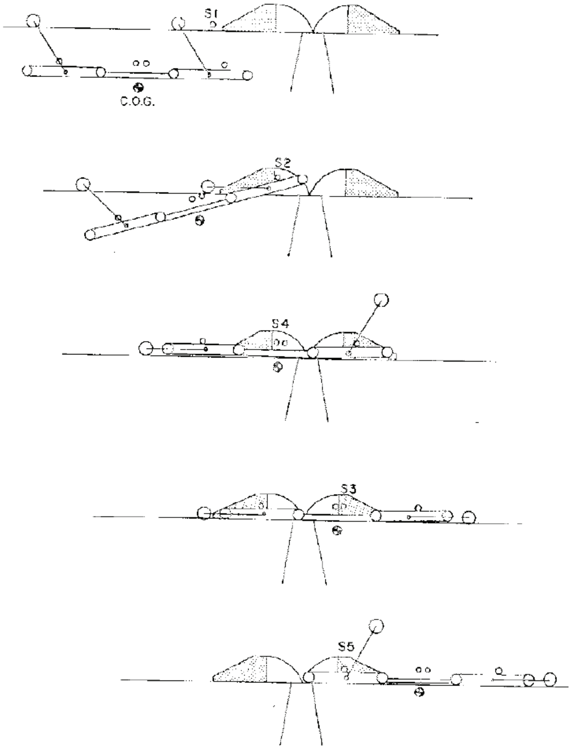


Fig. 7. Sequence of going over the tower in (Higuchi et al., 1991)

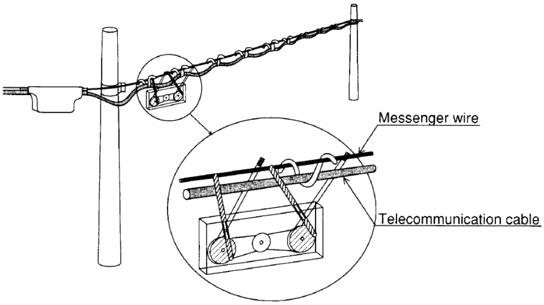
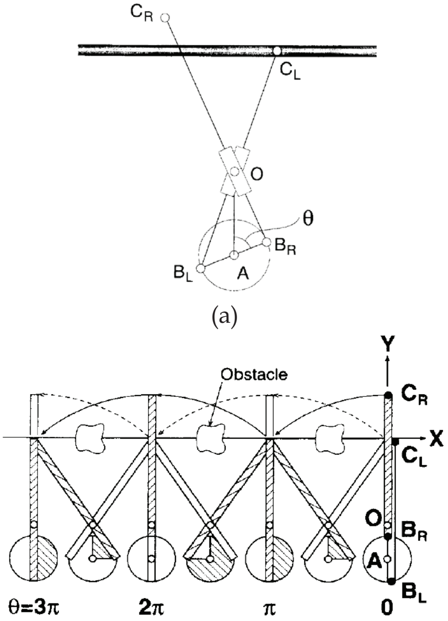


Fig. 8. Proposed cable-climbing mechanism in (Tsujimura & Morimitsu, 1997)

The proposed mechanism can provide constant moving speed, which is ideal for inspection, is simple to do, stable, and simple to control, but cannot transfer to the angled lines.



(a)

(b)

Fig. 9. Locomotion principle of the proposed mechanism in (Tsujimura & Morimitsu, 1997) (a) linkage that provides gait movement and (b) simulation of movements of one of the robot's arms

Meanwhile, some researchers focused on making fully operational robots to carry out special tasks on the power lines, even though they were not able to perform the task fully autonomously. One such robot was presented by Campos et al. in 2002. The proposed robot, shown in Figure 10, is a simple but operational cable-climbing mechanism for installation and removal of the aircraft warning spheres. This robot can only navigate on part of the line between two towers without avoiding any obstacle. Similar mechanisms, shown in Figs. 11

and 12, can be found in (Sato Ltd., 1993) and (Cormon Ltd, 1998), respectively. The proposed mechanisms consist of a trolley with two pulleys on top that can move the trolley and all the required manipulators and the inspection tools. The mechanisms proposed by Campos et al., Sato Ltd., and Cormon Ltd. have been tested in real working conditions and the two latter are commercially available. Although these robots are not able to pass the obstructions on the power lines or transfer to the next span, they are simple and operational.

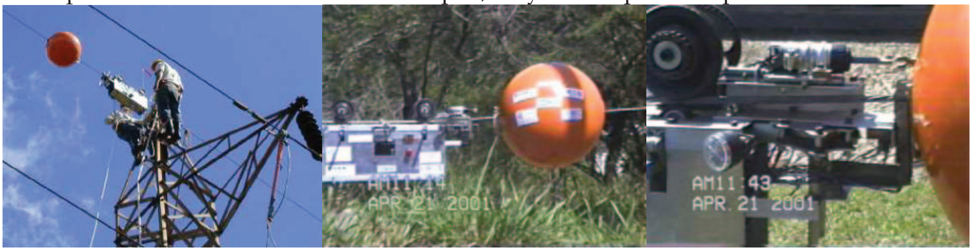


Fig. 10. The aircraft warning sphere installation and removal mechanism proposed in (Campos et al., 2002)

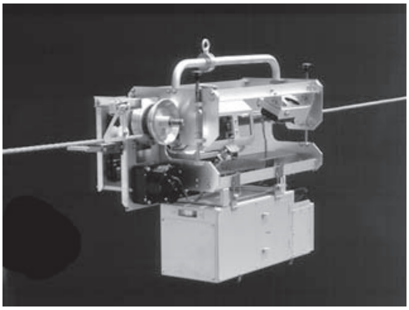


Fig. 11. Automatic overhead power transmission line damage detector developed by (Sato Ltd., 1993).

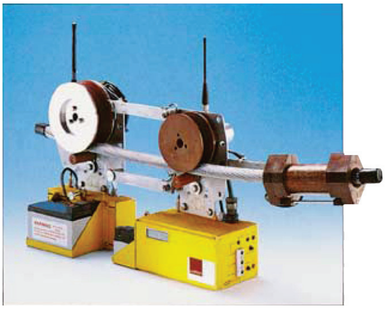


Fig. 12. Mobile corrosion detector proposed in (Cormon Ltd., 1998)

Some more throughput mechanisms were developed in 2004. One of these mechanisms has been shown in Figure 13. This figure shows a sketch of the mechanical mechanism designed by (Tang et al., 2004). The proposed robot has two front and rear arms and a body. There is a gripper on top of each arm and a running wheel on top of the body. In addition, using wheels in the gripper design have enabled this robot to move along the line back and forth even when the grippers grasp the line.

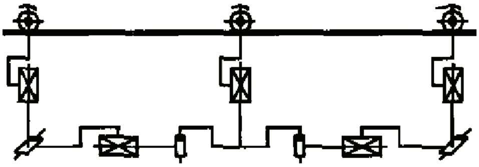


Fig. 13. Proposed mechanism by (Tang et al., 2004)

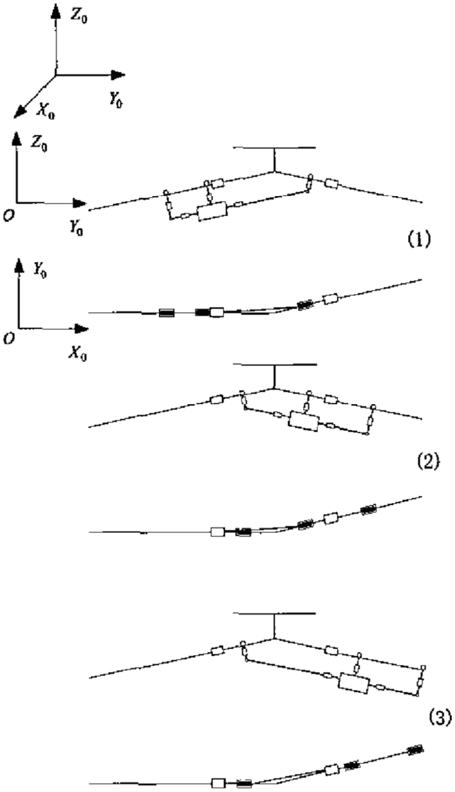


Fig. 14. Proposed obstacle navigation procedure in (Tang et al., 2004)

The obstacle navigation process is shown in Figure 14. When the robot detects an obstacle, the rear arm gripper grasps the line, and the front arm elongates to pass the obstruction. In the second step, the front arm gripper grasps the line, and the running wheels get off from the wire. Next, the two grippers continue to move forward, and consequently the body can move across the obstacle. Finally, the running wheel turns over the wire and grasps it, the rear arm gripper is detached from the line and pass the obstacle. Considering different distances between two consecutive obstacles and different tower sizes in mechanism design, the proposed robot can pass all types of the obstacles on straight lines autonomously.

Similar crawling mechanism also proposed by (Wolff et al., 2001), and modified by (Nayyerloo et al., 2007). In the latter, a mechanical mechanism, as shown in Figure 15, was proposed. The mechanism has three similar grippers mounted on top of three arms, which can go up and down. These three independent grippers make the robot able to be fixed to the line or move along the line easily when it is hung from the line. Two motors in the driving system mounted on top of the middle arm drive the whole robot. The front and rear arms can move along the robot length synchronously using the arm driving mechanism and two connection rods, which have connected these two arms together. The middle arm is fixed to the robot body.

The proposed arm driving mechanism plays two roles for the robot: translation of the front and rear arms along the robot length and translating the robot itself. The front and rear arms have been mounted on two nuts of a main screw, which represents the robot body. If the main screw is driven while the middle gripper has been clamped to the line, both movable arms will move along the line together. In a same way, there is another possibility to fix two movable arms to the line and drive the screw to move the robot body.

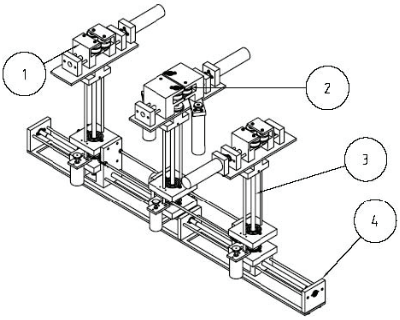
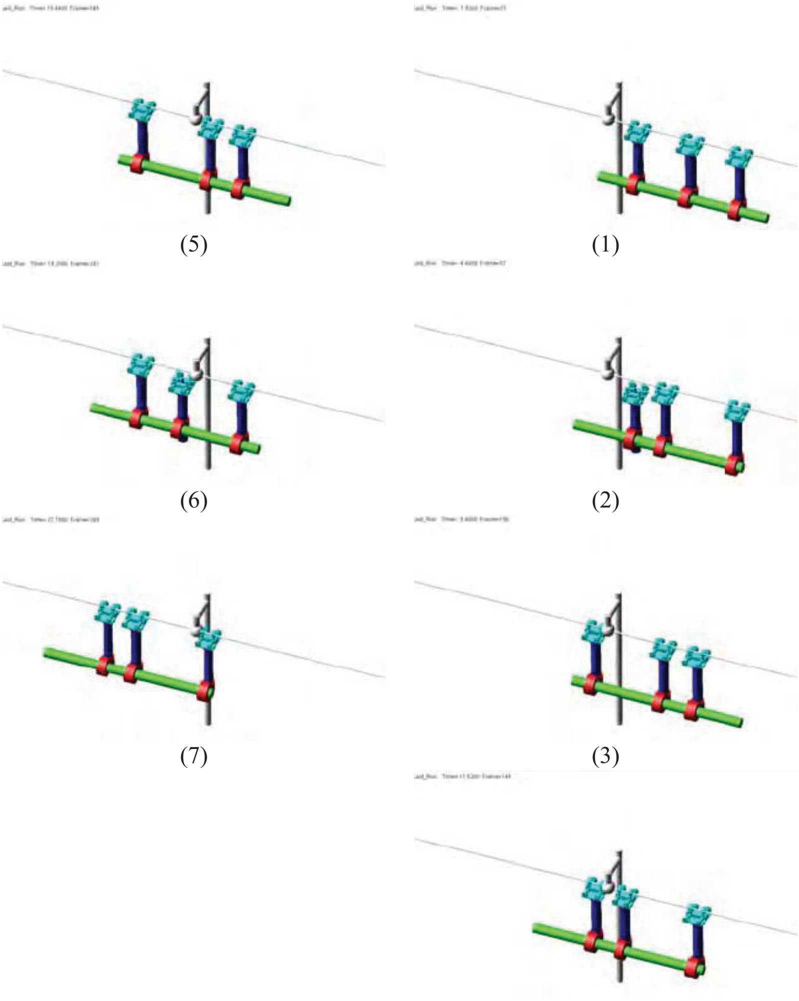


Fig. 15. Proposed mechanism by Nayyerloo et al., 1) the gripping mechanism, 2) the driving system, 3) one of the three arm mechanisms, and 4) the arm driving mechanism (Nayyerloo et al., 2007)

When the robot detects an obstruction, the gripper of the closest arm to the obstacle opens, and the arm goes down to avoid contact. The next step is translating the arms forward by fixing the middle arm to the line and driving the robot's main screw to pass the front arm to the other side of the obstacle. When the front arm passes the obstacle, it goes up and grips the line on the other side of the obstruction (1-3 in Figure 16). At this stage, the robot needs to make enough room on the other side of the obstacle to transfer the middle arm. This will be done by fixing the front and rear arms' grippers to the line and driving the main screw to move the middle arm as close as possible to the obstacle, and then fixing the middle arm to the line and moving the front and rear arms forward (4 and 5 in Figure 16). To transfer the middle arm to the other side of the obstacle the front and rear arms' grippers grasp the line, the middle gripper is detached from the line, the middle arm goes down, and the main screw is driven to move the middle arm under the obstruction. The middle arm grasps the line on the other side of the obstacle and makes the robot stable to pass the rear arm (6 and 7 in Fig 16). Following the same concept, the rear arm passes the obstacle, and the robot returns to its original configuration.


(5) (1)

(6) (2)

(7) (3)

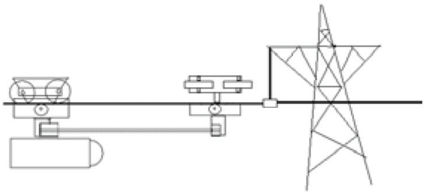
(4)

Fig. 16. Obstacle traversing mechanism proposed in (Wolff et al., 2001) and (Nayyerloo et al., 2007)

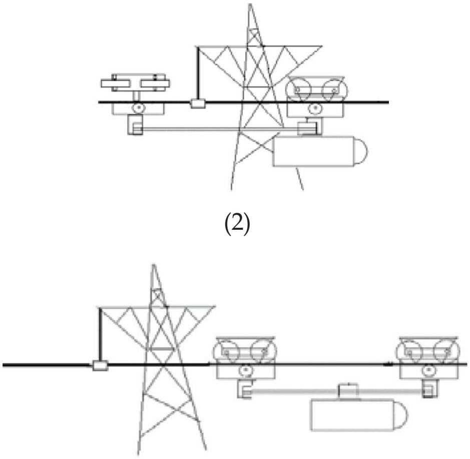
The proposed mechanism is simple to control, fast, and stable in overcoming obstacles. Even in windy climates, the three flexible arms can lift the robot body and make it as close as possible to the line to avoid swinging with large amplitudes. Another interesting feature of the proposed mechanism is its capability to navigate on paths with different shapes i.e. the robot's path could be even non-parabolic. The lengths of the robot arms can be adjusted according to the slope of the navigating path to keep the robot horizontal in all situations during the movement. The robot also has three hinged joints at the end of each arm that allow the grippers to be adjusted according to the slope of the line i.e. only grippers are sloped and the arms remain vertical in all situations. Besides the advantages of the proposed mechanism, it still needs to be modified to be able to pass tension towers with angled lines.

Another interesting obstacle traversing mechanism for power lines inspection robots proposed by (de Souza et al., 2004). The obstacle overcoming procedure is shown in Figure 17. The configuration in this figure has two sets of three wheels to move the robot along the line. When the robot detects an obstacle following obstacle avoidance procedure is followed: the box in middle of the robot, which contains all the required inspection tools, moves back along the track, the front set of wheels releases the line and rotates, the rear set of wheels is moved to surpass the obstacle (1 in Figure 17), and the front set of wheels grips the cable on the other side of the obstacle. Next, the box is moved forward to the other end of the track, the rear set of wheels releases the line and rotates then the robot moves until the rear set of wheels surpasses the obstacle (2 in Figure 17). The rear wheel set grasps the line again, and the robot goes back to its original configuration (3 in Figure 17). The same concept in obstacle overcoming mechanism, as shown in Figure 18, was also used by Sun et al. (Sun et al., 2006), but without centroid adjustment. In this work, the authors tried to optimize the mechanism design through using some simulation and analysis software's such as Pro/E and ANSYS. Thus, both designs are simple, stable, and fast in obstacle avoidance, but they apparently cannot pass the tension towers and transfer to angled lines. Moreover, such mechanisms should be modified to be able to overcome the obstructions such as warning spheres, which have more protrusion from the line than the clamps.

Another interesting cable-climbing mechanism, which is similar to the works in (de Souza et al., 2004) and (Sun et al., 2006) with a modification in arms movement, was presented by Fu et al. in 2006. Their inspection robot, shown in Figure 19, has two arms with driving wheels mounted on top of each arm. The arms can go up and down, and the driving wheels, which are combined with a gripper mechanism to grasp the line when it is required, can move the robot along the line. When the robot encounters an obstacle ahead, the main body is moved forward to balance the robot's weight according to the front arm position. Next, the rear arm raises its driving-gripping set from the line, passes the obstacle, and lowers down to grasp the line on the other side of the obstacle. To pass the front arm to the other side of the obstacle, the robot needs to balance its weight according to the rear arm position. Next, the front arm releases the line, goes up, and passes the obstacle. In this procedure, the rear arm becomes the front arm and vice versa. When the both arms pass the obstacle, the robot should go back to its original configuration again (Fu et al., 2006).



(1)



(2)

(3)

Fig. 17. Obstacle traversing mechanism proposed in (de Souza et al., 2004)

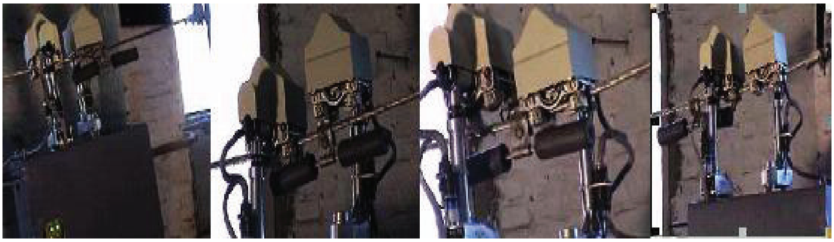


Fig. 18. From left to right, obstacle overcoming process in (Sun et al., 2006)

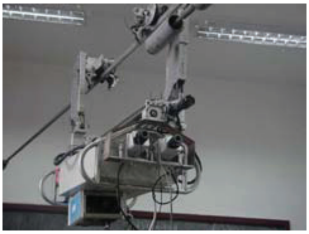
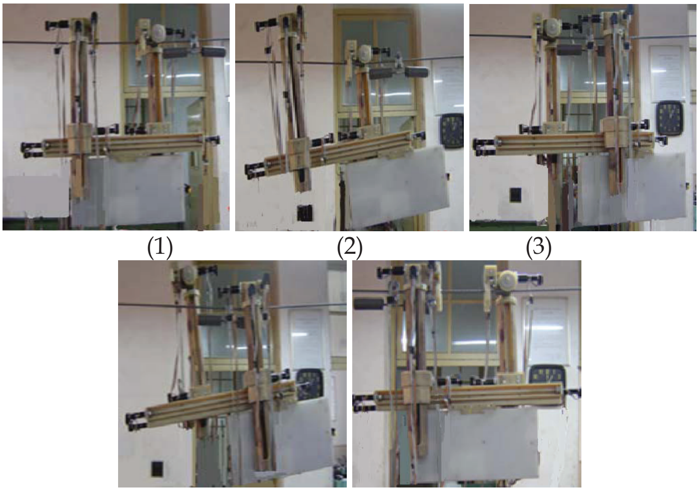


Fig. 19. The cable-climbing robot presented in (Fu et al., 2006)

Similar work to the project accomplished by Fu et al. has been recently carried out by Ren and Ruan. As shown in Figure 20, similar method has been used for navigation and obstacle traversing in (Ren & Ruan, 2008). These two mechanisms can only pass obstructions with small protrusion from the line such as different kinds of clamps and dampers, and more importantly, these two mechanisms cannot transfer to the angled lines.

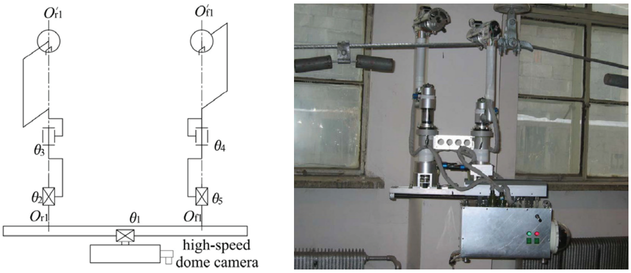


(1) (2) (3)

(4) (5)

Fig. 20. Obstacle traversing mechanism proposed in (Ren & Ruan, 2008).

One of the most simple and efficient mechanisms, which resolves some of the issues in (Fu et al., 2006) and (Ren & Ruan, 2008), proposed by Zhu et al. in 2006. The robot configuration, as shown in Figure 21 (a) and (b), has two arms equipped with a special gripper combined with a driving wheel. This special running-gripping mechanism has been shown in Figure 22. When the robot detects an obstacle on its way, it stops, grasps the conductor with the front gripper and moves its main body under the front arm in order to minimize the required torque needed for crossing the obstacle (Figure 23 (a)). Next, the rear arm lifts the rear running wheel up, and the front arm rotates the whole robot around its own axis. Finally, the rear arm lowers its wheel set on the conductor (Figure 23 (b)), and then the same process is repeated with the arms' roles changed. Such an obstacle traversing strategy makes the robot’s arms simple with only two degrees of freedom. Also the required torques in the joints and on the line will be small enough, thus there is no need for the heavy and powerful motors.



(a) (b)

Fig. 21. Robotic mechanism configuration proposed by Zhu et al.: (a) sketch of the robot's configuration, and (b) the robot prototype (Zhu et al., 2006)

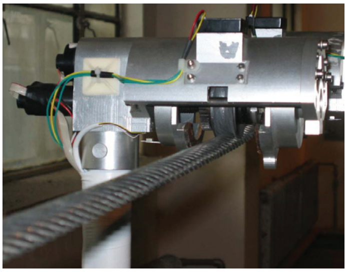


Fig. 22. Special running-gripping mechanism proposed in (Zhu et al., 2006)

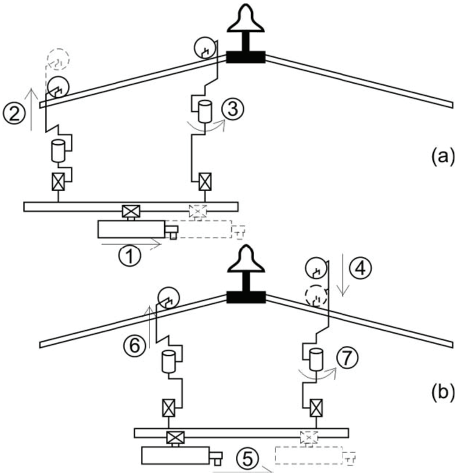


Fig. 23. Obstacle avoidance procedure proposed by Zhu et al. (Katrasnik et al., 2008)

The proposed mechanism is simple, stable and efficient. Moreover, with some minor modifications in the gripper design to increase the contact area between the gripper and the line, the robot will be even more stable in windy climates. Another interesting feature that the proposed mechanism can provide with minor changes in the existing gripper design is the ability to transfer to angled lines. The robot rotation angle about the arms' axes in the obstacle avoidance procedure can be adjusted to the angle of the line ahead, thus the robot can easily transfer to an angled line, but as in this situation, the grippers are not parallel to each other, they should be able to rotate about the arms' axes to be able to adjust themselves to the line angle.

The mechanism has a minor disadvantage that should be taken into account when dealing with acquired inspection or navigation data. When the robot passes an obstacle, it has to rotate 360° around the arms' axes. Hence, the installed inspection or navigation sensors on the robot body, such as image system, will rotate 360° under the obstacle to pass it. This rotation in the robot main body has been missed out in both the original work and (Katrasnik et al., 2008). It therefore has not been shown in Figure 23, which has been taken from the latter work. This rotation may cause some undesirable inspection or navigation data.

One of the most important efforts to develop an operational cable-climbing mechanism has been made by Montambault and Pouliot over the past five years. Their teleoperated robot, LineScout, shown in Figure 24, is the third generation of its previous prototypes designed by this team. LineScout uses driving wheels for locomotion. These wheels allow the robot to not only move quickly along the power line, but also to roll over some obstacles e.g. compression splices and vibration dampers. To clear other types of obstacles, the robot follows the approach schematized in Figure 25 (a). As shown in this figure, LineScout has three independent frames: the wheel frame (dark frame), which includes two motorized wheels called "traction wheels", the arm frame (light frame), with two arms and two grippers, and the center frame (white circle), which is called "extremity frame". This frame links the first two frames together and allows them to slide and pivot (Montambault & Pouliot, 2007). As an obstacle is detected, the arm frame is opened, and its two arms and grippers temporarily support the robot while the wheel frame is being transferred to the other side of the obstacle with the wheels flipped down under the obstacle. The wheel frame also includes a pair of safety rollers (small rectangles beside the driving wheels) for platform stability when crossing obstacles. The center frame supports the electronics cabinet and battery pack. The role of the center frame is generating the movements of the two other frames by sliding them in opposite directions and also supporting about 40% of the platform's weight to minimize the cantilevered load applied when the two other frames are in total extension during the obstacle clearance sequence as shown in top picture in Figure 25 (b).

Although the appropriateness of the proposed line inspection robot has been shown through several lab and field tests for straight lines between suspension towers, the mechanism still needs to be modified to be able to pass the tension towers with angled lines. However, it surpasses all its previous designs in terms of considering different real field requirements such as resistance to intensive electro-magnetic field and durability of the inspection task that are vital toward having an operational inspection robot.

In order to decrease the above mentioned complexities in designing a versatile cable-climbing mechanism capable of clearing all types of the obstructions and able to transfer to the angled lines on various power transmission lines, a new type of robot proposed by Katrasnik et al. in 2008 (Katrasnik et al., 2008). The idea is based on the combination of a flying robot such as (Williams, 2000) or (Golightly & Jones, 2005) with a cable-climbing robot to navigate on the power lines. The proposed flying-climbing robot, schematized in Figure 26, removes all the complexities in the mechanical mechanism design arising from the obstacle avoidance system by flying over the obstacles or flying from one span to another span to pass the towers. The implemented cable-climbing mechanism, which is the default navigation system of the robot on the line between the obstacles, ensures stability in navigation and provides performing the inspection from close distances to the line that is necessary for various line inspection sensors such as corrosion detectors. Using the climbing mechanism as the default navigation system also removes the flight control complexities in existing flying inspection robots when the robot navigating between the obstacles.

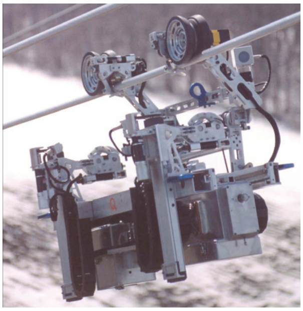
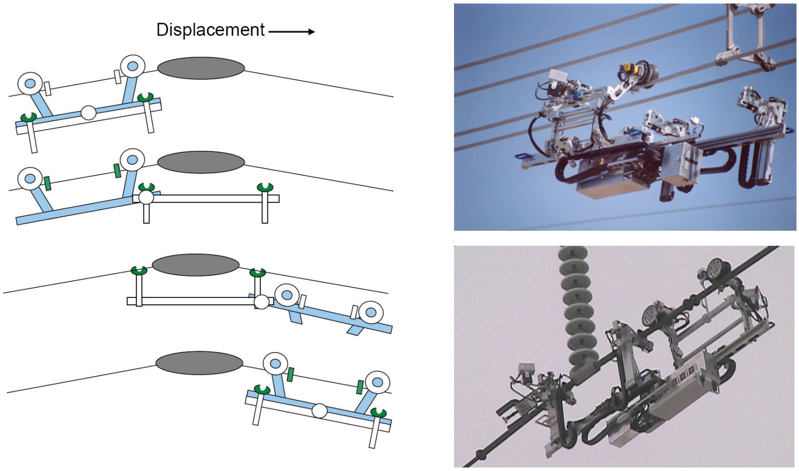


Fig. 24. LineScout mobile platform (Montambault & Pouliot, 2007).



(a) (b)

Fig. 25. LineScout obstacle-clearing sequence: (a) schematic of the obstacle-clearing procedure (Pouliot & Montambault, 2008) and (b) LineScout clearing obstacles in real working conditions (Montambault & Pouliot, 2007)

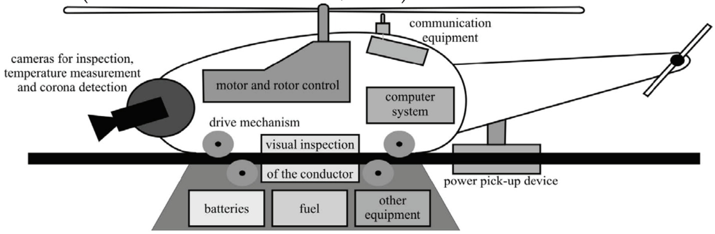


Fig. 26. Flying-Climbing robot (Katrasnik et al., 2008)

Katrasnik et al. have compared the two possibilities in the power lines inspection robot design, flying and climbing, with their novel mechanism proposed in (Katrasnik et al., 2008). The results are shown in Table 1. As the table shows, four major criteria have been chosen for this comparison: Availability and simplicity of the robot's design and construction, quality of the acquired inspection data, fully autonomous inspection, and adaptability to different situations on the power lines or universality. The comparison criteria have been sorted in order of importance and appropriate weight, as shown in Table 1, has been given to each criterion.

	w	Climbing	Climbing-Flying	Flying
Design and Construction	4	1|4	2|8	3|12
Inspection quality	3	2|6	3|9	1|3
Autonomy	2	3|6	2|4	1|2
Universality	1	1|1	2|2	3|3
Total score		17	23	20

w=weight, rank | weighted score

Table 1. Robot type comparison table (Katrasnik et al., 2008)

Various types of unmanned aerial vehicles (UAVs) are commercially available, and there is no need to start from scratch and build a new flying platform for inspection of the power lines. Hence, the highest point in the first criterion has been given to the flying robot. Also with some modifications on available UAVs in the market, the climbing-flying robot is achievable, thus the second highest scored design is the climbing-flying robot.

To compare the inspection data quality, the most problematic case is the flying robot (Katrasnik et al., 2008). Vibrations of the flying platform and its distance to the power lines during the inspection affect quality and resolution of the captured images, which are the main inspection data. The climbing mechanism has less vibration than the flying robot due to the line support and has been ranked higher than the flying robot. The climbing-flying robot can inspect the line equipment (obstacles) from additional angles when it flies over them. In that, the highest grade has been given to the climbing-flying robot.

In terms of autonomy, due to complexity of the flight control, making a fully autonomous flying robot is certainly more difficult than developing an autonomous climbing mechanism. Moreover, when the robot is attached to the power line, the required power can be supplied through the live-line that makes the robot more independent and autonomous.

The last evaluation item in Table 1 is the universality of these three mechanisms. The flying robot is completely unattached to the power lines, and thus has more maneuverability over different power lines. The climbing-flying robot needs some modifications in the climbing side to be adapted to different conductor sizes, but the climbing robot needs major modifications especially to be able to pass different obstacles. In that, the highest score has been given to the flying robot, followed by the flying-climbing mechanism, and the lowest scored design in terms of universality is the climbing robot.

4. Conclusions

Uninterruptable electricity supply to a vast number of customers has utmost importance for power companies. Therefore, regular live line inspection is a vital task for the power companies to find damage to the line to prevent likely blackouts. The most desirable inspection method in dangerous live-line environment is certainly a fully autonomous method without human operator intervention.

Traditional patrol inspection using helicopters is costly, do not provide a platform for precise inspection of the line from close distances, and due to the risk of contact with the live line in windy climates still is not sufficiently safe for the operators. In that, researchers have endeavored to build robots capable of performing the live line inspection autonomously.

Unmanned aerial vehicles and cable climbing robots are the two main categories of the power line inspection robots. UAVs are complex in control and do not provide an appropriate platform for internal inspection of the conductors. For instance, corrosion detector sensors, which examine the line from a close distance, cannot be used on such platforms. However, as UAVs fly over the power lines, their design is not directly dependent on geometrical characteristics of the lines. Moreover, these flying robots have been already commercialized and can be modified for live line applications. The second main category, the climbing robots, are much more dependent on the line equipment than the flying robots in terms of size and shape of the obstacles on the lines, surrounding electromagnetic field, voltage level of the line, etc.

Climbing robots should be specifically designed for the power lines from scratch. They are not currently available on the market, but they provide more autonomy and inspection data quality. The climbing platforms also allow more accessibility to variety of inspection sensors than the flying robots. This unique feature has been made research in the cable-climbing mechanism design popular.

This chapter reviewed some of the main efforts made over the past 20 years in the field of cable-climbing mechanism design to provide a basis for future developments in this field. History of the research in this field shows that due to the huge benefit of early detection of likely damage to the line, even the cable-climbing robots capable of only climbing on part of the line between two obstacles are in use, and further researches in this field will definitely benefit the power companies to efficiently manage their assets. In addition, based on the reviewed works, a flying-climbing platform which is a commercially available UAV modified with a cable-climbing mechanism would enormously benefit the line inspection quality and the design universality.

5. References

Aggarwal, R. K.; Johns, A. T.; Jayasinghe, J.A.S.B. & Su, W. (2000). An overview of the condition monitoring of overhead lines, *Electric Power Systems Research: An International Journal,* No. 53, 2000, pp. 15-22, Elsevier Science S.A., PII: S0378-7796(99)00037-1

Aoshima, S.; Tsujimura, T. & Yabuta, T. (1989). A wire mobile robot with multi-unit structure. *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pp. 414-421, Tsukuba, Japan, September 4-6, 1989.

Campos, M. F. M.; Bracarense, A. Q.; Pereira, G. A. S.; Pinheiro, G. A.; Vale, S. R. C. & Oliveira, M. P. (2002). A mobile manipulator for installation and removal of aircraft warning spheres on aerial power transmission lines. *Proceedings of the IEEE International conference on Robotics and Automation*, pp. 3559-3564, Washington DC, USA, May 2002.

Cormon Ltd Corrosion Monitoring Systems (1998). Overhead line corrosion detector. Data Sheet No. PTP 002, Rev. 04/98, available at: http://www.cormon.com

de Souza, A.; Moscato, L. A.; dos Santos, M. F.; de Britto Vidal Filho, W.; Ferreira, G. A. N. & Ventrella A. G. (2004). Inspection robot for high-voltage transmission lines. *ABCM Symposium Series in Mechatronics,* Vol. 1, pp. 1-7, 2004, Available from: www.abcm.org.br/symposiumSeries/SSM_Vol1/Section_I_Robotics/SSM_I_01.pdf , Accessed: 27-Sep-2008.

Fu, S.; Li, W.; Zhang, Y.; Liang, Z.; Hou, Z.; Tan, M.; Ye, W.; Lian, B. & Zuo, Q. (2006). Structure-constrained obstacle recognition for power transmission line inspection robot. *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 3363-3368, Beijing, China, October 9-15, 2006, Available from: http://ieeexplore.ieee.org/

Golightly, I. & Jones, D. (2005). Visual control of an unmanned aerial vehicle for power line inspection, *Proceedings of The 12th International Conference on Advanced Robotics (ICAR '05),* pp. 288-295, Seattle, WA, USA, July 18-20, 2005, Available from: http://ieeexplore.ieee.org/

Higuchi, M.; Maeda, Y.; Tsutani, S. & Hagihara, S. (1991). Development of a mobile inspection robot for power transmission lines. *The Journal of Robotic Society of Japan,* Vol. 9, No. 4, pp. 57-63 (in Japanese)

Katrasnik, J.; Pernus, F. & Liker, B. (2008). New robot for power line inspection, *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics (RAM),* pp. …-…, Chengdu, China, September 21-24, 2008.

Montambault, S. & Pouliot, N. (2007). Design and validation of a mobile robot for power line inspection and maintenance. *Proceedings of the 6th International Conference on Field and Service Robotics (FSR),* pp. 1-10, Chamonix, France, July 9-12, 2007, Available from: http://hal.inria.fr/inria-00194717/fr , Accessed: 13-Oct-2008.

Nayyerloo, M.; Yeganeh-parast, S. M. M.; Barati, A. & Saadat Foumani M. (2007). Mechanical implementation and simulation of MoboLab, a mobile robot for inspection of power transmission lines. *International Journal of Advanced Robotics Systems,* Vol. 4, No. 3, 2007, pp. 381-386, ISSN 1729-8806

Pouliot, N. & Montambault, S. (2008). Geometric design of the LineScout, a teleoperated robot for power line inspection and maintenance. *Proceedings of The IEEE International Conference on Robotics and Automation,* pp. 3970-3977, Pasadena, CA, USA, May 19-23, 2008, Available from: http://ieeexplore.ieee.org/

Ren, Z. & Ruan, Y. (2008). Planning and control in inspection robot for power transmission lines. *Proceedings of The IEEE International Conference on Industrial Technology (ICIT),* pp. 1-5, Chengdu, China, April 2008, Available from: http://ieeexplore.ieee.org/

Sato Kensetsu Kogyu Co., Ltd. (1993). Automatic overhead power transmission line damage detector. available at: http://www.sato-k.co.jp/technology/sonshou-e.html accessed: 6-Oct-2008.

Sawada, J.; Kusumoto, K.; Munakata, T.; Maikawa, Y. & Ishikawa Y. (1991). A mobile robot for inspection of power transmission lines. *IEEE Transactions on Power Delivery,* Vol. 6, No. 1, pp. 309-315

Sun, C.; Wang, H.; Zhao, M. & Liu, H. (2006). 3D simulation and optimization design of a mobile inspection robot for power transmission lines. *Proceedings of The 6th World Congress on Intelligent Control and Automation,* pp. 8986-8991, Dalian, China, June 21-23, 2006, Available from: http://ieeexplore.ieee.org/

Tang, L.; Fang, L. & Wang, H. (2004). Development of an inspection robot control system for 500kV extra-high voltage power transmission lines, *Proceedings of The SICE Annual Conference,* pp. 1819-1824, Sapporo, Japan, August 2004, Available from: http://ieeexplore.ieee.org/

Tsujimura, T. & Morimitsu, T. (1997). Dynamics of mobile legs suspended from wire. *Robotics and Automation Systems: An International Journal,* No. 20, 1997, pp. 85-98, Elsevier Science S.A., PII S0921-8890(97)00019-5

Williams, M. (2000). Investigation on machine vision and path planning methods for use in an autonomous unmanned air vehicle, Ph.D. Thesis, School of Informatics, University of Wales, Bangor, England, December 2000.

Wolff, A. and Khajepour A. (2001). A crawling robot for high voltage power lines inspection, Senior Design Project, University of Waterloo, Department of Mechanical Engineering, Waterloo, Ontario N2L 3G1, Canada, 2001.

Zhu, X.; Wang, H.; Fang, L.; Zhao, M. & Zhou, J. (2006). A novel running and gripping mechanism design based on centroid adjustment, *Proceedings of The IEEE International Conference on Mechatronics and Automation,* pp. 1471-1476, Luoyang, China, June 2006, Available from: http://ieeexplore.ieee.org/

5

Bionic Limb Mechanism and Multi-Sensing Control for Cockroach Robots

Weihai Chen[1], XiaoQi Chen[2], Jingmeng Liu[1], Jianbin Zhang[1]
[1]*Beijing University of Astronautics and Aeronautics, China*
[2]*University of Canterbury, New Zealand*

1. Introduction

In the past twenty years, robotics has undergone rapid development. Through the usage of new technologies and new materials, robots have found widespread applications in a variety of industry sectors. In contrast to traditional robots, a new branch of biologically inspired robots has emerged strongly. By studying biological systems like insects, robotics researchers strive to seek in-depth understanding of different kinds of bionic movement phenomena that exist in nature.

The rolling of a wheel is regarded as a mode of motion with supreme efficiency, but it is only suitable for robot to move on a flat surface. Leg type locomotion is much more desired for a robot to transverse on a rugged topography, and is considered more superior than wheel type in rough terrain. Leg type is a common movement in nature, from human to insects, with similar walking modes. These biological systems do not roll, but have biped or myriapod alternate walk. In recent years, robotics researchers have shown special interests in insects, hoping to understand the mechanisms of leg type movement and its superior motion on rough terrains as opposed to wheel type movement.

It is found that when running on a complex terrain, animals can adjust themselves to disturbances (Karalarli et al, 2004). This is hard for robot that walks with wheels to realize. In the dire situation of being caught, insects run away very swiftly to escape from the danger. They have extraordinary motion agility. A cockroach can run one step in 50 ms or at a velocity of 20cm/s. It can jump over an obstacle that is as high as three times of its stature, without slowing down. It spooks quickly with very little response time.

Cockroaches can move and run swiftly and smoothly no matter what terrains to transverse. Their movement smoothness, flexible self-control, and regulation ability have aroused researchers' deep interests to seek the secret of the movement. Cockroach's control over its agile movement far exceeds high speed computer control in terms of response time and control functions. Scientists have failed to find satisfactory explanation about cockroach's supreme performance and control ability in adverse conditions. It is foreseen that overcoming this difficulty will be an important breakthrough for many correlative subjects, lead to the emergence of new research methods, and promote the development of intelligent robot technology to the benefit of humankind.

Bionic cockroach robots can be applied broadly to earthquake relief, riot, search and rescue, and space exploration in rugged and unstructured natural terrain. Cockroach's quick reflection and dodge mechanism can be applied to aeroplane collision avoidance. To bring biorobots a step closer to real applications, novel locomotion mechanism, multi-sensor information processing and fusion, and intelligent control must be thoroughly investigated to emulate cockroach's moving rapidity, flexibility and stability. Cockroach limb is intriguing, and has calculation function and local intelligence. Intensive research is required to mimic the mechanism configuration and physiological characteristics of insect limb.

A collaborative research project between Beijing University of Aeronautics and Astronautics and the University of Canterbury has been launched to improve bionic robot's movement mobility and intelligent control level through multi-sensor fusion. It explores all-weather vision sensor, head palp and leg feathers based on fibre optic and tactile sensing; and studied multi-sensor information processing and fusion in depth for the realization of cockroach robot's intelligent control. It proposes Backstepping disperse adaptive control scheme based on robust information filter to solve control difficulty caused by nonlinear system modelling error, and disturbance factors. Correlative theory of bionic cockroach robot and its application features contribute towards the development of biorobot control theory and application fields in the field of bionic robotics.

This paper firstly gives a brief overview of bionic cockroach robots that have been developed in the past decades. It then discusses the design of bionic mechanism using parallel kinematics and spring joints in offering a realisable approach to emulate the functions of cockroach legs. It is followed by proposed schemes on all-weather vision system and tactile system. It further proposes a distributed multi-CAN bus-mastering system based on FPGA (Field Programmable Gate Array) and ARM (Advanced RISC Machine) microprocessor, and FPGA-based motion control. Finally conclusions are drawn.

2. Overview of Bionic Cockroach Robots

A cockroach has an effective formation of six legs that accomplish complex movements regardless of terrain conditions, as shown in Fig. 1. In rapid movement, the front and hind legs of one side and middle leg of the other side forms a group. As a result, three points distribution ensures excellent movement stability (Bai et al, 2000).

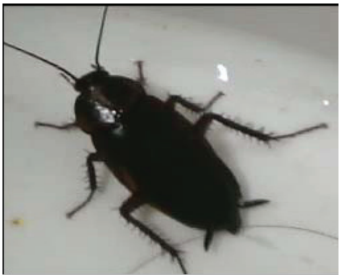


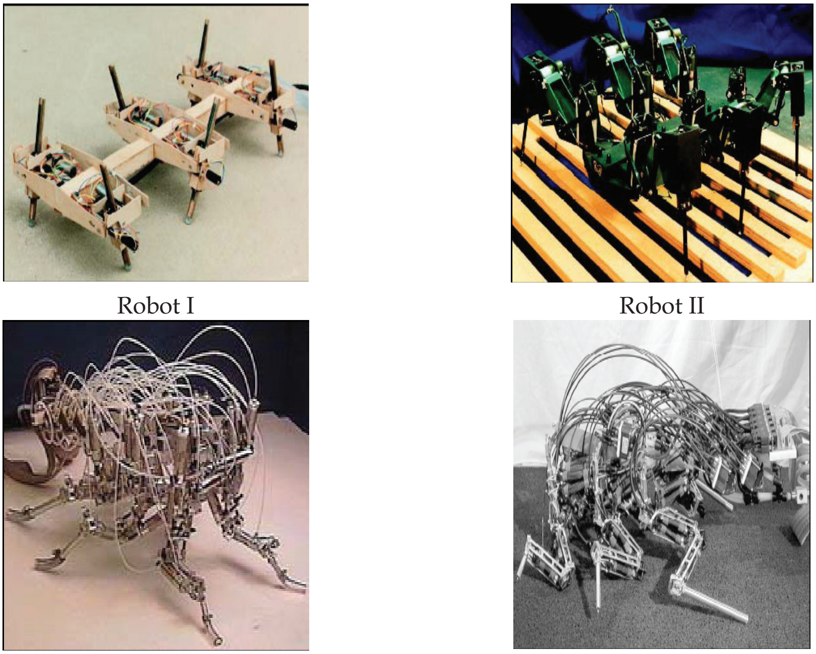
Fig. 1. Cockroach with six legs spreading out.

Biologists at the University of California at Berkeley have experimented on cockroach's adjustment to external disturbances. A very small cannon was fixed to a cockroach. A lateral thrust with a duration of 10 milliseconds is exerted to the cockroach using recoil of cannon.

It was found that this cockroach had adjusted its body balance well before taking the next new step. It turns out that a cockroach has control balance ability that far exceeds high speed computer control. So far, researchers struggle to find a satisfactory scientific explanation about cockroach's such natural ability. Many multidisciplinary difficult issues involved in the bionics study leads to wild science fascination, and bionic robotics has become an emerging research subject that deserves in-depth exploration.

In the past bionicst had attempted to develop new fashioned vehicles to achieve agile motion on rough terrains by studying cockroach locomotion. At the same time, biologists were intrigued by neural network of cockroach's nervous system which works very effectively despite its simple structure. In September 2004, The National Science Foundation (NSF) announced a $5 million, five-year grant to the University of California, Berkeley. The grant funds biologists, engineers and mathematicians from universities across the country to try to understand the mechanical and neurological basis of locomotion. This signifies an important effort in search of in-depth understanding of cockroach locomotion and its intelligence.

Owing to the special features of cockroach, in recent years robotics academics have devoted intensive research effort to the development of cockroach robot. Based on the development in the past decades, cockroach robots can be roughly categorised into three types. The first type is “Robot” series, shown in Fig. 2.


Robot I

Robot II

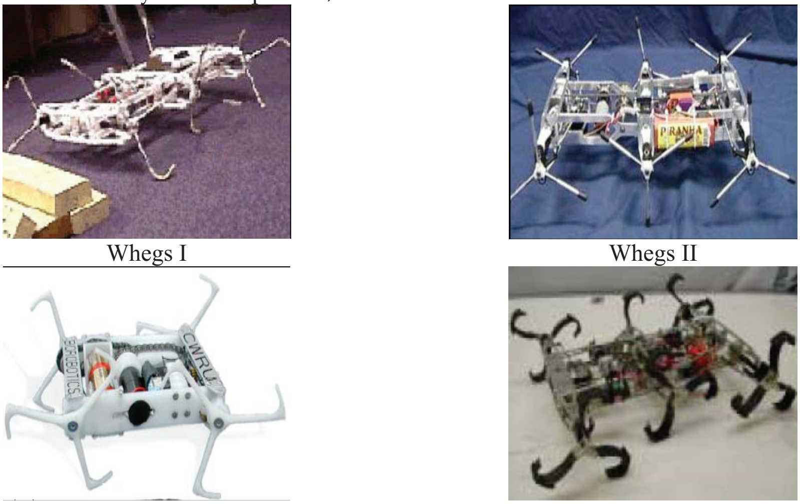
Robot III

Robot IV

Fig. 2. Robot series of cockroach robots

In this series, the contour design, developed by Case Western Reserve University, has features of bionic mechanism. Its first generation prototype Robot I is made of six legs of two degrees of freedom (DOF) and driven by Direct Current (DC) motors. It has a bionic neural network controller. In the second generation prototype Robot II, each leg has 3 DOF. The robot adopts a distributed control system to achieve coordinated motion of legs and insect gait. Robot's physical height and orientation are taken charge by a central controller. In the third generation prototype Robot III, the number of degrees of freedom is distributed among the six legs unequally. There are a total of 24 DOF, with each of two front legs having 5-DOF, middle leg 4-DOF, and hind leg 3-DOF. Furthermore, pneumatic drives are employed to obtain a larger explosive force. Such design makes front leg most agile, and the leg has guiding and sensor function. Middle leg is agile enough to hold up and turn the robot. Hind leg is least agile, and is provided with a strong propulsive force. (Quinn & Ritzmann, 1998). The fourth generation prototype Robot IV is based on Robot III, and uses artificial muscle to approach biologic bionics (Quinn et al, 2003).

The second type of cockroach robots is represented by Whegs that is also developed by Case Western Reserve University. The Whegs series of cockroach robots is shown in Fig. 3. The contour design does not follow cockroach's bionic mechanism. Instead it takes advantage of both wheel type movement and leg type movement, then abstracts features of cockroach movement to carry out mechanism design and function simulation (Schroer et al, 2004; Quinn et al, 2004). Whegs robot is thus based on abstract biology theory (Allen et al, 2003). In Whegs I, each leg has only one rotational DOF for wheel type movement. It is built with a tri-spoke structure which allows for movement on rugged road. Whegs II can bend its body to have better movement agility than Whegs I. Whegs III's structure has the feature of micro motion and it has only four tri-spokes instead of six. The terminal of its tri-spoke has barbs which facilitate wall climbing and moving on rugged road. Whegs IV is similar to Whegs II, but it does not have flexible joint as in Whegs II. Its main improvement is that its tri-spoke structure has flexibility and compliance, which benefits transversal motion.



Whegs I

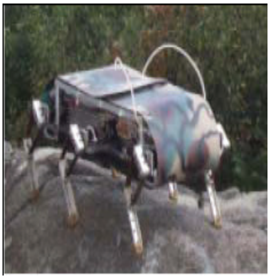
Whegs II

Whegs III

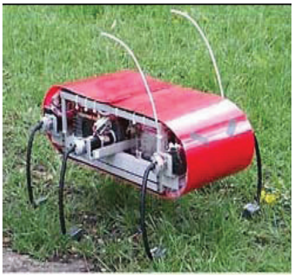
Whegs IV

Fig. 3. Whegs series of cockroach robots

The third type of cockroach robots is RHex, developed by University of Michigan, UC Berkeley and McGill University. Fig. 4 shows the prototypes developed through different stages of development. RHex is similar to Whegs in that 6-DOF bionic cockroach movement is achieved based on abstract biology theory (Sarandli et al, 2000). The first generation prototype Rhex 0.0 was developed in 1999. Then Rhex 0.1 and RHex 0.2 were improvements on RHex 0.0. The improvement of RHex 0.1 is mainly the mechanism, resulting in a 10% weight reduction. Rhex 0.2 incorporates tactile sensors to detect surrounding environment (Moore et al, 2002; Komsuglu et al, 2001). In recent years, controller with gait adaptive algorithm further enhances function of RHex (Weingarten, 2004).



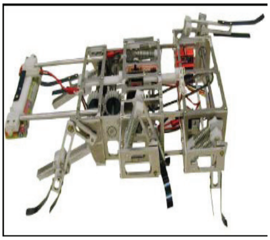
RHex 0.1



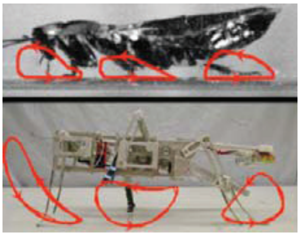
RHex

Fig. 4. RHex series cockroach robot

Case Western Reserve University has further studied cockroach's mechanical structure, and developed MechaRoach. The robot’s walking and wall climbing functions are built according to the mode of cockroach's bionic motion (Wei et al, 2004). Fig. 5(b) compares the leg structure of MechaRoach with a biologic cockroach in different scale factors. Different from Robot series mechanism bionics, MechaRoach has only one initiative joint in each leg, and it realizes mechanism bionics through crank rocker (Boggess, 2004). Such motion function is not as versatile as those in the previous three types of cockroach robots.



(a) MechaRoach



(b) Comparison of legs structure

Fig. 5. MechaRoach cockroach robot

Currently the research focus of cockroach robot is mainly on motion agility and smoothness control. As far as leg mechanism is concerned, bionics research mostly deals with simulation of cockroach leg structure topology that treats hip joint, knee joint and ankle joint as serial mechanism. Latest human bionics analysis manifests that human shoulder joint movement depends on muscular cable-driven to drive joints. Such cable drive is an inter-coupled parallel mechanism linking shoulder, elbow and wrist.

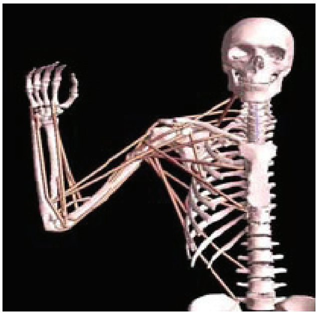


Fig. 6. Sketch of human arm drive

It is found from the structural analysis of cockroach leg that Cockroach motion control is accomplished by a parallel structure. What impact does a parallel structure have on cockroach's motion state? Does it help cockroach's stability of motion, load capacity and skip ability? Does it cause adverse impact on velocity of movement? Discovering these answers may provide new clues for humans to explore the secret of cockroach motion ability. In this regard, developing parallel mechanism to emulate cockroach leg has become an important premise for cockroach motion control.

Conventionally robot links and joints are rigid. But biologic limbs and joints are flexible. Cockroach leg's flexibility provides spontaneous regulation of movement stability with little computation in the brain. Hence, constructing cockroach legs with flexible materials to emulate the bionic functions is indeed a significant step towards bionic legs. So far, there have been reports about using insulation elastomer man-made muscle driver (Kornblush et al, 2002). Although flexible material brings benefits to motion stability, it faces some practical difficulties in motion control.

Cockroach has superior movement adaptability to differtent terrain conditions. Such mobility is attributable to two bionics aspects: i) cockroach body and legs distribution of structure, and ii) fast sensor information processing capacity and virtual intelligent control. In the latter, touch and vision sensing plays an important role in realizing the intelligent motion control, and hence requires in-depth research. In addition, sensor information processing empowered by real-time computation and multi-sensor information fusion allows a cockroach robot to perceive its operational environment responsively and accurately. These two aspects of sensing and information processing are challenges that must be overcome to mimic bionic operations of a cockroach.

Although robotic vision has been widely used in a controlled environment, how to make it adaptable to varying and unknown environments remains to be a challenge. Human visual system is highly robust, and removes redundant information and extracts useful information that is related to current visual behavior from profuse visual information. As an illustration, once fixing their point of attention to an object, eyes follow the moving object so that the object is imaged in the central fovea (sharp image) all the time. When the object exceeds the physiologic limit that eyeball can turn, human will resort to head, even body's turn to follow the object and obtain a clear image as far as possible.

When humans observe a scene, eyes often focus on some characteristic points of the scene. Fourier spectrums of these characteristic points contain many high frequency components and amounts of information. If we can control eyes and make eyes fix on the goal of interest, then the speed of image processing can be enhanced greatly. For a complex system that contains multi-sensor information, owing to modeling error, external disturbance, load fluctuation and temporary set causing position error, state space variables are inter-coupled. Consequently it is very challenging to design realizable filter and controller in system state space. Effective methods have to be devised to realize multi-sensor information processing, hence intelligent motion control.

Although a cockroach has agile movement, its nervous system is very simple (Beer et al, 1997). The limited capacity of the simple neural network shows that the nerve centre does not take care of everything itself, and that each leg's movement is self-controlled by each leg's controller. It is therefore suggested that cockroach robot control system should adopt principal and subordinate distributed control structure (Espenschied, 1996). Master controller of brain level allocates task for each master controller of leg level based on programming task requirement. Master controller of leg level sends commands to its subordinate controllers which are the individual joint controllers. There are information interaction between principal and subordinate controllers to realize intelligent motion control. Control algorithms should be simplified to accelerate controller's operational speed.

Cockroach robot has more than 10 years research history, but it is still in its infancy. In the ongoing project, the concept of region control has been proposed. It is designed to substitute routine point control scheme. Region control has many examples in life, such as chess player placing chessman. Players do not need to place chessman in the decussate point exactly, but a region near the ideal point. It is obvious that the point is the limit of region and reducing the region leads to the point. Intuitively it can be concluded from the problem of placing chessman that region control is easier than point control and requires much less computational time. The velocity of body's movement using region control can be faster. Ascertaining the size of the region of interest based on task requirements helps a cockroach robot achieve movement rapidity and flexibility.

3. Design of Bionic Limb for Smooth Motion

3.1 Multi-Discipline Fusion Approach

In the multi-discipline fusion approach, bionics, mechanism and disperse adaptive control theory are combined to realize harmonious development of bionic mechanism and modern control theory. Preliminary research indicates that dynamics characteristic of organism incarnates bionic dispersed intelligence. Disperse adaptive control theory and technology, which studies bionic mechanisms, extends biologic dispersed intelligence to artificial intelligence. Multi-discipline fusion approach offsets single subject’s difficulty caused by limitation of technology. Combining cockroach robot’s leg configuration and calculation function draws the knowledge from math, mechanics, mechanism, artificial intelligence, electronics and control theory.

Bionic cockroach robot can be viewed as an integrated sensing, opto-electro-mechanical (OEM) system with real time adaptive control. Such an OEM system should have small volume, high precision and good real-time performance. Commercial sensors like mechanical sensors and light frequency and phase modulation sensors cannot be easily deployed in bionic limb design due to the volume factor. Electric and magnetic sensors are small and sensitive, but have limitations in reliability and anti-interference electromagnetism stability. Especially for a cockroach robot, its figure should be gracile, and be easily integrated in arrays. Such a robot should have good dynamic response, high sensitivity and strong anti-interference electromagnetism ability. Based on synthesizing all factors, differential light intensity modulation fibre optic sensor becomes a preferred sensor of the cockroach robot sensor system.

3.2 Biomimetics Approach

Bionic cockroach robot can be considered as a parallel mobile robot with six legs. Development of such a robotic system needs to cover robot's configuration design to achieve dexterous movement; and modelling and dynamics control of a cockroach robot with optimal configuration. This work aims to analyze existing techniques in design of parallel kinematic machines, and conduct fundamental research and innovative mechanism design to achieve motion smoothness in cockroach robots.

In consideration of joint drives of hip, knee and ankle moving in partial coupling, configuration design, dexterous workspace, and design fundamental of optimal coupling are to be ascertained. In terms of modelling, kinematics and dynamics control, high redundant arithmetic control and analysis of singularity workspace and control arithmetic of avoiding singularity need to be studied.

In developing a bionic mechanism, optimum coupling design hip, knee and ankle joints, corresponding to the model of system and analysis of movement, is necessary. The core of this problem is to discover the secret of cockroach's movement mobility based on mechanism theory, to study high redundant control arithmetic and mechanism design when being overdriven, and to supply hardware base for the realization of bionic cockroach robot.

In terms of leg mechanism design, two approaches are considered; i) bionics approach, and ii) abstract transplant approach. The base of bionic cockroach robot's mechanism design of hip, knee and ankle comes from illumination of research on hip, knee and ankle joints of cockroach. There are two bionic methods: one is mechanism biomimetics, the other is function biomimetics. Function biomimetics is combined with mechanism biomimetics for the design of cockroach robot's leg configuration. Not only does such a combinatorial design method assimilate the merit that biologic cockroach's limb mechanism possesses movement agility and smoothness, but also overcomes the difficulty that complete imitation of cockroach's structure and functions is impossible because of technological limits.

The abstract transplant approach is to emulate cockroach limb being elastic. It isimpossible for a stiff pole to realize limb mobility. While it is difficult to find an elastic material having similar properties to cockroach's limb, limb's local function can be simulated with a spring mechanism which would greatly simplify the leg mechanism design.

Before the detailed design, theoretical analysis of leg mechanism of cockroach robot, kinematics and dynamics modelling are carried out. It is followed by simulation and experimental verification. Verifying the correctness of the theoretical model via simulation can economize time and cost, and is simple and effective. Further experimental studies and prototyping are conducted to validate rationality and correctness of correlative theory and arithmetic, improve the design and reliability, and provide the feedback to refine the theory.

3.3 Design of Bionic Joints

3.3.1 Design of Bionic Hips and Knees

Based on physiological characteristics of cockroach's front, middle and hind legs, the motion feature of each leg is observed. Front leg is deft and mainly used to turn and adjust body pose. Middle leg is swift and mainly serves to hold and turn body. Hind leg is strong and powerful, and drives a cockroach to walk. The mechanisms of front, middle and hind legs are designed differently to suit the motion characteristics. Fig. 7 illustrates front, middle, hind legs structure.

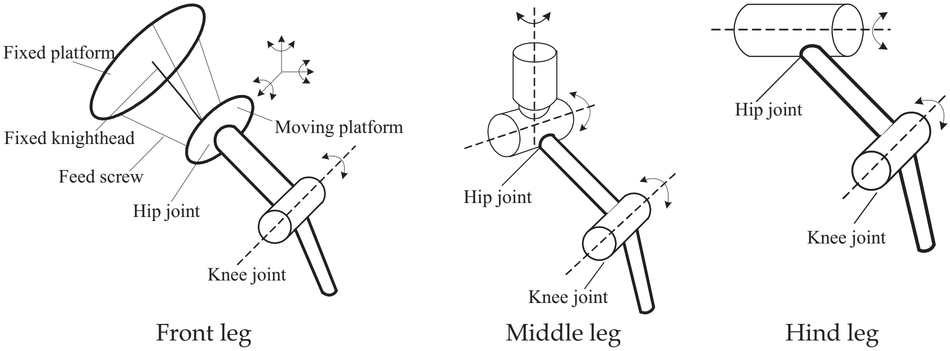


Fig. 7. Structure sketch of cockroach robot's leg

In the front leg, hip joint is designed to be 3-DOF globe joint, and knee joint to be 1-DOF rotary joint. In the middle leg, hip joint is a 2-DOF rotary joint, and knee joint 1-DOF rotary joint. As for the hind leg, hip and knee joint are all designed to be a 1-DOF rotary joint. Ankle joints of all legs are fixed structure. Altogether 18 degrees of freedom are required to realizing cockroach robot's agility function completely. The 3-DOF in the front leg hip joint is important for cockroach robot mobility and terrain adaptation. In this project, the concept of parallel kinematic globe joint is proposed to realise the front leg hip joint.

The concept of globe joint emulates biomimetics exhibited by biological systems. Human hip and shoulder joints are globe joints. They contain all rotational DOF of Euclidean space, and therefore have outstanding movement rapidity and mobility. A common approach in designing bionic leg is that the robot globe joint is approximated by two 2-DOF joints that have two orthogonal axes and the link is constructed as a 1-DOF rotary joint. This work proposes a scheme that realizes globe joint function by three parallel telescopic mechanisms. The drive for the telescopic mechanism may adopt one of three feasible methods, namely, air cylinder, pneumatic artificial muscle and feed screw. In this work, feed screw driven by motor is chosen for its simple motion control.

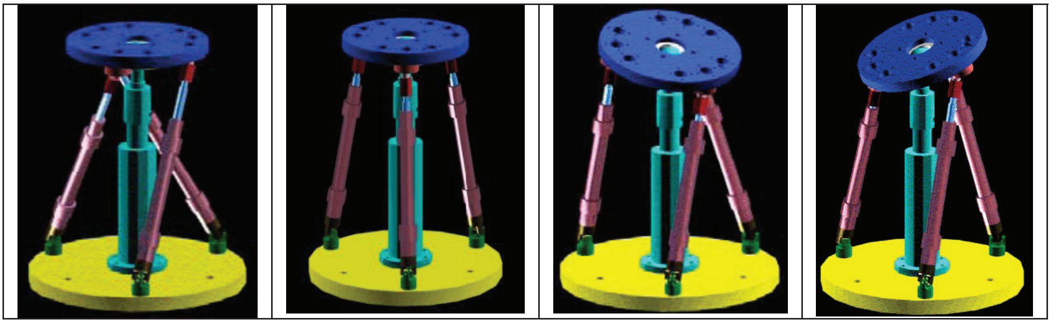


Fig. 8. Sketch of globe joint

3.3.2 Design of Bionic Flexible Joints

Most biological organisms are flexible. It is one of the main reasons why an organism can easily complete all kinds of difficult movements. Such a built-in flexibility in robot joints would allow a robot to move reposefully. For the bionic cockroach robot under development, flexibility is in-built at globe joint and rotary joint. Front leg flexible globe joint, shown in Fig. 8, comprises a moving platform, a fixed platform and four knightheads that connect the two platforms. Moving and fixed platforms are two disks with different diameters. The centres of moving and fixed platforms are connected by an invariable knighthead while the other three knightheads are connected by telescopic feed screws.

Fig. 9 illustrates the single feed screw connection of globe joint. A flexible element is installed between feed screw nut and joint matrix, which makes front leg of cockroach flexible. The parallel link is coupled to the fixed and moving platform through universal joints. Fixed coordinate is placed in triangular centre of lower platform.

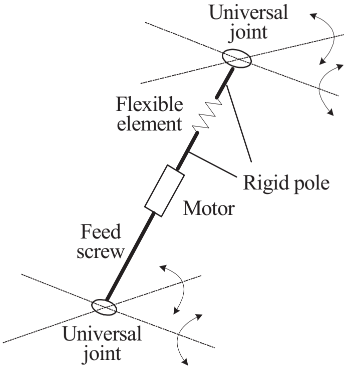


Fig. 9. Design of bionic cockroach robot’s flexible joint

The model of universal joint is shown in Fig. 10. The two axes of rotation in the universal joint are two orthogonal axes of the plane that the fixed or moving platform belongs to. The outer ring turns relative to the ground along axis 1, while the inner ring turns relative to outer ring along axis 2. Proper setting of flexible element can be used to fix the other rotational DOF. Elastic material can be used to design the foot, and this will make the bionic robot more adaptable to terrains.

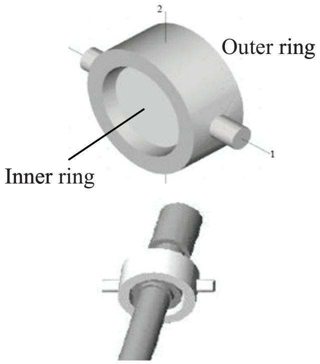


Fig. 10. Globe joint feed screw connection

3.4 Parallel Driver Structure

A driver structure would affect each leg and unitary movement performance of cockroach robot if the coupling between hip and knee joints is weak. Before the design of mechanism, modelling and movement analysis of the bionic system are carried out.

Different mechanism coupling modes are studied by using graph theory.

Change of configuration is analyzed to seek best description method of coupling mechanism, and studied with structurology, kinematics and dynamics.

Universal kinematics and dynamics models containing geometry and movement restriction are established.

The effect of singularity configuration on coupling mechanism form is analysed.

Self-motion manifold under high redundancy condition, and mission-oriented optimal control are formulated.

The dynamical equation for single body is established using the Newton-Euler method. Then multi-body dynamical equation is then established. Constraint counterforce can be eliminated by substitution.

At the initial stage of designing biorobot, 3D robot modelling, dynamic performance and control simulation are integrated using virtual prototyping technology. Firstly, apply modern design theories to biorobot domain and establish 3D dynamic simulation.

Secondly, establish a model to finalise biorobot performance analysis and obtain test data in order to improve biorobot system design performance, economize physical prototype, finalise the design and simulation platform for the design and theoretical analysis of biorobot.

Thirdly, establish mechanical model and dynamical model of the biorobot using virtual prototyping technology. Biorobot overall performance is forecasted, and the feasibility of trajectory is verified. Movement simulation and statics, kinematics and dynamics analysis are carried out to achieve necessary displacement, velocity, acceleration, force and moment curve. As such, optimal joint configuration is obtained, and physical design of the prototype is optimised to improve the overall performance of the biorobot.

4. Multi-Sensing in Bionic Cockroach Robot

A cockroach has an exceptional ability to navigate freely in all-weather conditions. In addition to its visual navigation, it has a powerful detection system built into its legs and feelers to detect its contact states with the environment. These sensing and navigation abilities are important for biologically inspired robot which needs to execute demanding tasks in difficult situations, such as search and rescue, homeland security, logistics in natural disaster, etc.

Multi-sensor information fusion technology is the key to realize intelligent motion control of cockroach robot. To ensure the fidelity of time-dependent sensor information, the information processing has to be carried out in real time. In face of a large amount of information including visual images, the real-time processing becomes very difficult. Cockroach has visual, tactile, taste, smell sense function, etc. For practicality, only visual and tactile sensors are considered at this stage. The vision system mainly utilizes infrared imaging sensors, and the tactile sensing system is built upon optical fibre sensors.

4.1 Development of All-Weather Visual Navigation Systems

4.1.1 Imaging Device

Infrared imaging technology has been widely used for sensing natural environment where a robot operates. For a bionic cockroach robot to emulate its biological counterparts, its visual sensing system must satisfy two requirements, i) real-time binocular stereo image acquisition, and ii) real-time high precision 3D imaging processing and recognition. These would equip the robot with all-weather situational awareness and judgment ability.

Non-scan infrared imaging system and multivariate array infrared detector are able to provide real-time environment image. The fundamental is that infrared radiation power is converted to electrical signal detected by the detector. After being amplified, the signal is converted to a video standard signal. One disadvantage of commonly available infrared imaging devices is that they are large and cumbersome. There is a need to improve on the size of optical lens, develop integrated optics, and subsequently miniaturise the entire imaging system.

4.1.2 Calibration

The calibration process aims to establish the relationship between two views in order to extract 3D visual information about the operating environment. Imaging aberrance presented in real life and caused by lens affects the accuracy of image processing results. Matching of two correlation images in the binocular visual device, and abstraction of image characteristic points require data fusion. The calibration process associates images captured by two video cameras that are unattached, and to abstract common information for restoring the fidelity of imaging information.

4.1.3 Recognition

Identification of image feature points and reconstruction of three-dimensional entity data are the key to the visual navigation ability of a cockroach robot. Changing of actual imaging circumstance will lead to the different imaging effect and excursion of characteristic points in binocular imaging. This would cause the probability problem in truthful identification of image features.

So far there has been no practical system that could recognise natural features in all-weather conditions reliably. Therefore, it is necessary to develop a robust and novel detection algorithm that combines high processing speed and efficiency. Wavelet algorithm can be applied for navigation of mobile robots.

The image recognition involves image segmentation, identification and movement judgment. Optimal internal and external imaging parameters can be solved using Tsai imaging model. At the same time, binocular image matching handles standard imaging template and imaging characteristic points. Then correlation pre-processing of images is carried out to reduce image noise and enhance background contrast.

In the abstraction and identification of image characteristic points, the image processing system adopts the wavelet arithmetic to identify contour objects of obtained visual image. It also recovers object shapes and obtains the 3D shape outline of the object by utilizing binocular visual demarcation and data fusion. It further modifies the current 3D model and obtains the position error and external position error of the object by comparing the preliminary 3D model with the current 3D model obtained from video images. Filtering out the false characteristic points, true characteristic points based on current 3D model can be obtained.

4.2 Development of Novel Tactile System

Cockroach's powerful sensing abilities are further enhanced through its tactile perception (leg pressure sensing) of the environment. Indirectly a cockroach senses the leg velocity, high-frequency vibration, surrounding wind velocity, contact softness, ground condition, obstacles, etc. At present there is not much research work that studies the function of fuzz in cockroach's limb. To emulate some of the tactile abilities of a biological system, a highly integrated tactile system with good stability and high precision need to be developed to satisfy the navigations needs of cockroach robot.

4.2.1 Fibre Optic Sensing for Cockroach Robot Tentacles

Fibre optic sensors are identified as a potentially suitable candidate to emulate the sensing functions of cockroach leg feathers and head tentacles. Light intensity modulated fibre optic sensor has small volume, high precision and real-time characteristics. Considering the fine structure of cockroach leg feathers, supersensitive light intensity modulated fibre optic sensors are deployed. The sensing system collects real-time data of pressure, vibration, direction of wind and contact softness, which are produced by the robot's leg movement and its contact with the environment. Composite signal is obtained using optical fibre array. Through multi-path signal processing based on difference measurement step by step, environmental information can be extracted and situational awareness can be achieved.

To mimick the function of cockroach head tentacles, high strength optical fibre is adopted. Changes of light intensity caused by the change of external pressure, wind direction and vibration are thus detected in real time. These physical changes are converted to electronic signals which can be processed internally using photoelectricity transition array. Besides tactile sensor, head tentacles incorporates non-contact near-infrared ranging sensor to enhance the robustness of locating objects and detecting obstacles in a poor visual environment. This sensing approach adopts a specific type of bismuthate optic fibre which is controllable and has near infrared high degree of transparency to guide and focalize light. Thereby suitable ranging position can be selected by changing the position of the head of optic fibre. For illuminant and photosignal transition devices, integrated near infrared semiconductor illuminant, which has a high contrast from background environment light, and photoelectric conversion semiconductor array are chosen respectively.

4.2.2 Cockroach Robot's Tactile System - Leg Feathers and Head Palp

Inspired from tentacles of rodents, a tentacle sensor based on the Position Sensitive Detector (PSD) and Laser Diode (LD) has been designed. The sensor uses PSD as the sensing element; and LD as the incidence light source. The sensor has certain advantages including compact structure; light weight; and ease of processing, assembling, and debugging.

The 2D PSD element, measured 3 mm × 3 mm, can detect the rotation displacement and direction of tentacle simultaneously. To be able to detect texture and roughness of an object in contact, the tentacle of the sensor is designed to be thin poles made of flexible material of 9 ~ 15 cm in length. A light-shading film is fixed near the root of the tentacle, about 2 ~ 3 mm away from the root, In addition, a small hole with a diameter of 0.8 ~ 1.2 mm is opened from the film to receive the light from LD. Through this mechanical design, the tentacle automatically returns to its initial position if it is not in contact with the object to be measured. In a sense, the flexible element resets tentacle.

The PSD is installed on the side opposite the LD. Therefore the sensor can detect the root displacement of tentacle forming on the X-axis and Y-axis of light-shading film. As a result, a voltage signal is output, representing the mechanical displacement of the root of tentacle caused by the bending of the tentacle.

If the tentacle is in contact with an object, a bending deformation is produced on the tentacle. The deformation force causes the movement of the light-shading film fixed on the root of tentacle. As a result, the location of the incidence light spot irradiating onto the photosensitive surface of PSD changes, and the PSD produces an increase in current which represents the change of displacement and direction of the tentacle. By converting current to voltage in the signal conditioning circuit of PSD, a corresponding voltage increment is taken. Then data collection and processing can be carried out to calculate the tentacle movement.

5. FPGA-Based Information Processing and Motion Control

5.1 Control system based on FPGA and ARM

Field Programmable Gate Array (FPGA), essentially logic cells, facilitates real-time processing and control arithmetic of tactile and visual multi-sensor information. Multi-heterogeneity FPGA combines a large number of FPGAs taking charge of different tasks. These tasks include central processing unit in charge of operation, data collection, logic management, etc. The distribution and scheduling of the tasks have great effect on the speed of FPGA.

The control system hardware structure comprises three core parts: Advanced RISC Machine (ARM) processor, distributed multi-CAN bus-mastering system based on FPGA, and CAN bus controller and CAN bus servo driver which controls robot joints. The architecture provides stage treatment for control information and real-time servo control. It solves multi-joint coordinated control of bionic cockroach robot joints, and effectively reduces requirements for bus bandwidth in the networked control system.

The core of FPGA integrates multi-CAN bus controller utilizing System on Programmable Chip (SOPC) technology. Each CAN bus is connected with 3~5 control nodes according to load requirement. Control nodes are either network servo motor drive based on CAN bus or various sensors. Data of multi-path CAN bus can communicated with CAN bus controller at the same time.

The main problem of distributed control system is synchronization of nodes. Multi-CAN bus architecture adopted in this work synchronize the data from all upper computer nodes (CAN bus controller of FPGA). This way, it satisfies broadcasting frame synchronization standards of CAN Servo communication protocol. Thereby CAN Servo communication protocol extending to multi-CAN bus is realized, and CAN bus servo control and synchronization are achieved.

Embedded system platform is formed by ARM processor and Real-Time Application Interface (RTAI). In the cockroach robot control system, the ARM processor adequately performs robot's tactile and visual signal processing, path planning, motion control, etc. RTAI, a real-time extension of Linux, allows a user to write applications with strict timing constraints for Linux. It has easy transplant characteristics, and is well suited for embedded applications. The software system based on ARM and RTAI is divided into non-real time tasks and real-time tasks.

Non-real time tasks are not related to controls running in Linux. They include human computer interaction, upper network communication, system tactile signal and visual signal acquisition, etc. Real-time tasks run in RTAI, such as path planning, motion control and interpolation process, and servo control requiring low-level sensing and position servo information processing. External interrupts utilize hardware interrupts of RTAI to further enhance real-time servo control, which allows for processing of system emergency and changing servo signals in real time. Real-time tasks implemented in RTAI communicate with Linux tasks through RT FIFO provided by RTAI.

The adoption of the embedded distributed network control system has the advantages of both network servo control and centralized control. Its bus controller is based on FPGA, and the topology form adopts star topology and bus topology. Embedded distributed control system based on ARM and FPGA provides information processing at stages and accurate servo control.

5.2 Real-time information processing and transmission

Multi-sensor information fusion based on FPGA technology is adopted to carry out pre-processing and encoding of sensor signals. Then the real time sensor information is transmitted to the robot master controller CAN data bus. The sensor information sources are primarily vision and tactile sensors. In such a complex system containing multi-sensor information, state space variables are inter-coupled owing to modelling error, external disturbance, load fluctuation, and imponderable dithering of cockroach robot's movement causing position error. Thus it is difficult to design and implement filter and controller in state space.

A possible approach being evaluated is to use Backstepping disperse adaptive controller based on robust information filter. The basic idea of this design method is to convert system state space variable to information space variables by robust information filter. System filter and controller can be designed by utilizing structure simplicity of expressing system information space variable. Returning to state space after solving information space, state variables can be solved through inverse transforms. This way, not only does it greatly simplify the design of filter and controller, but also provides a multi-sensor information fusion approach recovering more complete system information from local information.

The research on Backstepping disperse adaptive control scheme involves five steps progressively: i) lumped linear system, ii) disperse linear system, iii) disperse non-linear and model uncertain system, iv) adaptive control of disperse non-linear and model uncertain system by designing robust information filter, and v) K steps advance distributed evaluation arithmetic to effectively cover time delay and design Backstepping disperse adaptive control scheme based on robust information filter.

In developing sensor information processing system and control arithmetic based on SOPC technology, the information processing system is realized through integrating DSP module, RAM, ROM, CPU, etc. into a single FPGA. Data processing is carried out in both software and hardware. Internal hardware circuit employs multi heterogeneity array based on logic cell concept. It adopts building block design. Each module has its own storage and processor. Emulating biologic neural neurons, function modules (FM) are connected by time tag event module (TM). TM acts like the synapsis of human nervous system and is the handshake interface between function modules. External information from FM first enters TM, and TM determines the work mechanism and property of FM.

Each TM communicates with immediate function modules. Each FM's function can be described as different models, such as state oriented model, activity oriented model, structure oriented model and data oriented model, etc. Furthermore new models are formed by combining these models. The general structure of modular neural network system of FPGA is shown in Fig. 11.

It is important to simplify computation in FPGA design. Chip-level optimization is needed to implement control arithmetic to meet the stringent real-time operation requirements of the intelligent control system for cockroach robots

The internal board-level of information processing system bus adopts Xilinx RocketIO™ Multi-Gigabit Transceiver (MGT) - high speed data transmission technology, and accommodates different protocol designs of bandwidth from 622 Mb/s to 3.125 Gb/s per channel. Transceiver supports data rate as high as 3.125 Gb/s per passage and can satisfy various requirements of increasing data transmission rate. Output of information processing system adopts Low Voltage Differential Signal (LVDS) interface, and data output can reach 655Mb/s. Terminal adaptation has low power consumption, low radiation and fail-safe characteristic to ensure reliability.

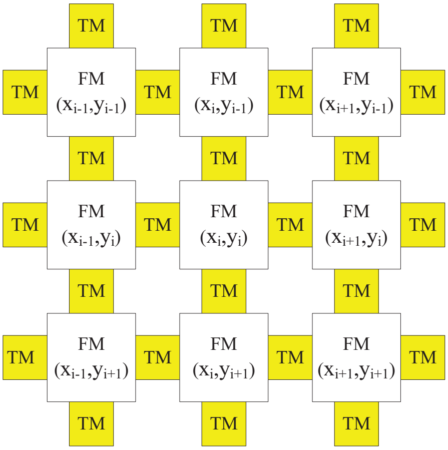


Fig. 11. Modular neural network system of FPGA

The internal information processing and control arithmetic adopts neural network technology, parallel processing of a large amount of information, and large-scale parallel calculation. The information system has plasticity and self-organization. It can realize system's study improvement mechanism when being triggered by external environment incentive conditions. Adaptive error compensation in information processing is achieved by changing internal programmable hardware structure parameters and software arithmetic. Information processing and information storage are combined, which differs from conventional computers whose storage address and content are separate.

5.3 FPGA-Based Motion Controller

Bionic cockroach robot requires high redundancy control. Besides controlling each walking leg efficiently and precisely, inter-harmony of six legs is another difficulty for control system. The control system has to deal with interferences among walking legs, and to complete corresponding movements exactly. Furthermore, bionic function demands that cockroach robot must adopt different movement modes based on different circumstances, which pose further challenges to control system design.

Because of very complex movement and smooth motion control in a highly dynamical environment, conventional dynamics control methods are found to be unsuitable for the movement control for two reasons:

Modelling error, external disturbance, load fluctuation and temporary set of limb may cause position errors.

Processing of a large amount of sensor information may lead to time lag.

For these reasons, backstepping disperse adaptive control method has been proposed for real-time movement control in the presence of position errors and time lag of sensing information. Equally important is to design a method that can execute real-time motion control at high speed.

With the rapid development of the semiconductor industry, SOPC technology has attracted more and more attentions. It is a new comprehensive electronic design methodology requiring skill sets of EDA software, hardware description language, FPGA, computer components and interfaces, assembly language or C language, DSP algorithms, digital communications, embedded systems development, construction, testing on chip system, etc. Comparing with the traditional design technology that has difficulties in meeting the needs of system, network, multimedia, high speed, low power consumption, and other applications, SOPC can integrate functional module such as processor, memory, peripherals and multi-level user interface circuits into one chip. It has been increasingly favoured because of its flexible, efficient and reusable design features.

Cockroach robot is an intelligent system integrating bionics, mechanics, sensing, information processing, and intelligent control. In an attempt to equip a bionic cockroach with intelligence in a small volume, a new chip called "smart brain chip" based on FPGA and SOPC technology has been conceptualized for the prototype cockroach robot. Smart brain chip integrates DSP, memory, and external I/Os. It has the function of motion controller that includes PWM signal to control the speed and position of motor, RS485 communications, wireless network, and sensor data acquisition.

As illustrated in Fig. 12, the motion controller based on FPGA consists of modules such as data instruction interface module, axis management module, digital PID module, T-curve generation module, S curve generation module, data acquisition and processing module, PWM module, synchronized module, and interrupt management module.

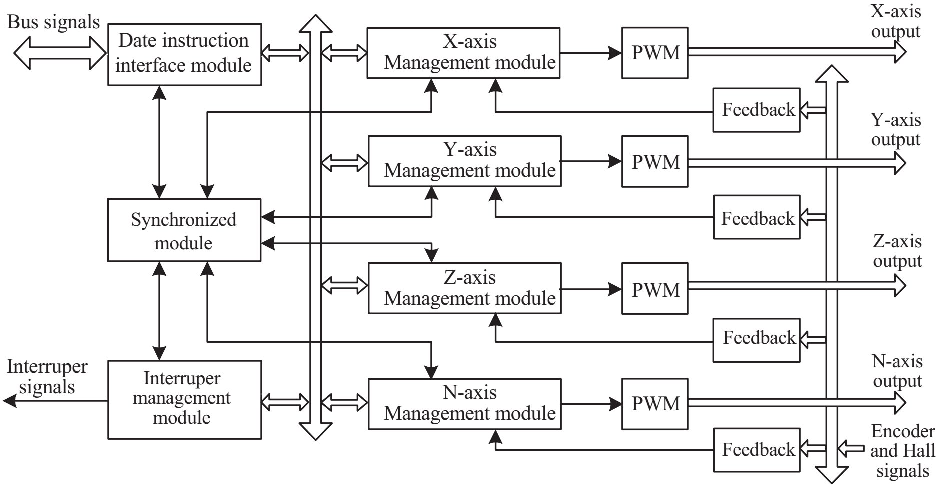


Fig. 12. The internal structure of motion controller

Low-level control algorithm is implemented as digital PID. The basic functions of the controller include data cache, control algorithms, signal feedback, and PWM generation. The controller is designed to control 18 DC servos motors in the robot joints.

6. Conclusions

The bionic cockroach robots have gone through a few generations over the past decades. However their motion versatility and sensing and navigation abilities are still far from their biological counterparts.

One key aspect of bionic cockroach robots is multi-domain fusion approach towards bionic mechanism design and motion smoothness control. It combines bionics, novel mechanism and disperse adaptive control theory to realize harmonious bionic motion. Parallel kinematic mechanism coupled with spring mechanism offers a realisable approach to emulate the functions of cockroach legs

For a robot to mimic the powerful sensing and navigation abilities of a cockroach, a multi-sensing fusion system has been proposed. It consists of binocular vision system based on infrared imaging, and tactical sensors using fibre optic sensors and position sensitive detectors.

The large amount of sensing information and real time motion control of 6-leg robots require careful consideration of control system architecture. This paper has conceptualised a distributed multi-CAN bus-mastering system based on Field Programmable Gate Array (FPGA) and Advanced RISC Machine (ARM) microprocessor. The system architecture provides stage treatment for control information and real-time servo control. It consists of there core modules: (i) nodes of CAN bus servo drive, (ii) distributed multi-CAN bus-mastering system based on FPGA, and (iii) software system based on ARM and Real-Time Application Interface (RTAI).

A new chip called "smart brain chip" based on FPGA and SOPC technology has been proposed to implement intelligent motion control for cockroach robot. It interfaces with devices including motors, encoders, hall sensors; and execute low-level control algorithm and smooth motion curve.

7. Acknowledgement

This work is supported by Natural Science Foundation of China under the research project 60775059, 863 Program of China under the research projects 2006AA04Z218, and 2008AA04Z210.

8. References

Karalarli, E.; Erkmen, A. M. & Erkmen, I. (2004). “Intelligent Gait Synthesizer for hexapod Walking Rescue Robots”, *Proc. of IEEE, Inter. Conf. on Robotics and Automation*, pp. 2177 – 2182.

Bai, S. P.; Low, K. H. & Guo, W. M. (2000). “Kinematographic Experiments on Leg Movements and Body Trajectories of Cockroach Walking on Different Terrain”, *Proc. of IEEE Inter. Conf. on Robotics and Automation*, San Francisco, USA, pp. 2605 – 2610.

Quinn, R. D. & Ritzmann, R. E. (1998). “Construction of a Hexapod Robot with Cockroach Kinematics Benefits both Robotics and Biology”, *Connection Science*, Vol. 10, No. 3, pp. 239 – 254, 1998

Quinn, R. D.; Nelson, G. M.; Bachmann, R. J.; Kingsley, D. A. et al. (2003). “Parallel Complementary Strategies For Implementing Biological Principles Into Mobile Robots”, *The Int. Journal of Robotics Research*, Vol. 22, No. 3, pp. 169 – 186.

Schroer, R. T.; Boggess, M. J.; Bachmann, R. J. & Quinn, R. D. et al. (2004). “Comparing Cockroach and Whegs Robot Body Motions”, Proceedings of the IEEE International Conference on Robotics & Automation, New Orleans, pp. 3288– 3293.

Quinn, R. D.; Kingsley, D. A.; Offi, J. T. & Ritzmann, R. E. (2004). “Automated Gait Adaptation for Legged Robots”, *IEEE Int. Conf. On Intelligent Robots and Systems*, Lausanne, Switzerland, pp.2652 – 2657.

Allen, T.J.; Quinn, R. D.; Bachmann, R. J. & Ritzmann, R.E. (2003). “Abstracted biological principles applied with reduced actuation improves mobility of legged robots”, *Proc. of IEEE/RSJ Inter. Conf. of Intelligent Robots and Systems*, Las Vegas, USA, pp.1370 –1375.

Saranli, U.; Buehler, M. & Koditschek, D. E. (2000). “Design, Modeling and Preliminary Control of a Compliant Hexapod Robot”, *Proc. of IEEE, Inter. Conf. on Robotics and Automation*, pp. 2589 – 2586.

Moore, E.Z.; Campbell, D.; Grimminger, F. & Buehler, M. (2002). “Reliable Stair Climbing in the Simple Hexapod 'RHex'”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp 2222-2227.

Komsuglu, H.; McMordiey, D.; Saranlix, U. & Moore N. et al. (2001). “Proprioception Based Behavioral Advances in a Hexapod Robot”, *Proc. of IEEE, Inter. Conf. on Robotics and Automation*, pp. 3650-3655.

Weingarten, J. D.; Lopes, G. A. D.; Buehler, M. & Groff, R. E. et al. (2004) “Automated Gait Adaptation for Legged Robots”, *Proc. of IEEE Inter. Conf. on Robotics and Automation*, New Orleans, pp. 2153-2158.

Wei, T. E.; Quinn, R. D. & Ritzmann, R. E. (2004). “A CLAWAR That Benefits From Abstracted Cockroach Locomotion Principles”, *Inter. Proc. of Climbing and Walking Robots Conference*.

Boggess, M. J.; Schroer, R. T.; Quinn, R. D. & Ritzmann, R. E. (2004). “Mechanized Cockroach Footpaths Enable Cockroach-like Mobility”, *Proc. of IEEE, Inter. Conf. on Robotics and Automation*, New Orleans, pp. 2871-2876.

Kornbluh, R.; Kornbluh, R.; Pei, Q. & Stanford, S. et al. (2002) “Dielectric elastomer artificial muscle actuators: toward biomimetic motion”, *Proc. SPIE*, Vol. 4695, pp 126-137.

Beer, R. D.; Quinn, R. D.; Chiel, H. J. & Ritzmann, R. E. (1997). “Biologically Inspired Approaches to Robotics”, *Communications of the ACM*, Vol. 40, No. 3, pp.31 – 38.

Espenschied, K S.; Quinn, R D.; Beer, R D. & Chiel, H. J. (1996). “Biologically based distributed control and locusl reflexes improve rough terrain locomotion in a hexapod robot”, *Robotics and Autonomous System*, Vol. 18, pp. 59-64.

6

Biologically-Inspired Design of Humanoids

Xie M., Xian L. B., Wang L. and Li J.
School of Mechanical & Aerospace Engineering, Nanyang Technological University
Singapore

1. Introduction

Human beings are the most advanced creatures in the nature because of the combined abilities of learning and performing both physical and mental activities. In terms of physical activities, a human being is very skilful in undertaking both manipulation and biped walking. And, in terms of mental activities, two impressive behaviours are analysis and synthesis. Because of the mental power of doing analysis and synthesis, it is unique for human beings to achieve discoveries and inventions. For instance, human beings have gained a better understanding of the nature through a series of important discoveries, which in turn fuel human being's creativity leading to inventions. As result of human-made inventions, our lives are much enjoyable than before.

Interestingly, among the human-made inventions, the most challenging one will be humanoid robots (Hirai et al, 1998). One reason is that a humanoid robot is a unique platform which integrates both complex manipulation (i.e. dexterous grasping and motions) and complex locomotion (i.e. bipedal locomotion) (Ishida et al, 2004). Another reason is that a humanoid robot is a unique platform which helps us to discover the physical principles behind human-like skills and human-like intelligence (Xie et al, 2004). In the past twenty years, we have seen many humanoid robot projects around the world. Among them, we can mention HRP (Kaneko et al, 1998), BIP2000 (Espiau et al, 2000), ASIMO (Sakagami et al, 2002), QRIO (Ishida et al, 2004), HOAP (Kurazume et al, 2005) and HUBO (Kim et al, 2005).

In this chapter, we will discuss the issues behind the blueprints of a humanoid robot's body, brain and mind. Also, we will show examples of solutions to these important issues, which are implemented on our LOCH humanoid robot.

2. Blueprint of Artificial Life

With the advance in mechanics, electronics, control, and information technology, it is natural for people to dream of creating artificial life, which could possess a sophisticated body, brain and mind (Xie, 2003). And, it is always the dream of human beings to create an artificial life called *robot*, which could help us to perform dirty, difficult, or even dangerous, jobs. Before we venture into the creating of artificial life, it is interesting to ask this question: *What is an artificial life?*

This is a difficult question. Only the designer of life is able to provide the full answer. However, from an engineering point of view, we can identify the key steps which evolve non-life into life.

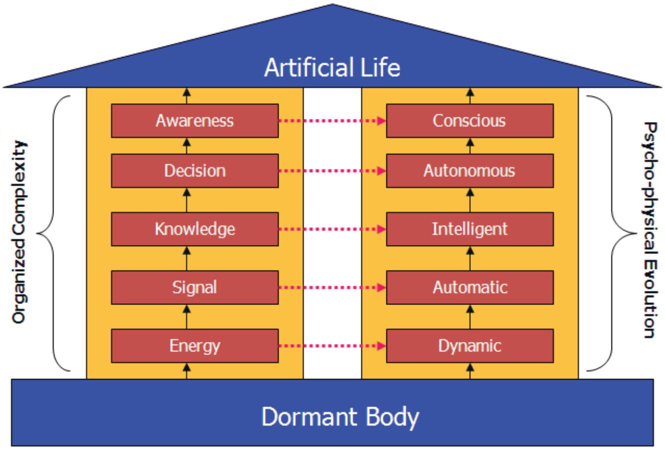


Fig. 1. Five steps leading to artificial life.

Refer to Figure1. We believe that the evolution from non-life to life will go through the following five key steps:

- Step 1: To be a dynamic system.
  When a dormant body could respond to energy, such a dormant body will become a dynamic system. In engineering, the use of actuators to drive a mechanism is a typical example of creating a dynamic system which could respond to electric energy.
- Step 2: To be an automatic system.
  When a dynamic system could respond to signal, such a dynamic system will become an automatic system. By default, a dynamic system has its own transient and steady-state responses when energy is applied to it as input. In order to control a dynamic system for the purpose of achieving intended responses, it is necessary to create a feedback mechanism so that a dynamic system will be able to directly respond to signals, which in turn control the release of the energy. Such a feedback mechanism can be called a *behavioural Mind* which plays the role of doing sensory-motor mapping.
- Step 3: To be an intelligent system.
  When an automatic system could respond to knowledge extracted from signals, such an automatic system will become an intelligent system. And, it is necessary to know the principles behind the design of a *cognitive Mind*, which will have the ability of extracting knowledge from signals such as visual or auditory signals.
- Step 4: To be an autonomous system.
  When an intelligent system has the innate ability of making its own decisions and acts according to its own decisions, such an intelligent system will become an autonomous system. Therefore, an autonomous system must have a *creative Mind* which is able to manipulate knowledge so as to synthesize decisions.
- Step 5: To be a conscious system.

Finally, when an autonomous system has a *conscious Mind* which is able to be aware of any consequence of doing (or being) and not-doing (or not-being), such an autonomous system will become a conscious system. When a dormant body reaches the level of being a conscious system, we can say that a life or artificial life is born.

These five steps are important to guide discoveries in science and inventions in engineering. For instance, the answer to the question of "what are the principles behind a human being's mind?" is yet to be discovered in science. And, the question of "how to create an artificial mind which could extract knowledge from signals?" is still a big challenge in engineering.

3. Blueprint of Humanoid Robot

3.1 Blueprint of Body

In general, a humanoid robot's body will consist of structure and mechanism (Kim et al, 2005). The purpose of structure is to house a humanoid robot's computational modules, communication modules, actuation modules and sensing modules. And, the main structure will be located at the trunk of a humanoid robot. On the other hand, the purpose of mechanism is to produce motions which in turn enable a humanoid robot to perform actions. And, a mechnism is much more complex than a structure.

Since a humanoid robot has a human-like body, the blueprint of a humanoid robot's body will cover the mechnism of neck, the mechanism of arm, the mechnism of hand, the mechanism of waist, the mechanism of leg and the mechanism of foot. And, the generic design requirements for a humanoid robot's body include:

1. The payload and degrees of freedom at the neck.
2. The payload and degrees of freedom at each arm.
3. The payload and degrees of freedom at each hand.
4. The payload and degrees of freedom at the waist.
5. The payload and degrees of freedom at each leg.
6. The payload and degrees of freedom at eah foot.

In order to enable a humanoid robot to perform human-like motions, the layout of degrees of freedom and the angle ranges of these degrees of freedom must closely follow the corresponding data of a human being. However, a human being's body is a bio-mechanic system, which has redundancy in kinematics. Therefore, it is necessary to do some simplification. For instance, a humanoid robot's neck may just have two degrees of freedom. And, a humanoid robot's waist is good enough to have two degrees of freedom as well.

In mechanics, a degree of freedom indicates an allowable motion between two rigid bodies along, or about, an axis. In order to specify the orientation of an axis, it is necessary to define a global reference coordinate system as shown in Figure 2.

Refer to Figure 2. The global reference coordinate system is placed on a ground. The Z axis is perpendicular to the ground. The rotation about Z axis is called *Yaw*. The Y axis is in the coronal plane of a humanoid robot. The rotation about Y is called *Pitch*. And, the X axis is the in sagittal plane of a humanoid robot. The rotation about X axis is called *Roll*.

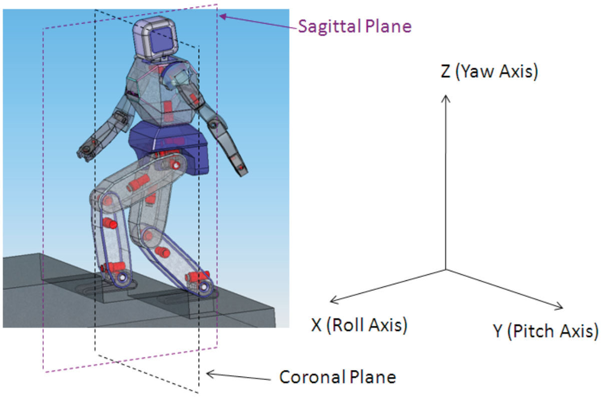


Fig. 2. Definition of reference coordinate system and (Yaw, Pitch, Roll).

In practice, a humanoid robot may have a distribution of degrees of freedom as follows:

a) Neck: Two degrees of freedom.
b) Arm: Six degrees of freedom so that the wrist can be in any orientation.
c) Hand: Ten degrees of freedom in total, and two degrees of freedom per finger.
d) Waist: Two degrees of freedom.
e) Leg: Six degrees of freedom so that the ankle can be in any orientation.
f) Foot: One degree of freedom.

Figure 3 shows a typical layout of the degrees of freedom in a humanoid robot.

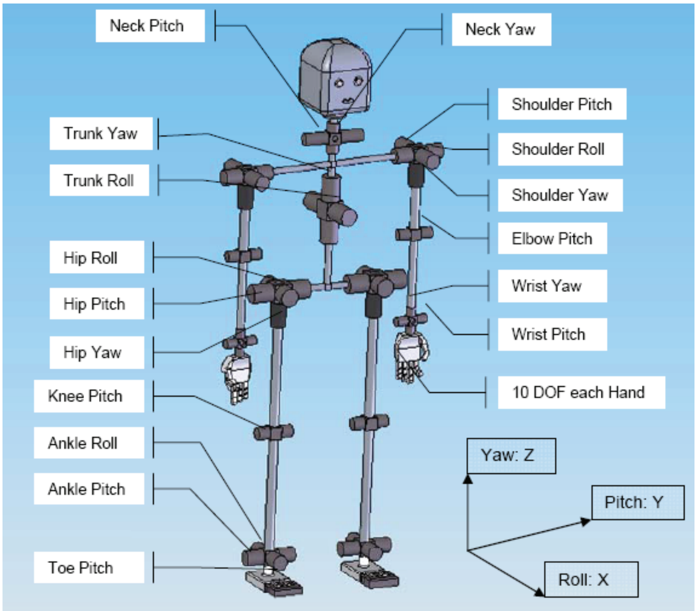


Fig. 3. Layout of degrees of freedom in a humanoid robot.

We can see that the orientations of these degrees of freedom are as follows:

1. Neck: (Yaw, Pitch)
2. Arm: (Pitch, Roll, Yaw) at shoulder; Pitch at elbow; (Pitch, Yaw) at wrist.
3. Hand: (Yaw, Pitch) at thumb; (Pitch, Pitch) at other fingers.
4. Waist: (Yaw, Roll)
5. Leg: (Pitch, Roll, Yaw) at hip; Pitch at knee; (Roll, Pitch) at ankle.
6. Foot: Pitch.

3.2 Blueprint of Brain

In a human being's brain, there are: a) cerebrum and b) cerebellum. The cooperation of these two organs makes a human being extremely powerful in undertaking: a) knowledge-centric activities and b) skill-centric activities. Interestedly, the knowledge-centric activities are orchestrated by the cerebrum. And, the neural system in the cerebrum is divided into different zones, each of which has a specific function such as speech, vision, reading, writing, smelling, reasoning, etc.

On the other hand, the skill-centric activities are controlled by both the cerebrum and the cerebellum. For instance, the cerebrum controls the skill-centric activities at the cognitive level, such as: planning, coordination, and cooperation. And, the cerebellum controls the skill-centric activities at the signal level with a network of feedback control loops, each of which consists of: a) sensing neurons, b) actuating neurons and c) control neurons.

In engineering terms, a human brain can be treated as a distributed system with two main controllers and many sub-controllers. Therefore, the blueprint of a humanoid robot could follow such a design, which is based on a network of distributed microcontrollers under the supervision of two main host computers, as shown in Figure4.

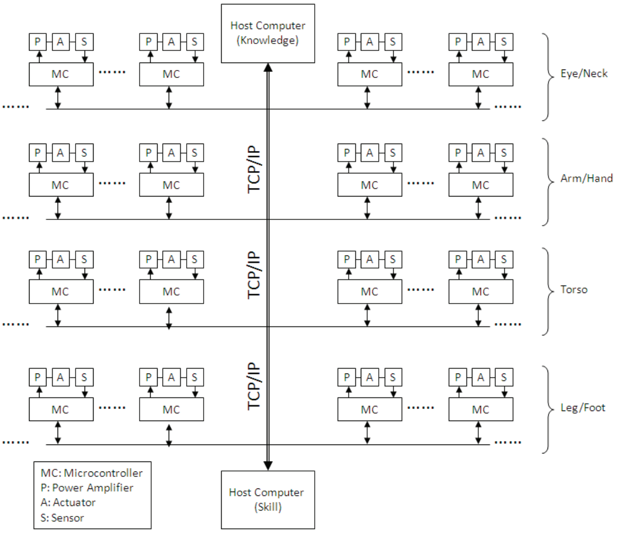


Fig. 4. A network of distributed microcontrollers and two main computers.

Refer to Figure4. Each microcontroller has the abilities to do: a) sensing, and b) control. And, each of the main computers will have the built modules for wired and wireless communications, which will enable a humanoid robot to act and interact with human beings or other humanoid robots.

3.3 Blueprint of Mind

A human being has a powerful mind, which enables him/her to perform both mentally and physically challenging activities. Also, it is interesting to note that a human being's mind is a composite mind which consists of: a) behavioral mind, b) cognitive mind, c) creative mind and d) conscious mind.

In engineering terms, a behavioral mind is responsible for the control and coordination of skill-centric activities such as grasping, manipulation, walking, and running, etc. And, the basic principle behind a behavioral mind is the feedback control mechanism.

For a humanoid robot, the coordinated control of the motions at the joints will give rise to a complex behavior. And, at each, there are two types of motion: a) unconstrained motions and b) constrained motions. Therefore, at each joint, there must be three feedback control loops such as a) position control loop, b) velocity control loop and c) torque control loops as shown in Figure 5.

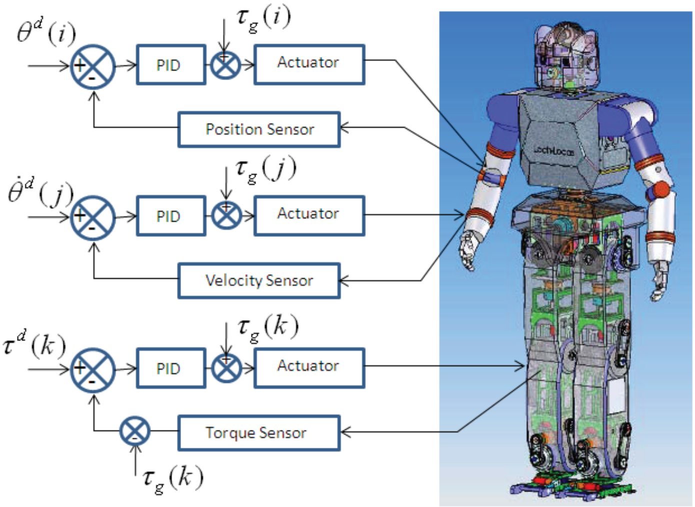


Fig. 5. Behavioral mind consisting of feedback control loops at the joints.

In Figure 5, $(\theta^d, \dot{\theta}^d, \tau^d)$ is a set of desired joint angle, desired joint velocity and desired joint torque. And, $\tau_g(i)$ is the torque for gravity compensation by joint i.

Due to the advance in control engineering, the principle behind a behavioural mind is well-understood. However, it is still a challenging to discover the principles behind a cognitive mind, a creative mind and a conscious mind.

It is worthy noting that there is no significant progress in artificial intelligence, despite the huge amount of research efforts devoted to this field and other related fields such as cognitive sciences, machine learning, and natural language processing. This stagnation is largely due to the lack of clear statements of the fundamental questions to be faced by artificial intelligence. People fail to clearly define such fundamental questions because the main focus of the actual research efforts is on the so-called *computer-aided human intelligence*, which is real and not artificial. And, the typical example of such endeavour is the IBM's deep blue project, in which the computer has no intelligence at all.

Here, we believe that artificial intelligence literally means the self-intelligence in robots or machines. Therefore, the fundamental questions to be faced by artificial intelligence include:

- What is the physical principle behind the transformation from visual signals into the cognitive state of knowing the meanings behind these signals? And, how to implement this physical principle onto a robot or machine?
- What is the physical principle behind the transformation from auditory signals into the cognitive state of knowing the meanings behind these signals? And, how to implement this principle onto a robot or machine?
- What is the blueprint behind the representation of learnt meanings? And, how to implement this blueprint onto a robot or machine?
- What are the processes and their working principles, which manipulate the learnt meanings for various activities related to knowledge and skill?

It is interesting to note that human mind is implemented onto a human brain, which has a neural architecture. In addition, a human mind is very powerful in undertaking both knowledge-centric activities and skill-centric activities. However, these two types of activities are very different. Then, we may ask this fundamental question: *What are the models behind a neural architecture, which share a same hardware infrastructure?*

In (Xie et al, 2004), we first show that the mathematical principle behind a neural network and hidden Markov model has the root on the state space equation, which describes the dynamic behaviours of any dynamic system. In other words, the model of a neural architecture for skill-centric activities is rooted on the *state space equation*.

Also, in (Xie et al, 2004), we outline a meaning centric framework for the representation of learnt meanings. We believe that a natural language is the best solution of knowledge representation. The evidence is our libraries, in which all learnt knowledge is documented in natural languages. Therefore, one crucial issue in artificial intelligence is: *what is the universal principle for a human mind to represent a natural language*?

Here, we advocate a concept-physical principle for the representation of a natural language, as shown in Figure 6, in which the main features are:

1. Meanings can be divided into two levels: a) the elementary meanings and b) the composite meanings.
2. A real world is composed of two related worlds, namely: a) physical world and b) conceptual world.
3. A physical world exists because of the existence of physical entities, which include nature-made objects and human-made objects.
4. A conceptual world exists because of the existence of conceptual entities, which include the words in natural languages.
5. The elementary meanings in the physical world refer to the properties and constraints of the entities in the physical world, while the elementary meanings in a conceptual world (note: each natural language depicts one conceptual world) refer to the properties and constraints of words in a conceptual world.
6. Each physical entity has at least one corresponding word in a conceptual world.

7. Each property of a physical entity has at least one corresponding word in a conceptual world.
8. Each constraint of a physical entity has at least one corresponding word in a conceptual world.
9. Interactions among the physical entities due to the constraints will create the composite meanings such as configurations, behaviours, events and episodes.
10. Interactions among the conceptual entities due to the constraints will create the composite meanings such as phases, sentences, concepts and topics.

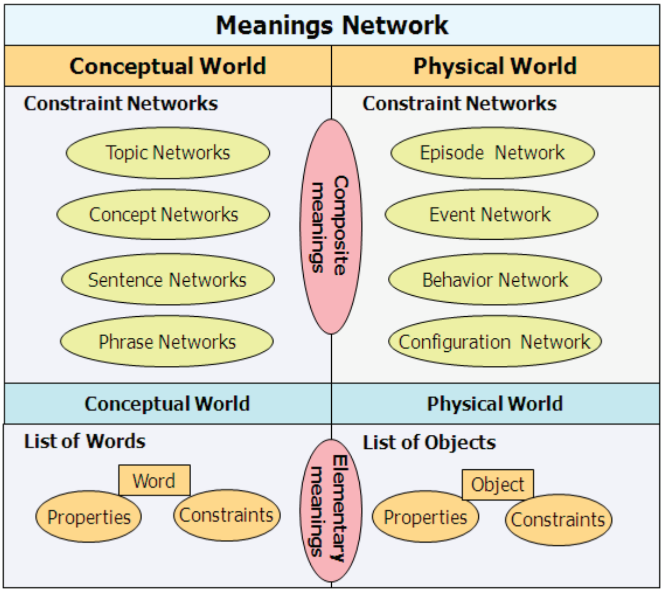


Fig. 6. Knowledge representation by meaning network.

In Figure 6, we have clearly defined what the knowledge (i.e. meanings) is. And, features 6, 7 and 8 solve the important problem of symbol grounding faced by the old paradigm for the study of natural language understanding.

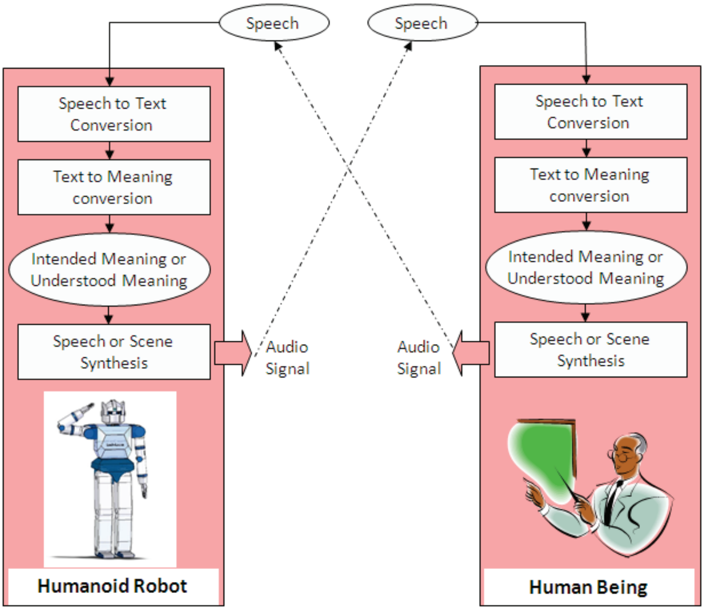

Speech
Speech
Speech to Text Conversion
Text to Meaning conversion
Intended Meaning or Understood Meaning
Speech or Scene Synthesis
Audio Signal
Audio Signal
Speech to Text Conversion
Text to Meaning conversion
Intended Meaning or Understood Meaning
Speech or Scene Synthesis
Humanoid Robot
Human Being

Fig. 7. Scheme of conversational dialogue with a natural language.

With the implementation of such a meaning network as shown in Figure 6, it becomes possible for robots to learn and converse in human language, insead of making human beings to continue to learn and program in machine language. Most importantly, the ability of undertaking conversational dialogues with a natural language, as shown in Figure 7, will be a decisive step for humanoid robots to be deployed into home enviroment for various services.

In summary, the blueprint of a human-like mind is two fold. First of all, the model for skill-centric activities is rooted on state space equation, which is effective in describing the dynamic response of any dynamic system. Second, the model for knowledge-centric activities is rooted on meaning network, which enables a range of mental processes to do jobs such as learning, analysis, synthesis, understanding, decision-making, and inference.

4. Design Considerations

When we implement the generic blueprints of body, brain and mind, we must consider several issues which may not be trivial.

4.1 Actuation by the Principle of Many-to-Many Coupling

Although it is relatively easy to choose materials and to size the motors during the detailed design of a humanoid robot's body, the issue of actuation is generally overlooked. One reason is because many researchers and engineers believe that the solution to actuate a joint is simply to use an actuator. However, careful analysis will revieal that there are different ways of actuating the joints in a body.

For instance, a human body's skeleton is actuated by musles. And, it is interesting to note that some joints are actuated by multiple muclses in order to achieve the redundancy in actuation. In this case, when one mucle faces a problem, a joint will not be handicaped in general. Such a scheme of coupling many muscles with many joints is useful in achieving reliable actuation. In engineering terms, the idea of coupling many actuators with many joints can be depicted in Figure 8.

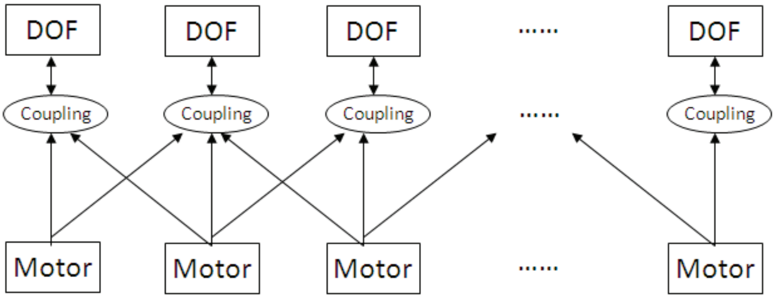


Fig. 8. Actuation scheme of coupling many actuators with many joints.

Another example is to use a single actuator to independently drive many joints in a robot (Zhu et al 2000; Xie, 2003). Such a scheme of coupling one actuator with many joints is useful if we want to reduce the weight and cost.

Unfortunately, today's robots are still follow the old scheme of coupling one actuator with one joint. This scheme has the least reliability. When one actuator fails, a joint will systematically fail. As a result, a robot will fall as shown in Figure 10, in which the knee joint of the robot's right leg fails during the climbing of staircase.

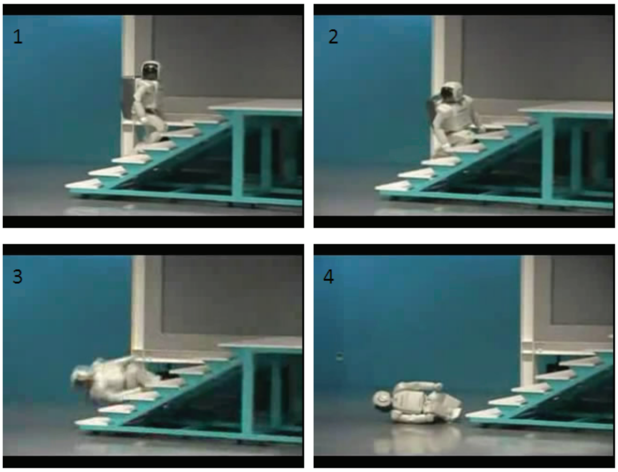


Fig. 9. Failure due to actuation scheme of coupling one actuator with one joint.

In engineering terms, the idea of coupling one actuator with one joint is simple to be implemented. Figure 10 illustrates the actuation scheme of coupling one actuator with one joint.

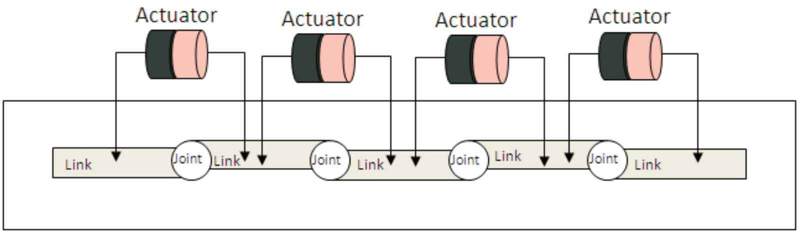


Fig. 10. Actuation scheme of coupling one actuator with one joint.

4.2 Foot with Variable Length

A human being's foot is not a single rigid body. It has at least one pitch joint (Bruneau, 2006). Interestingly, such a design enables a foot to have a variable foot length, which in turn will have two advantages.

First of all, a larger foot helps increase the stability of a humanoid robot during standing or walking. Second, a smaller foot (i.e. a shorter foot length) helps a humanoid robot to run/jump at ease. As shown in Figure 11, when a humanoid robot is lifting up the body, the lifting force comes from the torque at the ankle joint (or knee joint). From the relationship between force and torquem, it is clear that the shorter the foot length is, the larger the lifting force will be, when the torque remains the same.

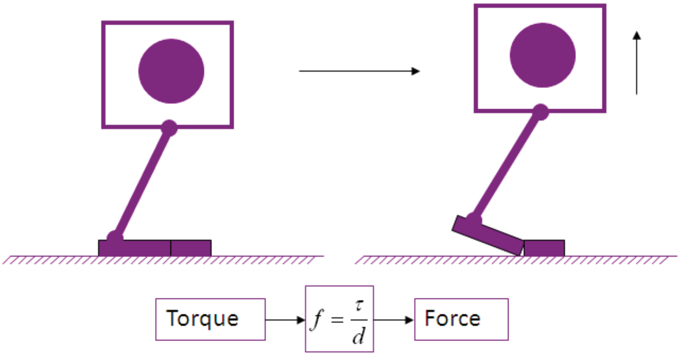


Fig. 11. A variable foot length helps generate larger lifting force.

Similarly, when a humanoid robot lands with a same impact force, the shorter the foot length is, the smaller the impact torque at the ankle joint will be.

In pratice, we can design a humanoid robot's foot with two rigid bodies. When a humanoid robot's foot consists of two links with the lengths of (l_1, l_2), the foot could have three possible configurations of length, such as: a) $l_1 + l_2$, b) l_1, and c) l_2.

4.3 Body with Massive Network of Sensors

A human being's body is not only agile in performing motions, but also sensible in capturing visual, auditory, kinesthetic, olfactory, taste, and thermal signals. Most importantly, a human being's body is a massive network of sensors. Such a massive sensing capability helps simplify the complexity of decison-making in undertaking appropriate actions in response to sensed signals.

Due to cost, today, it is still difficult to develop a humanoid robot which is as sensible as a human beings.

4.4 Behavioral Control

A human being can perform a wide range of manipulation tasks through the execution of motions by his/her arms and hands. Hence, it is clear that the motions at the joints of hands and arms are dictated by an intended task. In industrial robotics, it is well-understood that the inputs to the motion control loops at the joint level come from a decision-making process started with an intended task of manipulation. And, such a decision-making process includes:

- Behavior selection among the generic behaviors of manipulation as shown in Figure 12(a).
- Action selection among the generic actions of manipulation as shown in Figure 12(b).
- Motion description for a selected action.

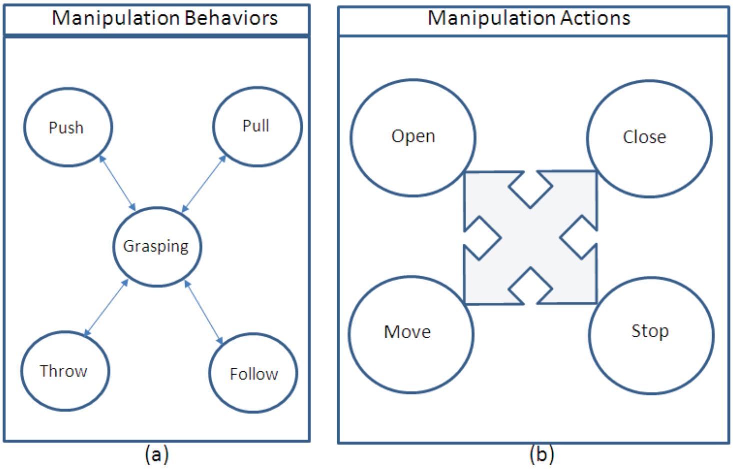


Fig. 12. Generic behaviours and actions for manipulation.

On the other hand, in the effort toward the design of planning and control algorithms for biped walking, not enough attention has been paid to this top-down approach of behavioral control. For instance, a lot of works is focused on the use of ZMP (i.e. zero-moment point) to generate, or control, dynamically stable gaits. Such stability-centric approaches do not answer the fundamental question of how to walk along any intended trajectory in real-time and in real environment. Because of the confusion on the relationship between cause and effect, one can hardly find a definite answer to the question of how to reliably plan and control a biped walking robot for any real application.

Here, we advocate the top-down approach to implement the behavioral control for biped locomotion. And, the inputs to the decision-making process for biped walking can be one, or a combination, of these causes:

- Locomotion task such as traveling from point A to point B along a walking surface.
- Self-intention such as speed-up, slow-down, u-turn, etc.
- Sensory-feedback such as collision, shock, impact, etc.

The presence of any one of the above causes will invoke an appropriate behavior and action (i.e. effect) to be undertaken by a humanoid robot's biped mechanism. And, the mapping from cause to effect will be done by a decision-making process, which will also include:

- Behavior selection among the generic behaviors of a biped mechanism as shown in Figure 13(a).
- Action selection among the generic actions of a leg shown in Figure 13(b).
- Motion description for a selected action.

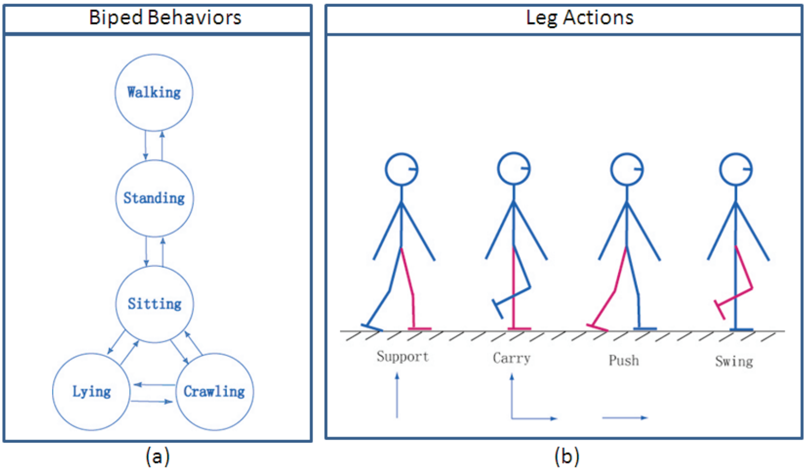


Fig. 13. Generic behaviours and actions for biped locomotion.

In order to show the importance of top-down approach for behavioral control, we would like to highlight the following correct sequence of specifying the parameters of walking:

- Step 1: To determine the hip's desired velocity from task, intention, or sensory feedback.
- Step 2: To determine the step length from the knowledge of the hip's desired velocity.
- Step 3: To determine the walking frequency (i.e. steps per unit of second) from the knowledge of the hip's desired velocity and the chosen step length.

In the above discussions, the motion description inside a behavioral control is to determine the desired values of joint positions, joint velocities, and/or joint torques, which will be the inputs to the automatic control loops at the joint level, as shown in Figure 14.

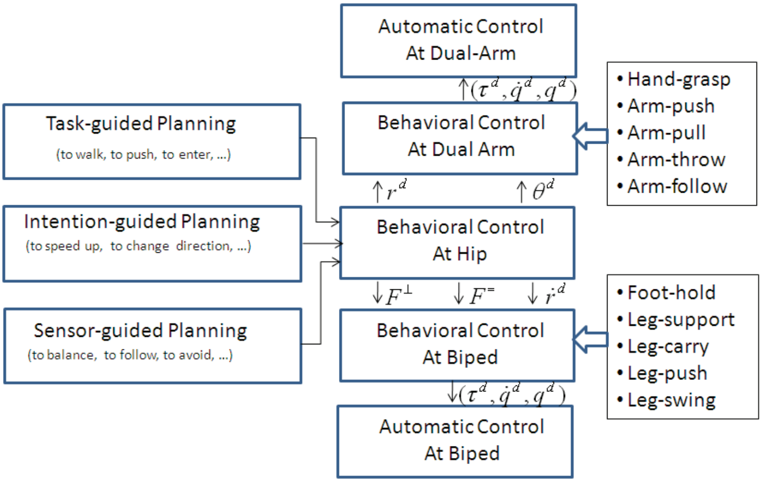

Automatic Control At Dual-Arm
$(\tau^d, \dot{q}^d, q^d)$
Task-guided Planning
(to walk, to push, to enter, ...)
Behavioral Control At Dual Arm
• Hand-grasp
• Arm-push
• Arm-pull
• Arm-throw
• Arm-follow
r^d
θ^d
Intention-guided Planning
(to speed up, to change direction, ...)
Behavioral Control At Hip
$F^\perp$
$F^=$
$\dot{r}^d$
Sensor-guided Planning
(to balance, to follow, to avoid, ...)
Behavioral Control At Biped
• Foot-hold
• Leg-support
• Leg-carry
• Leg-push
• Leg-swing
$(\tau^d, \dot{q}^d, q^d)$
Automatic Control At Biped

Fig. 14. Interface between behavioural control and automatic control.

4.5 Cognitive Vision

The behavioral mind of a humanoid robot will enable it to gain the awareness of its stability, and the awareness of its external disturbance. However, a human being is able to autonomously and adaptively perform both manipulation and location in a dynamically changing environment. Such an ability is quite unique due to a human being's vision which is intrinsically cognitive in nature.

In engineering terms, if we will design a humanoid robot with the innate ability of gaining the awareness of its workspace and/or walking terrain, it is necessary to discover the blueprint behind a cognitive vision and to implement such a blueprint onto a humanoid robot.

4.6 Cognitive Linguistics

Human beings can communicate effectively in using a natural language. And, the instructions to human beings can be conveyed in both written and spoken languages. In engineering terms, such a process of instructing a human being on what to do is very much similar to programming. But, this type of programming is at the level of a natural language. This is why it is called a linguistic programming. And, the purpose of linguistic programming is to make a human being to be aware of next tasks that he or she is going to perform.

Today, it is still a common practice for a human being to master a machine language in order to instruct a robot or machine on what to do. Clearly, this process of using machine language in order to communicate with robots has seriously undermined the emergence of humanoid robots in a home environment. In near future, it is necessary to design a humanoid robot which incorporates the blueprint of cognitive linguistics (yet to be discovered) so that it can gain the awareness of next tasks through the use of natural languages.

5. Implementations

5.1 Appearance and Inner Mechanisms

Our LOCH humanoid robot has the appearance and inner mechanisms as shown in Figure 15. And, the general specifications of the robot body are given in Table 1.

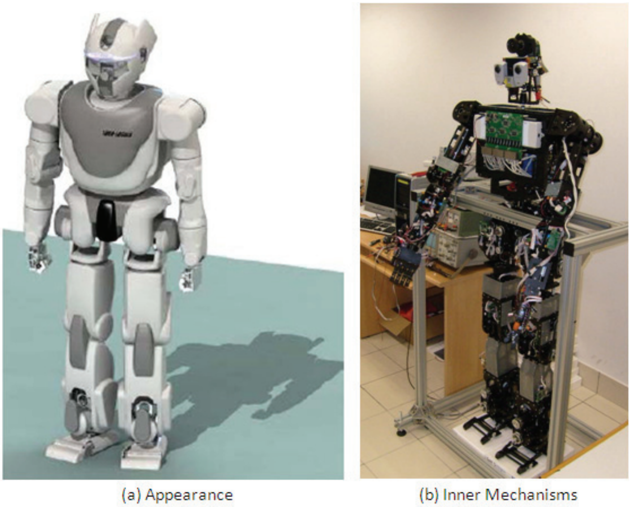

(a) Appearance
(b) Inner Mechanisms

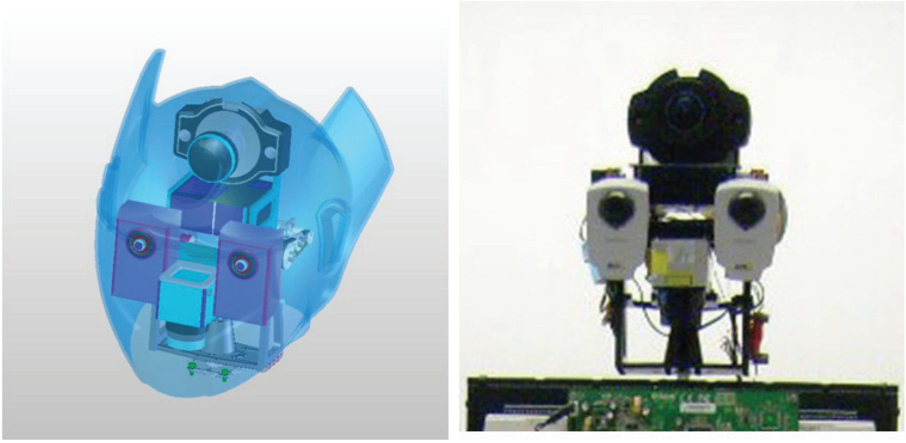
Fig. 15. LOCH humanoid robot: a) appearance and b) inner mechanisms.

Body weight:	80 kg
Body height:	1.75 m
Body width:	0.60 m
Body depth:	0.25 m

Table 1. Specifications of body.

5.2 Robot Head

The primary function of robot head is to sense the environment in which a humanoid robot is going to perform both manipulation and location. In our design, we have incorporated four types of environmental sensing capabilities, namely: a) monocular vision, b) stereovision, c) distance finder (up to 200 meters) and d) laser range finder (within 4 meters). Figure 16a shows the CAD drawing of the robot head, while the real prototype without external cover is shown in Figure 16b. And, the specifications of the robot head are listed in Table 2.



(a) CAD Model

(b) Actual Prototype

Fig. 16. Head of LOCH humanoid robot: a) CAD model and b) actual prototype.

Weight:	4 kg
Height:	22 cm
Width:	25 cm
Depth:	25 cm
Degrees of Freedom:	● Two DOFs at the neck (Yaw + Pitch)
Sensors	● One PTZ camera ● Two stereo cameras ● One distance finder ● One laser range finder ● Absolute encoder at each neck joint
Actuators:	● Two DC brush motors ● Two low-power amplifiers ● One micro-controller
Functions	● Visual perception ● Nod ● Gaze

Table 2. Specifications of Robot Head

5.3 Robot Trunk

The primary function of robot trunk is to house the host computers and power units. In addition, the robot trunk has two degrees of freedom which enable a humanoid robot to turn left and right, and also to swing left and right.

In Figure 17, we can see both the CAD model of the robot trunk and the real prototype of the robot trunk. And, the specifications of robot trunk are listed in Table 3.

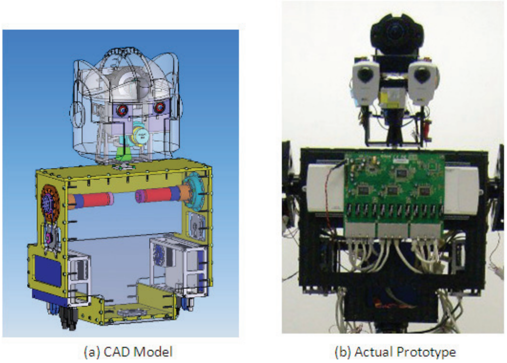


Fig. 17. Trunk of LOCH humanoid robot: a) CAD model and b) actual prototype.

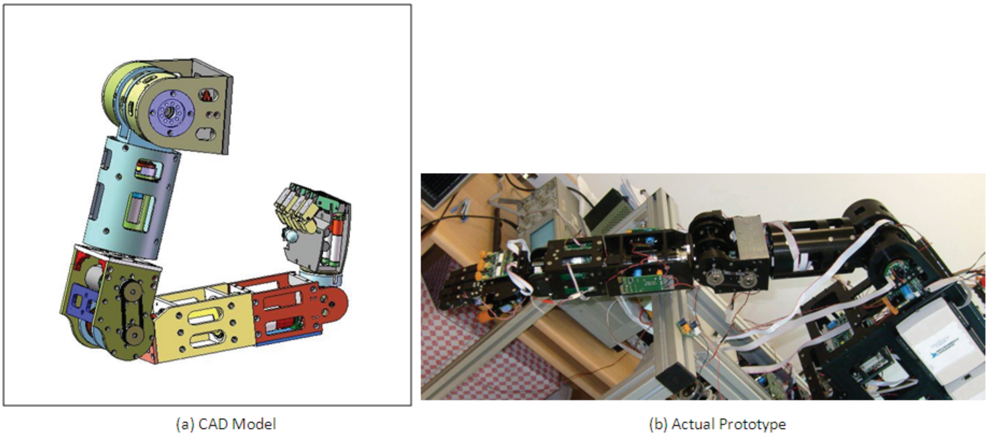
Height:	58 cm
Width:	40 cm
Depth:	20 cm
Weight:	24 kg
Computing Units:	● Two PC104 ● One wireless hub
Power Units:	● Capacity: 20 AH at 48 VDC ● Current: 20 A ● Voltage: 5V, 12V, 24V and 48V ● Weight: 15 kg
Degrees of Freedom:	● Two DOFs at the waist (Yaw + Roll)
Sensors:	● One 3-axis GYRO/Accelerometer ● Three microphones
Actuators	● Two DC brush motors ● Two low-power amplifiers ● One microcontroller
Functions	● Torso turn ● Torso swing

Table 3. Specifications of robot trunk.

5.4 Arms and Hands

Arms and hands are very important to a humanoid robot if it will perform human-like manipulation. And, the design of arms and hands should enable a humanoid robot to achieve these five generic manipulation behaviors: a) grasp, b) push, c) pull, d) follow and e) throw.

In Figure 18, we show both the CAD model and the real prototype of LOCH humanoid robot's arms and hands. We can see that LOCH humanoid robot has human-like hands, each of which has five fingers. And, the specifications of arms and hands are shown in Table 4.



(a) CAD Model

(b) Actual Prototype

Figure 18. Arms and hands of LOCH humanoid robot: a) CAD model and b) actual prototype.

Length:	● Upper arm: 32 cm ● Forearm: 28 cm ● Hand: 16 cm
Weight:	● Upper arm: 2.0 kg ● Forearm: 2.5 kg ● Hand: 1.8 kg
Degrees of Freedom:	● 3 DOFs in shoulder ● 1 DOF in elbow (Pitch) ● 2 DOFs in wrist (Pitch + Roll) ● 2DOFs in the thumbs ● 2 DOF in other fingers (one DOF is passive)
Sensors:	● 6-axis force/torque sensor at each wrist ● Absolute encoder at each arm joint ● Potentiometer at each hand joint ● Incremental encoder at each joint ● Pressure sensors at palm and fingers
Actuators:	● Six DC brush motors for each arm ● Six DC brush motors for each hand ● Six low-power amplifiers for each arm ● Six low-power amplifiers for each hand ● Three microcontrollers for each arm ● Three microcontrollers for each hand

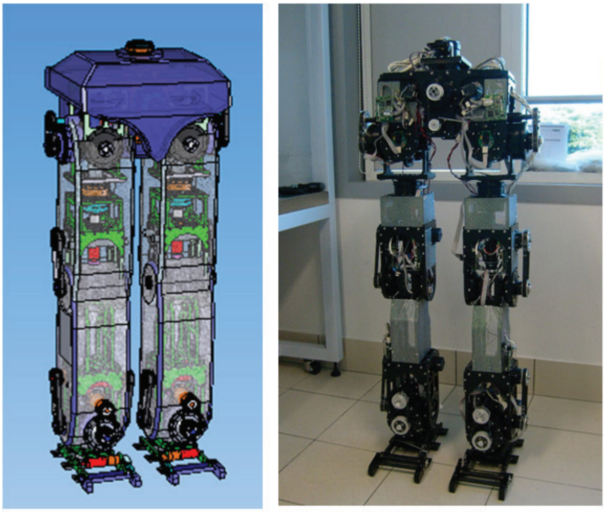
Functions:	● Grasp ● Pull ● Push ● Move ● Throw ● Hand-shaking ● Hand gesture ● Handling soft objects

Table 4. Specifications of robot arms and hands.

5.5 Legs and Feet

Legs and feet are unique features which differentiate a humanoid robot from an industrial robot. And, it is also very important to design legs and feet so that a humanoid robot could perform human-like biped walking/standing.

In Figure 19, we show both the CAD model and the real prototype of LOCH humanoid robot's legs and feet. It is worthy noting that LOCH humanoid robot has a ZMP joint in each joint, which is implemented by a six-axis force/torque sensor. This ZMP joint allows the control of the so-called in foot ZMP for leg stability (Xie et al, 2008). And, the specifications of arms and hands are shown in Table 5.



(a) CAD Model

(b) Actual Prototype

Figure 19. Legs and feet of LOCH humanoid robot: a) CAD model and b) actual prototype.

Length:	● Thigh: 42 cm ● Shank: 42 cm ● Foot: 31 cm
Weight:	● Thigh: 8.0 kg ● Shank: 6.0 kg ● Foot: 2.2 kg

Degrees of Freedom:	● 3 DOFs in each hip joint ● 1 DOF in each knee joint (Pitch) ● 2 DOFs in each ankle joint (Pitch + Roll) ● 1 DOF in each foot
Sensors:	● 6-axis force/torque sensor below each ankle joint ● Absolute encoder at each joint ● Incremental encoder at each joint ● Six pressure sensors below each foot
Actuators:	● Five DC brushless motors for each leg ● One DC brush motor for hip yaw ● One DC brush motor for each foot ● Five high-power amplifiers for each leg ● One low-power amplifier for hip yaw ● One low-power amplifier for each foot ● Four microcontrollers for each leg/foot
Functions:	● Foot-hold ● Leg support ● Leg carry ● Leg push ● Leg swing ● Standing ● Sitting ● Stepping ● Walking ● Running ● Climbing ● Crawling ● Entering/exiting car

Table 5. Specifications of robot legs and feet.

6. Discussions

A good design will enable a sophisticated analysis, control and programming of a humanoid robot.

6.1 Kinematics

In terms of analysis, two important aspects are kinematics and dynamics. As a humanoid robot can be treated as an open kinematic chain with bifurcation, the tools for analysing industrial arm manipulator are applicable to model the kinematics of a humanoid robot (Xie, 2003).

However, one unique feature with a humanoid robot is that there is no fixed base link for kinematic modelling. Therefore, an interesting idea is to describe the kinematics of a humanoid robot with a matrix of Jacobian matrices. For instance, if a humanoid robot has N coordinate systems assigned to N movable links, a NxN matrix of Jacobian matrices is sufficient enough to fully describe the kinematic property of a humanoid robot. And, in Figure20, J_{ij} refers to the Jacobian matrix from link i to link j.

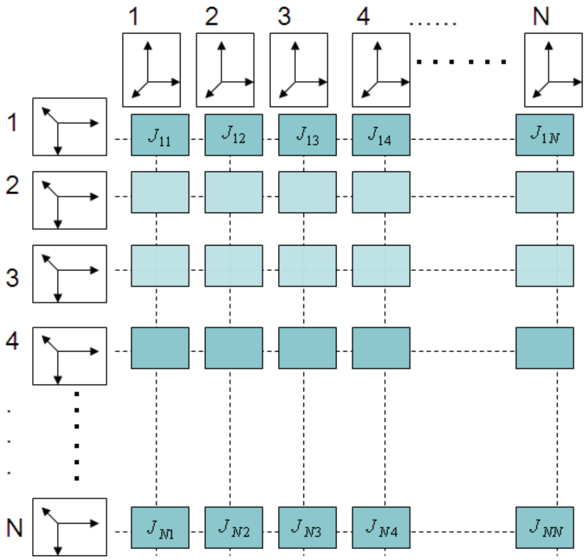
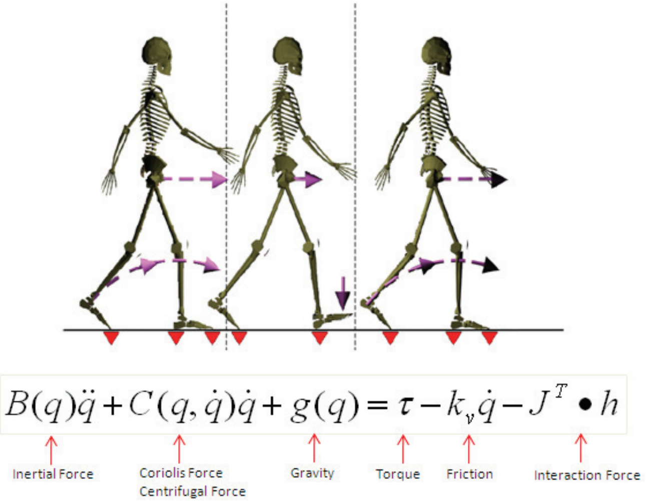


Fig. 20. A matrix of Jacobian matrices to describe the kinematics of a humanoid robot.

6.2 Dynamics

Given an open kinematic chain, the dynamic behaviour can be described by the general form of differential equation as shown in Figure21.

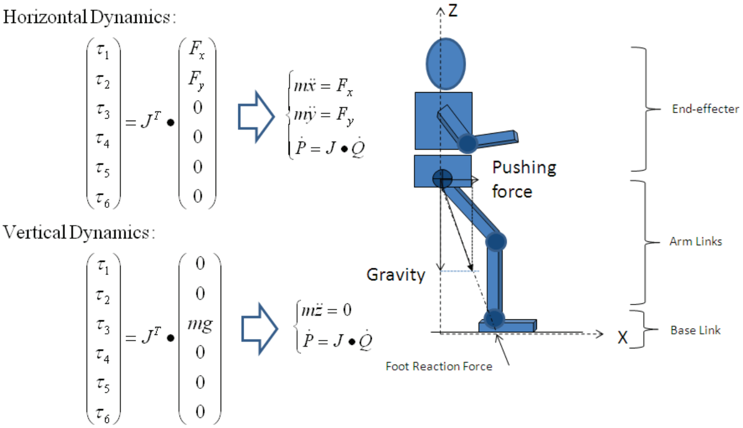
However, biped walking is not similar to manipulation. As a result, a common approach is to simplify a biped mechanism into a model called *linear inverted pendulum*. And, a better way to understand inverted pendulum model is the illustration by the so-called cart-table model (Kajita et al, 2003).



$$B(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=\tau-k_v\dot{q}-J^T\bullet h$$

Fig. 21. General dynamic equation of an open kinematic chain.

Here, we believe that we can treat a leg as an inverted arm with the foot to serve as the base link. In this case, the leg supporting the upper body of a humanoid robot is undergoing a constrained motion. And, it has both horizontal and vertical dynamics as shown Figure22.



Horizontal Dynamics:

$$\begin{pmatrix}\tau_1\\\tau_2\\\tau_3\\\tau_4\\\tau_5\\\tau_6\end{pmatrix}=J^T\bullet\begin{pmatrix}F_x\\F_y\\0\\0\\0\\0\end{pmatrix}\Rightarrow\begin{cases}m\ddot{x}=F_x\\m\ddot{y}=F_y\\\dot{P}=J\bullet\dot{Q}\end{cases}$$

Vertical Dynamics:

$$\begin{pmatrix}\tau_1\\\tau_2\\\tau_3\\\tau_4\\\tau_5\\\tau_6\end{pmatrix}=J^T\bullet\begin{pmatrix}0\\0\\mg\\0\\0\\0\end{pmatrix}\Rightarrow\begin{cases}m\ddot{z}=0\\\dot{P}=J\bullet\dot{Q}\end{cases}$$

Fig. 22. Inverted arm model to describe the dynamics of a biped mechanism.

In Figure22, we assume that a leg has a six degrees of freedom. J is the Jacobian matrix of a leg. $\dot{P}$ is the hip's velocity vector and $\dot{Q}$ is the vector of the joint velocities of the leg. And, m is the mass of a humanoid robot's upper body.

6.3 Human-Aided Control

Today's robots still have limited capabilities in gaining the situated awareness through visual perception and in making meaningful decisions. Therefore, it is always useful to design a humanoid robot in such way that a human operator can assist a humanoid robot to perform complex behaviours of manipulation and/or biped walking.
Therefore, it is interesting to implement virtual versions of a real humanoid robot, which serve as the intermediate between a human operator and a real humanoid robot, as shown in Figure23.

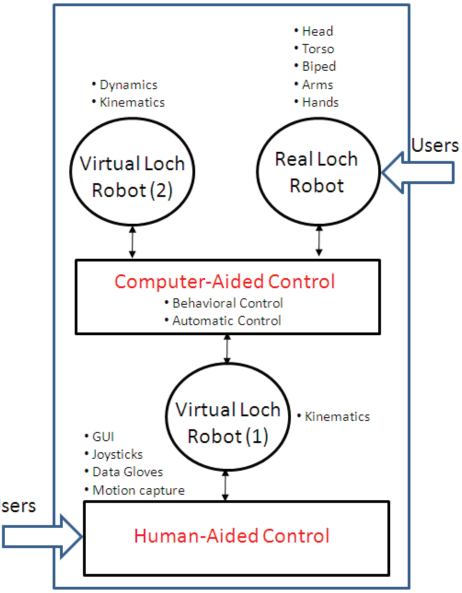


Fig. 23. Human-aided control through the use of virtual robots.

Refer to Figure 23. A human operator could teach a virtual robot to perform some intended tasks. Once a virtual robot has mastered the skill of performing a task, it will instruct the real robot to perform the same task through synchronized playback. On the other hand, a virtual robot could also play the role of relaying the sensory data of a real humanoid robot back to a human operator so that he/she will feel the sensation of interaction between a humanoid robot and its working environment.

7. Summary

In this chapter, we have first highlighted some characteristics observed from human abilities in performing both knowledge-centric activities and skill-centric activities. Then, we apply the observations related to a human being's body, brain and mind to guide the design of a humanoid robot's body, brain and mind. After the discussions of some important considerations of design, we show the results obtained during the process of designing our LOCH humanoid robot. We hope that these results will be inspiring to others.

8. Acknowledgements

The authors would like to thank the project sponsor. In particular, the guidance and advices from Lim Kian Guan, Cheng Wee Kiang, Ngiam Li Lian and New Ai Peng are greatly appreciated. Also, we would like to thank Yu Haoyong, Sin Mong Leng and Guo Yongqiang for technical support. Supports and assistances from Zhong Zhaowei, Yang Hejin, Song Chengsen and Zhang Li are gratefully acknowledged.

9. References

Xie, M.; Zhong, Z. W.; Zhang, L.; Xian, L. B.; Wang, L.; Yang, H. J.; Song, C. S. & Li, J. (2008). A Deterministic Way of Planning and Controlling Biped Walking of LOCH Humanoid Robot. *International Conference on Climbing and Walking Robots.*

Xie, M.; Dubowsky, S.; Fontaine, J. G.; Tokhi, O. M. & Virk, G. (Eds). (2007). *Advances in Climbing and Walking Robots*, World Scientific.

Bruneau, O. (2006). An Approach to the Design of Walking Humanoid Robots with Different Leg Mechanisms or Flexible Feet and Using Dynamic Gaits. *Journal of Vibration and Control*, Vol. 12, No. 12.

Kim, J.; Park, I.; Lee, J.; Kim, M.; Cho, B. & Oh, J. (2005). System Design and Dynamic Walking of Humanoid Robot KHR-2. *IEEE International Conference on Robotics and Automation.*

Ishida, T. (2004). Development of a Small Biped Entertainment Robot QRIO. *International Symposium on Micro-Nanomechatronics and Human Science*, pp23-28

Xie, M.; Kandhasamy, J. & Chia, H. F. (2004). Meaning Centric Framework for Natural Text/Scene Understanding by Robots*, International Journal of Humanoid Robotics*, Vol. 1, No. 2, pp375-407.

Xie, M. (2003). *Fundamentals of Robotics : Linking Perception to Action*. World Scientific.

Kajita, S. ; Kanehiro, F. ; Kaneko, K. ; Fujiwara, K. ; Harada, K. ; Yokoi, K. & Hirukawa, H. (2003). Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point, *IEEE International Conference on Robotics and Automation.*

Sakagami, Y. ; Watanabe, R. ; Aoyama, R. ; Matsunaga, C. ; Higaki, S. & Fujimura, K. (2002). The Intelligent ASIMO : System Overview and Integration. *IEEE International Conference on Intelligent Robots and Systems*, pp2478-2483.

Espiau, B. & Sardain, P. (2000). The anthropomorphic Biped Robot BIPED2000. *IEEE International Conference on Robotics and Automation, pp3996-4001*.

Hirai, K. ; Hirose, M. ; Hikawa, Y. & Takanaka, T. (1998). The Development of Honda Humanoid Robot. *IEEE International Conference on Robotics and Automation.*

Zhu, H. H. ; Xie , M. & Lim, M. K. (2000). Modular Robot Manipulator Apparatus. *PCT Patent Application*, No. PCT/SG00/00002.

Kaneko, K. ; Kanehiro, F. ; Yokoyama, S. ; Akachi, K. ; Kawasaki, K. ; Ota, T. & Isozumi, T. (1998). Design of Prototype Humanoid Robotics Platform for HRP. *IEEE International Conference on Intelligent Robots and Systems*, pp2431-2436.

The State-of-Art of Underwater Vehicles – Theories and Applications

W.H. Wang[1], R.C. Engelaar[2], X.Q. Chen[1] & J.G. Chase[1]
[1] *University of Canterbury, New Zealand*
[2] *University of Technology, Eindhoven, Netherlands*

1. Introduction

An autonomous underwater vehicle (AUV) is an underwater system that contains its own power and is controlled by an onboard computer. Although many names are given to these vehicles, such as remotely operated vehicles (ROVs), unmanned underwater vehicles (UUVs), submersible devices, or remote controlled submarines, to name just a few, the fundamental task for these devices is fairly well defined: The vehicle is able to follow a predefined trajectory.

AUVs offer many advantages for performing difficult tasks submerged in water. The main advantage of an AUV is that is does not need a human operator. Therefore it is less expensive than a human operated vehicle and is capable of doing operations that are too dangerous for a person. They operate in conditions and perform task that humans are not able to do efficiently or at all (Smallwood & Whitcomb, 2004; Horgan & Toal, 2006; Caccia, 2006).

First developed in the 1960's, development was driven by the demand from the US Navy (Wernli, 2001), which required them to perform deep sea rescue and salvage operations. In the 1970s, universities, institutes and governmental organizations started with the experimentation with AUV technology. Some of them were successful, most were not. Despite this, there was significant advancement in the development of AUVs. Since then, other sectors have realized the potential of such devices for all manner of tasks. The first of these was the oil and gas industry. These companies employed AUVs to reinforce in the development of off shore oil fields (Williams, 2004). In the 1980's, AUVs came into a new era as they were able to operate at depths well below commercial diver limits. Falling oil prices and a global recession resulted in a stagnant period in terms of AUV development in the mid 1980s. During the 1990s there was a renewed interest in AUVs in academic research. Many universities developed AUVs. This research was followed by the first commercial AUVs in 2000 (van Alt, 2000; Blidberg, 2001). Since then, AUVs have been developing at a fast rate (Smallwood et al., 1999; Griffiths & Edwards, 2003).

AUVs are now being used in a wide range of applications, such as locating historic ship wrecks like the Titanic (Ballard, 1987), mapping the sea floor (Tivey et al., 1998). More mundane applications consist of object detection (Kondoa & Ura, 2004), securing harbours, searching for seamines (Willcox et al., 2004), and, most recently, in scientific applications

(Curtin & Bellingham, 2001; Rife & Rock, 2002; Lygouras et al., 1998). In the past few years, advances in battery design and manufacture have led to batteries with high power densities, which have significantly increased the endurance of AUVs (Wilson & Bales, 2006). At the same time, the development of new technologies made the AUVs more accurate.

This book chapter aims to highlight theories and applications of technologies that are suitable for AUVs by literature review and a detailed AUV design. The chapter is therefore organized as follows. Firstly, Sections 2-6 provide an overview of the latest developments of five different subjects of an AUV, including: 1) the structure of AUVs; 2) controls and navigation; 3) propulsion, drive and buoyancy; 4) sensors and instrumentation of AUVs; and 5) power supply of AUVs. Next, Section 7 reports a relatively low-cost AUV recently developed at the University of Canterbury for shallow waters. Finally, future work and conclusions are given in Section 8.

2. Structure of AUVs

One of the most important aspects of an AUV is the hull. There are a number of different ways in which hull design can be approached (Allmendinger, 1990). These different design methods are typically specific to the situation/task. The main hull must be able to meet a number of key challenges.

Aspects that must be considered during hull design include:

- Pressure and/or depth required
- Operating temperature ranges
- Structural integrity for additions and tapings
- Impact conditions
- Water permeability
- Visual appeal and aesthetics
- Accessibility
- Versatility
- Practicality
- Restrictions for future additions
- Size requirements
- Corrosion and chemical resistance

Among these considerations, the hull of the AUV must be able to withstand the hydrostatic pressure at the target depth. Furthermore, it is desired that the hull is designed in such a way that the drag is minimized. When the vehicle moves at a constant speed, the thrust force is equal to the drag force. The less drag the AUV experiences, the less propulsive power is needed. These two requirements, the ability to withstand the hydrostatic pressure and the minimization of the drag, are dependent of the shape and size of the vehicle. The hydrostatic pressure is given by Equation (1).

$$P = P_a + \rho g h \tag{1}$$

With P the hydrostatic pressure in N/m^2, P_a the atmospheric pressure at sealevel in N/m^2, ρ the density of the water in kg/m^3, g the gravitational acceleration in m/s^2 and h the depth in m. The hydrostatic pressure increases with approximately $10^5 N/m^2$ per 10 meters. The hull must be able to withstand this force. A sphere is probably the first shape that comes into ones head, it is a good shape for withstanding pressure, but not for stability (Paster, 1986). A circular cylindrical hull is a good shape to resist the pressure (Ross, 2006). Many of the current AUVs have a circular cylindrical hull including the most popular in military and scientific use, the REMUS100 (Hsu et al., 2005; Evans & Meyer, 2004; Maurya et al., 2007). Some examples are shown in Figures 1. to 3.

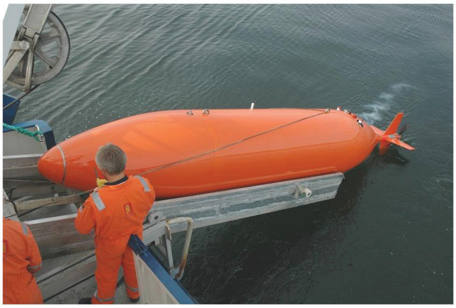


Fig. 1. The HUGIN 4500 autonomous underwater vehicle during deployment for sea trials (Kauske et al., 2007)

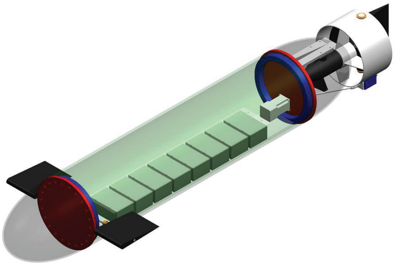


Fig. 2. The Cal Poly AUV model (Monteen et al., 2000)

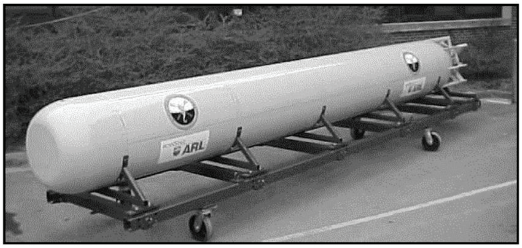


Fig. 3. The Seahorse AUV (Tangirala & Dzielski, 2007)

Some of the advantages of a cylindrical hull are (Ross, 2006):

- It is a good structure to resist the effects of hydrostatic pressure;
- Extra space inside the hull can be achieved by making the cylinder longer;
- It is a better hydrodynamic form than a spherical form of the same volume; and
- It can be easily docked.

The disadvantages of a cylindrical hull are the cavitation (Paster, 1986), and the instability of the vehicle (Ross, 2006). Cavitation is a phenomenon caused by the pressure distribution generated by the moving vehicle. The difference in local velocity of the body surface results in a pressure distribution. The point that has the maximum rate of change in curvature of the body has the negative minimum pressure. If this pressure reaches the vapor pressure of water, the water will start to boil. The bubbles formed by this boiling collapse when they reach the point where the pressure increases again. The collapse of the bubbles generates very high pressure. This leads to high noise levels and the possibility of damaging the vehicle (Paster, 1986).

Every object that moves in the water experiences drag force. This drag force (in Newton) is given by Equation (2).

$$F_{drag} = 1/2 \cdot \rho \cdot v^2 \cdot c_d \cdot S \tag{2}$$

With ρ the density of water [kg/m^3], v the velocity of the vehicle in m/s, c_d the unitless drag coefficient of the vehicle and S the surface area of the vehicle normal to the moving direction in m^2. The drag coefficient is dependent on the shape of the underwater vehicle. The nose of the circular cylinder used to be spherical, but this caused instability and cavitation (Paster, 1986). The shape of the nose was finetuned to resemble the front of a teardrop (Paster, 1986). A good hydrodynamic body shape design will reduce the drag and improves the range of the vehicle by 2 to 10 times, according to (Paster, 1986).

Another choice that needs to be made in the design phase is the choice for the material of the hull. The material should have a good resistance to corrosion, have a high strength to weight ratio and must be affordable. In the past, the most used material was steel. In (Ross, 2006) four materials are compared; high strength steel, aluminium, titanium and composites.

The advantages of high strength steel are the price and the fact that it is commonly used, so there is much knowledge of it. The major disadvantage of steel is the low strength to weight ratio.

Aluminium has a better strength to weight ratio than steel and is widely available. The drawback of aluminium is that it is anodic to most other structural alloys, making it vulnerable to corrosion. The strength to weight ratio of titanium is even better than that of aluminium, but it is an expensive material.

The most commonly used composite for marine vehicles is glass-fiber reinforced plastic (GFRP). GFRP is cheap with respect to other composites and has a very high strength to weight ratio. Carbon fiber reinforce composites (CFRP) are about 3 times more expensive than GFRP, but have a much higher tensile modulus than GFRP. Metal matrix composites (MMC) have a lot of advantages over GFRP and CFRP but are still in the development phase, making them very expensive (about 15 times more expensive than GFRP).

Another material that can be used for AUVs is acrylic plastic. Acrylic is already the most used material for pressure resistant viewports (Stachiw, 2004). The main advantages of acrylic are that it does not corrode and has a good strength to weight ratio. Furthermore, acrylic is transparent and there are acrylic submersibles that operate at depths up to 1000 meters below the surface.

AUVs that have an operating depth of tens of meters can also be constructed of PVC. The material is widely available and very cheap. With a hull made of PVC it is also easy to mount components on it (Monteen et al., 2000). Table 1 summarizes the properties of each material discussed. The specific strength is given by the ratio of the yield strength and the density.

Material	Density (kg/dm^3)	Yield strength (MPa)	Tensile modulus (GPa)	Specific strength (kNm/kg)
High strength Steel (HY80)	7.86	550	207	70
Aluminium alloy (7075-6)	2.9	503	70	173
Titanium alloy (6-4 STOA)	4.5	830	120	184
GFRP (Epoxy/S-lass)	2.1	1200	65	571
CFRP (Epoxy/HS)	1.7	1200	210	706
MMC (6061 Al/SiC)	2.7	3000	140	1111
Acrylic	1.2	103	3.1	86
PVC	1.4	48	35	34

Table 1. Material properties, from (Ross, 2006) and (Stachiw, 2004)

According to numerous authors the cylindrical shape is a very good one for an AUV. In the future MMC may be the best choice for the material, but until then GFRP is a good choice for AUVs. If the operating depth is only tens of meters, PVC is a good and cheap alternative.

3. Controls and navigation

An AUV must be able to operate autonomously. In order to achieve this it is essential that the computer in the AUV knows its current location at all time. This can be done by means of an accurate navigation system. Another reason that the AUV has got to know its location is because of the fact that gathered data is pretty much useless if the location from which it has been acquired is unknown (Leonard et al., 1998). To navigate properly a good, accurate controller is necessary for which first a mathematical model of the AUV is needed. The basic model for the AUV is described in Equation (3).

$$\begin{aligned} \mathbf{M}\dot{v} + \mathbf{C}(v)v + \mathbf{D}(v)v + \mathbf{g}(\eta) &= \tau \\ \dot{\eta} &= \mathbf{J}(\eta)v \end{aligned} \tag{3}$$

Where **M** is the inertia matrices for rigid body and added mass, $\eta = [x, y, z, \phi, \theta, \psi]^T$ the position and orientation (Euler angles) in inertial frame, $v = [u, v, w, p, q, r]^T$ the linear and angular velocities in body-fixed frame, **C** is the Coriolis matrix for rigid body and added mass, **D** is the quadratic and linear drag matrix, **g** is the buoyancy and gravity forces, τ is the thruster input vector and **J** is the coordinate matrix which brings the inertial frame into alignment with the body-fixed frame. The model is described in detail in (Fossen, 1994).

When a model is made for the vehicle it is possible to design a controller. For low speed vehicles, the horizontal and vertical movements can be decoupled, which makes the model of the vehicle less complex (Maurya et al., 2007; Williams et al., 2006; Ridao et al., 2001). Because a single fixed linear controller is not sufficient to deal with all the vehicle dynamics, a gain-scheduled controller is often used (Kaminer et al., 1995). First, a number of controllers are designed for a finite number of linearized models using H_∞-control. H_∞-control rests on a good theoretical basis and offers clear guidelines to achieve robust performance in the case of uncertainties in the plant (Zeng & Allen, 2004; Fryxell et al., 1996). These controllers are then combined using gain-schedule on some variables, making the overall controller a linear time-varying system. In multiple articles (e.g. Zeng & Allen, 2004; Jalving, 1994) three different controllers are designed. One is used to control the speed, the other for the heading and the last one for the depth.

In (Valavanis et al., 1997) four different architectures for control are described. The hierarchical architecture is a top-down approach which uses levels. The higher levels are responsible for overall mission goals, while the lower levels solve particular problems. It has a serial structure, which means that the higher levels send commands to lower levels. It is a well-defined structure, but has a lack of flexibility. The heterarchical architecture is a parallel structure. It has flexibility and is suitable for parallel processing. However, due to lack of supervision the communication can be intensive. With the subsumption architecture the different behaviors work in parallel, however one layer can subsume another layer. This architecture is robust and exhibits true dynamic reactive behavior. The disadvantage is the difficulty to synchronize the system.

Finally, the hybrid architecture is a combination of the three architectures. It is divided into two levels. The higher level uses hierarchical architecture while the lower uses either heterarchical or subsumption to control the hardware. It combines the advantages of the three architectures and is used in many vehicles (Williams et al., 2006; Valavanis et al., 2997; Gaccia & Veruggio, 2000). (Gaccia & Veruggio, 2000) makes use of an inner and an outer control loop. The inner loop is used for the control of the velocity and the outer loop is used for guidance control and to set the reference velocities.

When a controller is designed for the vehicle, the navigation system can be implemented. According to (Stutters et al., 2008) the accuracy of position estimation will degrade over time if the position of the AUV is not externally referenced. The lack of easy observable external references makes AUV navigation very difficult.

Leonard et al. (1998) describes the three primary methods of navigation, dead-reckoning and inertial navigation systems, acoustic navigation and geophysical navigation techniques. The sensors and instrumentation used for the measurement of the different variables are described in Section 5. Dead-reckoning integrates the vehicle velocity in time to obtain the position. The information is then processed by a Kalman filter which gives an estimate of the current position. The velocity measurement can be affected by sea currents, so operations near the seabed use Doppler velocity logs (DVL, see Section 5) to measure the velocity with respect to the ground. Inertial navigation uses the acceleration of the vehicle and integrates this twice. This is more accurate then the velocity measurement, but the initialization can be difficult (Lee et al., 2007).

The problem with both systems is that the position error increases as the distance traveled increases. This can be solved by surfacing the vehicle from time to time to obtain the correct position via GPS, but this is not always an option. There are some vehicles, using DVL, that are accurate to 0.01% of the distance traveled (Leonard et al., 1998).

A typical block diagram of an inertial navigation system is shown in Figure 4. Acoustic navigations use external transducers which return the acoustic signal send out by the vehicle. The travel time of the signal determines the position of the vehicle. However, reflections and differences in signal speed can negatively influence the measurement. Geophysical navigation uses a priori knowledge of terrain to identify the current position. The disadvantage of this system is that the maps of the terrain must exist, and this is not always the case. It also requires a lot of computational power to find the current position. To minimize this, the sys tem can be used in combination with dead-reckoning to limit the search area. The reliability of the system depends on the accuracy of the a priori map. With concurrent mapping and localization the vehicle builds up a map of its environment and uses that map to navigate in real time.

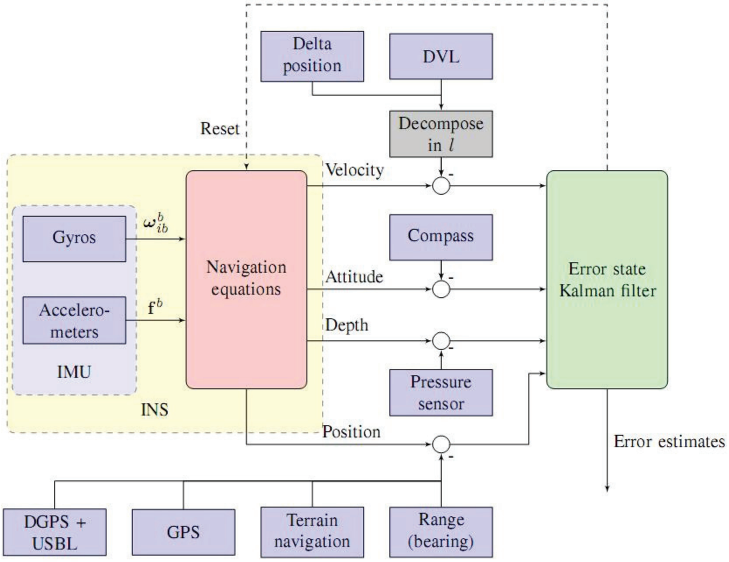


Fig. 4. Aided inertial navigation system (Fauske et al., 2007)

For short-range missions, up to 10km, calibrated inertial navigations systems can provide sufficient accuracy. The system can be extended with a DVL. For longer missions, up to 100km, the path taken by the AUV has a large effect on the accuracy. Concurrent mapping and localization works well for these missions, as long as the path contains many crossover points the technique corrects inaccuracies. Above 100km a geophysical navigation system is the only suitable solution. However, this technique is limited by the availability of maps (Stutters et al., 2008).

4. Propulsion, dive and buoyancy

The most common form of propulsion is via thrusters. For vertical movement thrusters or variable buoyancy systems can be used. The thrusters provide more accuracy and a faster response. If it is not a problem that the vehicle has to move horizontally in order to move vertically the vehicle can also use a single thruster for both the horizontal and vertical movement with the use of diving planes or a robotic wrist.

The kind of propulsion, the drive and the choice for the buoyancy are of great influence on the dynamics of the vehicle. There are a lot of choices that have to be made in the design phase. The horizontal movement of an AUV is usually empowered by thrusters. One of the reasons for this is that most underwater vehicles are powered by batteries (Smith et al., 1996) (see Section 6 for more information). In (Valavanis et al., 1997), 25 AUVs are described, of which the majority use thrusters for propulsion. The vertical movement can be done with thrusters or by a variable buoyancy system. The buoyancy of an AUV is the upward force on the vehicle that is caused by the surrounding water. If the buoyancy force is equal to the gravitational force on the vehicle, the vehicle is said to be neutral buoyant. It will neither sink nor rise.

When thrusters are used, the vehicle has neutral buoyancy (Serrani & Conte, 1999). The vehicle is then able to move vertically by using the thrusters. One of the advantages of this method is that the vehicle is able to hover without propulsion. A disadvantage of the technique is that the thrusters must remain on will moving vertically, thus consuming power.

There are four static and dynamic diving principles (Wolf, 2003): i) a piston type ballast tank; ii) a hydraulic pump based ballast system; iii) an air compressor based system; and iv) direct thrust systems. The first three concepts come from static diving technology, while the last concept is a dynamic diving technology.

The piston ballast tank (Figure 5a) is one of the most common static diving methods applied in submarine modeling. A piston ballast tank consists of a cylinder and a movable piston, and it works as a large syringe pump. With one end of the cylinder connected to surrounding water, movement of the piston sucks water in or pushes it out. When water fills the tank, negative buoyancy is achieved, so the AUV starts to descend. Conversely, when the tank is emptied, the AUV is positively buoyant, so it ascends. This setup also allows control of pitch motions of the AUV. Moreover, the pistons can be moved by a linear actuator, which is electrically easy to control. Hence, accurate depth control can be achieved with proper, yet straightforward, programming.

A hydraulic pumping system (Figure 5b) is similar to the piston ballast tank, but uses an internal reservoir of hydraulic fluid and a pump to actuate the piston's linear motion. Control of the valves and the pump for the hydraulic fluid allows it to flow in and out of the cylinders, so the surrounding water can be pumped in and out. Consequently, buoyancy of the AUV is changed.

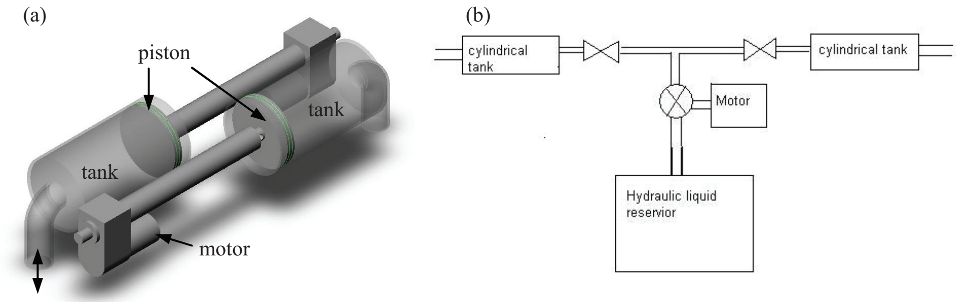


Fig. 5. Examples of two diving principles. (a) Piston ballast system with two tanks. (b) Schematic sketch of hydraulic pumping system

Air compressor systems are commonly used in some classes of submarine. The system is composed of a storage tank of compressed air, a water tank and two valves that are normally closed. To descend, the vent valve is opened, so the pressure difference results in water flowing in from the opening in the bottom of the water tank. When a desired amount of water is obtained for ballast, the vent valve is closed. In order to force the water out, the blow valve is opened to allow the compressed air into the tank so that water is pushed out via the bottom opening. Thus, by letting the water in and out of the water tank, the buoyancy of the AUV is changed.

Thrusters are a dynamic diving method. They require the AUV to be near neutrally buoyant. This approach uses the vertically mounted thrusters to force the AUV to dive. Turning off the thrusters or using them at a thrust less than the positive buoyancy allows controlled ascent. However, this method consumes a lot of power to keep the AUV under water, as the thrusters must remain powered at virtually all times. Being positively buoyant, however, this method is intrinsically failsafe, as the vehicle will come to the surface in the event of a power failure.

With a variable buoyancy system the vehicle is able to vary its buoyancy. The system usually contains a number of tanks that can be filled with water or gas. With this system the vehicle is able to move vertically by changing its buoyancy. Vertical movement and hovering is then possible without propulsion. The drawback of the system is that it is not as accurate as using thrusters. In (Tangirala & Dzielski, 2007) a variable buoyancy system is described that consists of two water tanks with pumps and valves. If more negative buoyancy is needed, the tanks are open to seawater. If positive buoyancy is needed, the water is pumped out of the tanks. In (Wasserman, 2003) a vehicle is proposed that uses air to acquire more positive buoyancy. The vehicle has a tank which can be filled with air coming from a compressed air tank. Water is drained from the tank when it is filled with air and more positive buoyancy is generated. There are also vehicles that use only one thruster for propulsion and do not have a variable buoyancy system (Cavallo & Michelini, 2004; Maurya et al., 2007). The vehicle described in (Cavallo & Michelini, 2004) uses a robotic wrist to position the thruster. This enables the vehicle to move horizontally and vertically. The vehicle is not able to move vertically without moving horizontally; however it is able to do vice versa. The vehicle described in (Maurya et al., 2007) cannot move vertically either without moving horizontally (again, vice versa it can), but instead of using a robotic wrist, it uses diving planes to move vertically.

5. Sensors and instrumentation

As described in Section 3 it is essential for an AUV to know its current position. In order to calculate that position a number of sensors are necessary. The most common sensor is a pressure sensor, it is used to measure the external pressure experienced by the vehicle. This pressure can be converted to a depth (Williams et al., 2006). For dead-reckoning navigation the vehicle speed is needed. There are numerous ways to measure the speed of the vehicle. Usually the velocity is measured using a compass and a water speed sensor. In (Modarress et al., 2007) a sensor is described which can measure the speed of the vehicle using particles that are present in the water. The speed of the particles is measured with diffractive optic elements. Small particles pass through two parallel light sheets and scatter light. The scattered light is collected and the speed of the particles is computed using the time-of-light and the physical separation of the two light sheets. The sensors are small, very accurate and insensitive for temperature changes and water pressure (Modarress et al., 2007). The problem with these techniques is that sea currents can add velocity components which are not detected by the speed sensor (Leonard et al., 1998).

For operations near the seabed, Doppler velocity logs (DVLs) can be used to measure the vehicle's velocity with respect to the ground. With these measurements the accuracy of the position estimation by the Kalman filter can improve greatly (Leonard et al., 1998; Lee et al., 2007). A DVL measures the Doppler shift of sonar signals reflected by the ground to obtain the velocity (Keary et al., 1999). The system becomes less accurate at low speeds. The correlation velocity log (CVL) is based on the same principle as the DVL, but emits two pulses in close succession. The echoes from the seabed are compared and used to calculate the velocity. This technique is more accurate at low speeds (Keary et al., 1999). Both systems are not influenced by sea currents.

The inertial navigation system (INS) uses accelerometers and gyroscopic sensors to detect the acceleration of the vehicle (Stutters et al., 2008). The measurements are not influenced by sea currents and are therefore more accurate. However, the system is more expensive than the velocity sensors (Leonard et al., 1998). INSs used to be equipped with mechanical gyroscopes. The latest INS uses laser gyroscopes or fiber optic gyroscopes that have no moving parts. This means no friction, leading to more accuracy. In (Fauske et al., 2007) sensor fusion is used to provide more accuracy to the INS. An error-state Kalman filter estimates the drift of the inertial sensors, using external information as measurement, e.g. a DVL and position updates by a mother ship, see Figure 4. With the use of more sensors for a number of parameters, a higher accuracy is achieved (Majumder et al., 2001).

Another aspect for which sensors are needed is obstacle detection. The vehicle must be able to detect obstacles before crashing into them. According to (Majumder et al., 2001) underwater cameras and active sonar are two of the most common sensors for obstacle detection. (Majumder et al., 2001) also states that there should always be at least two sensors, because in a sub-sea environment the information from one sensor can be of poor quality. Therefore, in the technique proposed both the information from the sonar and the camera are used for obstacle detection. Ultrasonic/acoustic sensor systems allow detection of objects far beyond the range of video. Current AUVs detect objects with long range sonar. Lower frequency waves suffer less attenuation in water than higher frequencies. However, the resolution for imaging sonars is better at higher frequencies (Toal et al., 2005).

In (Toal et al., 2005) a new technique is proposed using optical fibers for object detection. Two different sensors are used: one provides information without contact, the other provides information using contact with the object. The first is an extrinsic sensor which transmits light from the sensor end, if there is an object the light will reflect and received by a detector. The second sensor is an intrinsic sensor which does not transmits the light but contains the light within the fiber. A deformation of the fiber, so if the fiber touches an object, has a detectable effect on the light within the fiber. The vehicle described in (Williams et al., 2006) has two sonars. One is used for the determination of depth and obstacle detection. The other is an imaging sonar, which is used to build a map of the environment.

The main sensors for an AUV are a depth sensor, a compass and a speed sensor. With these sensors the vehicle can estimate its position. It is desirable to equip the vehicle with a Doppler velocity log to increase the accuracy of the estimates. Inertial navigation systems with laser or fiber optic gyroscopes are more expensive but also more accurate than the standard speed sensors. Sonar, underwater cameras or optical fibers can be used for obstacle detection.

6. Power supply

As mentioned in Section 1 an AUV must contain its own power. The most common power supply for AUVs is batteries (Smith et al., 1996). A number of AUVs use fuel cells for their power supply (Takagawa, 2007; Haberbusch et al., 2002) and a few use solar power (Jalbert et al., 2003). The advantages of using an electric propulsion over thermal propulsion are silent operation, ease of speed control and the simplicity (Smith et al., 1996).

The silver zinc battery was the most used power source in AUVs for 40 years (Smith et al., 1996; Winchester et al., 2002). But due to recent developments in lithium-ion batteries this has changed (Wilson & Bales, 2006). In (Bradley et al., 2001) different sorts of batteries are summarized. Batteries can be either primary or secondary, meaning non-rechargeable and rechargeable respectively. Most batteries described in the paper are secondary because the majority of batteries in AUVs are rechargeable. Primary batteries usually have a better endurance than secondary, but are more expensive in use.

The most common primary battery is alkaline. It is cheap and easy to work with. However they outgas hydrogen and are temperature sensitive. The lithium primary battery has a very high energy density, but is expensive. Of the secondary batteries lead acid cells are the easiest to work with, but they also leak hydrogen. Nickel cadmium cells are well known and have a flat discharge curve, but it is difficult to determine the state of charge. Nickel zinc cells have a good cycle life and a good energy density. Lithium-ion cells have the highest energy density of the secondary cells. However the circuitry to operate them in a system is complex. Silver zinc cells can handle high power spikes, but are expensive and have a very limited cycle life. As stated before the developments in the technology for lithium-ion batteries made them an attractive alternative to the silver zinc batteries (Wilson & Bales, 2006).

An overview of the batteries and their characteristics is given in Table 2. Usually every load has its own inverter which is powered directly from the main bus (Bradley et al., 2001), as opposed to the situation where the batteries powers several inverters that generate the voltages needed by the different components. This is because inverters are inexpensive and efficient.

Takagawa (Takagawa, 2007) describes a fuel cell for the power supply of the AUV. The fuel cell has a fairly larger capacity than batteries, but the problem with the system is that it is required to be installed in pressure vessels. Also proposed is a mechanism similar to the pressure compensation mechanism used in batteries, for the fuel/oxidant container. The system is small compared to systems using pressure resistant containers. Hydrogen peroxide has already been used with pressure compensation mechanism and is therefore chosen as oxidant, methanol is selected as fuel. The system can be used for underwater vehicles with a very long cruising capability (~10, 000km). The fuel cells can also reduce the logistics burden of the vehicle if the fuel and oxidant are stored in a high density format. Fuel cells that operate on hydrogen and oxygen are attractive power sources for AUVs because they are efficient, quiet, compact and easy to maintain. The total energy delivered by a fuel cell is limited only by the available fuel and oxygen (Haberbusch et al., 2002).

In (Jalbert et al., 2003) an AUV is described that operates on solar power. The main advantage is of course that the vehicle can stay in the water for months without having to be recharged. The vehicle surfaces daily to recharge its lithium-ion batteries.

The most commonly used power supply for AUVs nowadays is lithium-ion batteries. If a long operation time is required fuel cells are a good alternative. For very long operation times, solar power in combination with lithium-ion batteries is a good choice.

Chemistry	Energy density (Whr/kg)	Outgassing	Cycles	Comments
Alkaline	140	Possible, at higher temperature	1	Inexpensive, easy to work with
Li Primary	375		1	Very high energy density
Lead Acid	31.5	Yes	~100	Well established, easy to work with
Ni Cad	33	If overcharged	~100	Very flat discharge curve
Ni Zn	58.5	None	~500	Emerging technology
Li Ion	144	None	~500	Complex circuitry
Silver Zn	100	Yes	~30	Can handle very high power spikes

Table 2. Battery chemistries and their characteristics (Bradley et al., 2001)

7. Canterbury AUV

7.1 Background

As more than half of our oceans are deeper then 3km, one direction of the AUV developments is to explore deep waters. However, development of such AUVs imposes extreme design specifications for the hardware (Uhrich & Watson, 1992), incurring an unaffordable cost for most labs. By contrast, AUVs for shallow waters recently have gained more attention because of their potentially wide use combined with affordable cost. At the University of Canterbury, an AUV prototype has been recently designed with the primary purpose for inspecting and cleaning the sea chests of ships (Figure 6a), an application with significant impact in the area of bio-security.

Sea chests are the intake areas in the hulls of ships for seawater used for ballast, engine-cooling and fire-fighting. Grates on the outside of the chests prevent large organisms from being entrained in the water but many smaller organisms (Figure 6b) survive in the sea chests and are transported around the world creating a bio-security risk. The New Zealand government has placed a high priority on the development of systems and tools to protect native flora and fauna against invasion by unwanted foreign organisms.

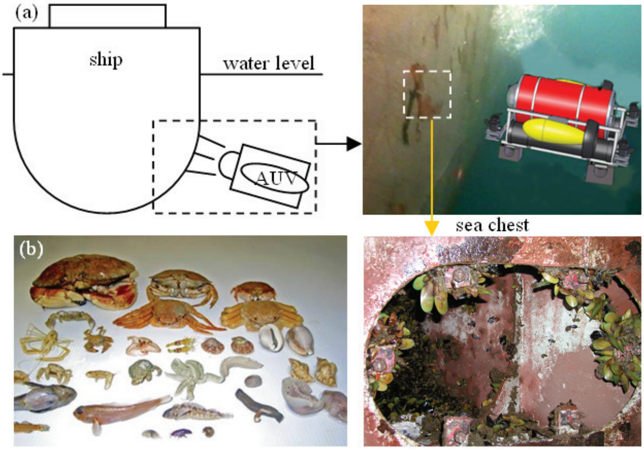


Fig. 6. Autonomous underwater vehicle inspecting and cleaning sea chest of ships. (a) The diagram of the AUV working on the sea chest of the ship. (b) A range of foreign invaders hiding in the sea chest.

To optimize the knowledge of, and reaction to, this threat, the first task is to inspect the sea chests and collect information about the invaders. Currently, divers are sent to do the job, which has inherent problems, including: i) high cost, ii) unavailability of suitably trained personnel for the number of ships needing inspection, iii) safety concerns, iv) low throughput, and v) unsustainable working time underwater to do a thorough job. To reduce the working load of divers and significantly accelerate inspection and/or treatment, it would be highly desirable and efficient to deploy affordable AUVs to inspect and clean these ship sea chests. Thus, this paper presents a low cost AUV prototype emphasizing the unique design issues and solutions developed for this task, as well as those attributes that are generalizable to similar systems. Control and navigation are being implemented and are thus not covered here.

7.2 Hull design

Figure 7. shows the AUV prototype (weighing 112kg, positively buoyant), which consists of basic components, including main hull, two horizontal propellers, four vertical thrusters, two batteries, an external frame, and electronics inside the main hull. This section focuses on the hull design.

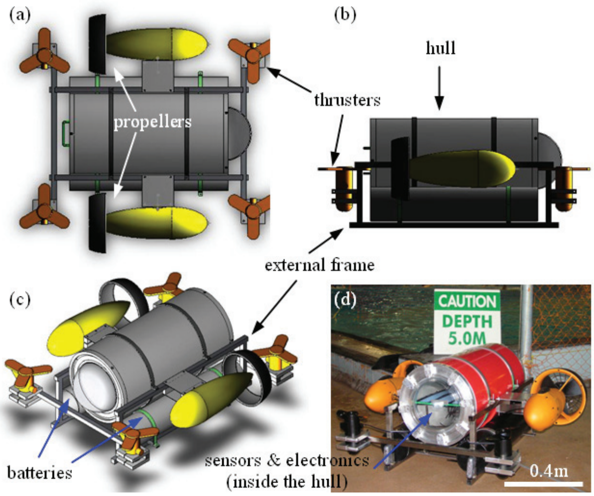


Fig. 7. The hull structure of the vehicle. (a)-(c) Design drawings of the vehicle: (a) Top view. (b) Side view. (c) Isometric view. (d) Real picture of the in-house made vehicle

The foremost design decision is the shape of the hull. Inspired by torpedoes and submarines, a cylindrical hull has been selected. A cylinder has favourable geometry for both pressure (no obvious stress concentrations) and dynamic reasons (minimum drag). To make the hull, three easily accessible materials were compared. The first option is to use a section of highly available PVC storm water pipe. The second option involves having a hull made from a composite material, such as carbon fibre or fibre glass. Mandrel spinning of such a hull will allow more freedom in radial dimensions. The process can in fact incorporate a varying radius along the length resulting in a slender, traditional hull. However, this process requires a large amount of design and set up time. A less desirable third option is to use a section of metal pipe, which is prone to corrosion and has a high weight and cost. As a result, the PVC storm water pipe option was selected.

Two caps were designed to complete the hull, and are attached to each end of the pipe such that they reliably seal the hull. The caps also allow access to the interior for easy repair and maintenance. The end cap design incorporates an aluminium ring permanently fixed to the hull and a removable aluminium plug. The plug fits snugly into the aluminium ring. Sealing is achieved with commercially available O-rings. Sealing directly to the PVC hull would have been more desirable; however this option was not taken for two main reasons. First, PVC does not provide a sealing surface as smooth and even as aluminium and is extremely hard to machine in this case due to the size of the pipe. Second, the PVC pipe is not perfectly round and subject to significant variability, which would make any machined aluminium cap subject to poor fit and potential leakage, decreasing reliability.

The design choices made can thus better manage these issues. More specifically, the design is based on self-sealing where greater outside pressures enforce greater connection between the cap, seals, and PVC hull portion. The O-ring seal employed is made of nitrile, which is resistant to both fresh and salt water.

7.3 Propulsion and steering

The design incorporates 2 horizontal thrusters mounted on both sides of the AUV to provide both forward and backward movement. Yaw is provided by operating the thrusters in opposing directions. The thrusters are 12V dive scooters (Pu Tuo Hai Qiang Ltd, China) that have a working depth of up to 20m.

The dive scooters are lightly modified to enable simple attachment to the external frame of the AUV. The thruster mounts consist of two aluminium blocks, which, when bolted together, clamp a plastic tab on each thruster. These clamps provide a strong, secure mount that can be easily removed or adapted to other specifications.

The force that can be generated by the thruster is characterized, as shown in Figure 8. The significant linearity between the thruster force and the applied duty cycle will significantly facilitate the design and implementation of any control scheme.

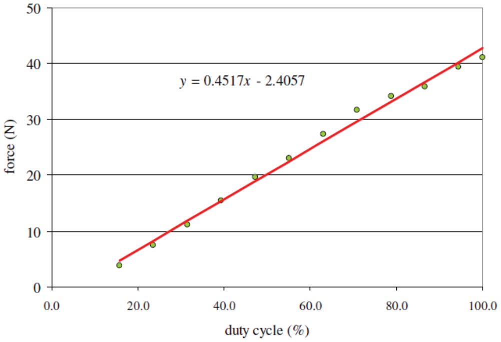


Fig. 8. Calibration of the motor: force with respect to duty cycle

A fluid drag force model is established to evaluate the speed that the AUV can achieve. Figure 9. shows the relationship between the drag forces with respect to the relative velocity of the vehicle. Under the full load of the two thrusters, the vehicle is able to achieve a maximum forward or backward speed of 1.4m/s (~5km/hour).

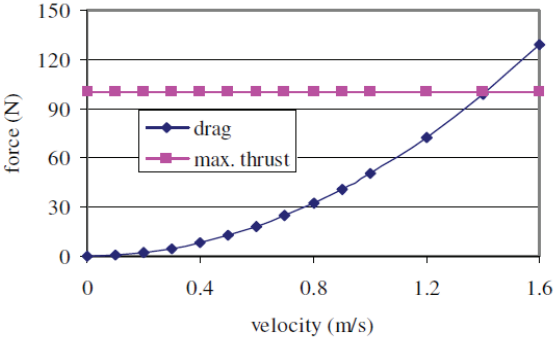


Fig. 9. Drag force of the AUV with different velocities

7.4 Ballast and depth control

Selection of a suitable ballast system is dependent on various factors, such as design specifications, size and geometry of the AUV hull, depth required, and cost. In this design, the hull is made of a PVC pipe with an outer diameter of 400mm and a length of 800mm. The required working depth is 20m. Hence, the ballast system selected not only has to meet the basic requirements enumerated above, but must also be able to fit in the hull. Preferably, all are at a relatively low cost.

First, installing two (2) 160mm inner diameter ballast tanks of 250mm length provides a net force of ±5kg. Additionally, the force required to actuate the piston head at 20m is calculated to be approximately 6000N. To generate such force on the piston head, a powerful linear actuator is needed. The specific linear actuator (LA36 24V DC input, 6800N max load, 250mm stoke length) can be sourced from Linak Ltd in New Zealand. However, the linear actuator has a duty cycle of 20% at max, which means that for every 20s continuous work, it must remain off for 80s before operating again, allowing the AUV to float uncontrolled. In addition, the cost of one linear actuator is US$1036, which would imply that similar actuators with longer duty cycles would cost a larger amount at this time.

Taking the second option, a hydraulic pumping system can be customized from Scarlett Hydraulics Ltd, New Zealand. The overall system has dimensions of 500mm × 250mm × 250mm. It consists of a 1.2KW DC motor, a pump, a 4L hydraulic fluid tank, two dual solenoid valves and two cylindrical tanks. This system meets the required specifications, but has some drawbacks. In particular, it occupies too internal space of the hull, and weighs approximately 20kg (a significant addition of weight). In this case, the overall hydraulic pumping system will cost up to approximately US$2264.

The third option air compressor system is cost effective and is easy to operate by controlling the vent and blow valves. However, the lack of accuracy in controlling compressed gas is a major disadvantage. In addition, performance and operating time are limited by the amount of stored gas. In this design, a 10L tank would be needed to fulfil the changes in buoyancy. In other words, a gas cylinder containing 10L of air compressed to at least 3bar is required for a single diving and rising cycle. Hence, to refill the gas cylinder, the AUV must float to the waters surface before all the air runs out or risk being lost. Regarding the on-site requirement that the AUV should operate for hours, the air tank must either be much bigger or far more highly pressurize, which leads to safety issues.

The fourth option thrusters are different from the previous three systems that all had to be installed inside the AUV. In contrast, thrusters can be attached externally. Hence, sealing is not as critical as it is for the other concepts. If the vehicle is trimmed positively buoyant, it is also reasonably fail-safe, unlike the other three methods. Additionally, the thrusters can be sourced from Pu Tuo Hai Qiang Ltd, Zhou Shan, China for US$55/unit, a reduction of 12-20× in cost if two are used. Each thruster fits in a 215mm × 215mm × 80mm box, and is driven by a 12V DC motor with a max thrust force of 5kg under water. By mounting the desired number of thrusters, a wide range of motions can be controlled, such as pitch and roll control.

Finally, each concept has its own advantages and disadvantages. Comparisons are summarized in Table 3. In this design, the major driving factors for the selection of ballast system are the cost and reliability. Piston ballast tank and thruster systems are reliable since these two depth control methods have been widely employed in most autonomous underwater vehicle development. Considering the cost, the thruster system is more effective. Hence, the thruster system is chosen as the final design.

	Diving Tech	Installation	Buoyancy	Sealing	Reliability	Overall Cost *
Piston ballast tanks	Static	Internal	+ ve, - ve, Neutral	Difficult	Used in most remote submarines	$2500
Hydraulic pumping system	Static	Internal	+ ve, - ve, Neutral	Difficult	Not reliable	$2710
Air compressor	Static	Internal	+ ve, - ve, Neutral	Difficult	Air on board is limited, compressed air hard to handle	$420
Thrusters	Dynamic	External	+ ve	None	Used in most ROVs with big size	$500

Table 3. Ballast comparison. * The cost is estimated as an overall system

There are four thrusters vertically mounted around the AUV with one at each corner (See Figure 7). Mounting four thrusters produces a total of 20kg thrust force at full load, and allows a wide range of motion control. They enable the control of not only the vertical up and down motion, but pitch and roll motions. To achieve this control, each thruster is connected to a speed control module that can be controlled via a central microprocessor. By inputting different digital signals, various forces thus speeds are generated. Therefore, desired motion control can be obtained by different combinations.

7.5 Electronics and control

7.5.1. Power supply

For long term operation, this design must locate the power supply on-board, unlike many current models that receive power over an umbilical link (Chardard & Copros, 2002). Since all the systems onboard the AUV are electric, sealed lead acid batteries are chosen for the power supply. These batteries have high capacity and can deliver higher currents, than other types of rechargeable battery (Schubak & Scott, 1995; Bradley et al., 2001). They are stable, inexpensive, mechanically robust and can work in any orientation, all of which are important considerations in a vehicle of this type. To supply enough current for the entire machine several batteries have to be joined together. Instead of adding dead weight to achieve neutral buoyancy extra batteries can be added as needed so that the total operating time of the AUV is higher than that required for a given application.

It is also highly desirable to locate the battery compartments separate from the main hull so that they can be interchanged in the field without opening the sealed main hull. To accommodate this requirement two tubes are fitted below the hull to house batteries. Within these tubes the batteries are connected to two bus bars. Each battery is fused prior to connecting to the bus bar, and the bars are isolated to the greatest extent possible to increase safety. These bus bars are then wired into the main hull, where a waterproof socket enables the quick interchange of battery compartments. A similar bus system exists inside the hull with connections to motors and electronic power supplies. Each of these internal connections is similarly fused. Longer term, it would be desirable to intelligently monitor the bus to track the state of each battery and overall power consumption.

7.5.2. Central processing unit

The central processing unit is responsible for accessing sensors, processing data and setting control outputs such as motor speeds. Several systems are considered for this unit, an embedded system using microprocessors, FPGAs or a small desktop PC. A microprocessor system, most likely based on an ARM processor would have low cost, size and power requirements and is easy to interface to both analogue and digital sensors, motors and other actuators. The processing power and memory allocations of these microprocessors are all more than sufficient for the simple control tasks likely to be required, but would struggle with larger sensor or processing tasks, such as image processing. An FPGA system would also be small and have low power requirements, but would be more expensive. While FPGAs work very well for fast, complex processing tasks such as image processing, their complexity in design and programming necessitates their use in parallel with other more flexible CPU choices. The last system considered is a small desktop PC. Although a desktop PC is bigger, more expensive and consumes more power than either of the prior two options, it provides immense processing power, memory and a diverse range of peripherals. It is therefore chosen in this initial design for the following primary reasons:

- Added power requirements were not an issue since we have a sizeable power supplies.
- Processing power is more than adequate for this initial design and future developments.
- Large volumes of memory are available, both volatile for program execution and solid state for storage of gathered data.
- Despite not having direct access to sensors and control units, a diverse range of peripherals available can be used, including USB, RS232 and Ethernet, enabling a potentially greater range of sensors and sensor platforms for developing broad ranges of specific applications.
- A USB module is already provided for a webcam for initial image sensing applications and an Ethernet module is provided for remote connection.

An AMD Sempron 3000+ processor and ASUS M2N-PV motherboard are used for this purpose. These models have lower power requirements and heat generation. Software interfaces this unit with sensors and motor controllers, as well as to a remote control PC. An automotive power supply (Exide, Auckland, NZ) is used to provide power for the computer. It takes a 12V DC input and converts it to the ATX standard power supply required by the PC. This module is also designed to be used in an electrically noisy and hostile environment and is ideally suited the specific design situations considered.

7.5.3. Sensors

When the AUV is used autonomously, after development there will be a large and extensive sensor suite onboard. Currently, the sensors onboard measure

- water pressure, from which depth can be determined
- water temperature, inner hull temperature and humidity

- the AUV position in the three principal axes: yaw, pitch and roll
- visual or digital image feedback via a webcam.

Submersible pressure sensors that are salt water tolerant and can measure up to the pressures required are difficult to acquire at low cost. The sensor chosen was sourced from Mandeno Electronics for US$121. This sensor measures up to twice the depth required, and outputs an analogue output between 0 and 100mV. Thermocouples from Farnell Electronics (Christchurch, New Zealand) are used to measure the water temperature, and provide an analogue output relative to the temperature difference between the two ends of the thermocouple. TMP100 sensors (Texas Instruments) are used to measure the base temperature of the thermocouple, and the hulls interior temperature. These sensors give a digital output using the I2C protocol. A HF3223 humidity sensor (Digi-Key) is used to measure humidity inside the hull. A MMA7260QT accelerometer (Freescale Semiconductor) is used to calculate orientation. The accelerometer has a 0-2.5V analogue output. The connection of the sensors is shown in Figure 10.

To eliminate signal noise, An Atmel AT90USB82 microprocessor is connected to the USB ports of the computer to move all noise sensitive data to the acquisition points. The analogue sensors are amplified using an INA2322 instrumentation amplifier, if necessary, and read by an ADS7828 analogue to digital converter. This converter is then connected to the Atmel microprocessor using a common I2C bus with the TMP100. The humidity sensor is attached to a clock input which converts the frequency based signal to a humidity based reading. The microprocessor performs some basic processing on this data, temperature compensating the pressure sensor and thermocouple, and calculating yaw, pitch and roll from the accelerometer readings.

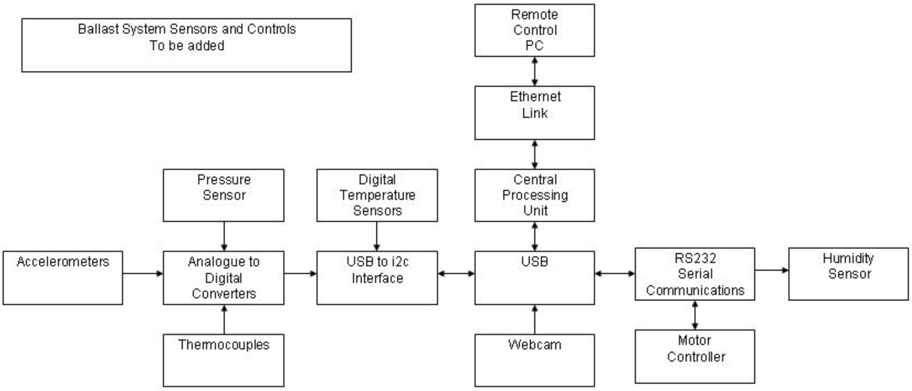


Fig. 10. The block diagram for electronic systems and control

Visual or digital image sensing is included via a Logitech webcam connected directly to the on-board computers USB port. The video stream can be sent back over a wireless remote control network connection to the remote PC. At this stage, no image processing is done on this stream on-board, and it is included purely to assist in manual control of the AUV at this time, and for use in later application development.

7.5.4. Propulsion motor driver

For the six motors (two for horizontal propulsion, and four for vertical ballast control), three (3) RoboteQ AX2500 motor controllers are used for control. Each controller is able to control two motors up to 120 amps, much higher than the 25 amps needed by the motors selected. The controllers are controlled via RS232 (serial port) interfaces, which are already available on the computer motherboard. Computer control of the controllers is easily achieved through a LabView or MATLAB interface, either manually or automatically, where both interfaces have been implemented to allow greater user ease of use.

7.5.5. Control system and communications

During testing and development, remote control is required for the AUV. Sensors readings need to be sent to a user, and control signals sent back to the AUV. Displaying the video feed from the webcam is also desired to provide the operator with visual feedback. High frequency radio transmissions are impossible underwater due to the high losses encountered during the air/water boundary (Leonessa et al., 2003). Lower frequency transmissions could have been used to communicate with the AUV, but they do not possess enough bandwidth to send the required data. An umbilical Ethernet cable is being used for this remote link between the AUV and an external control computer for this development phase. Figure 6. shows the electronics and control structure. Note that in an actual, developed application, or final development thereof, the robot will be acting autonomously and this umbilical will not be required.

8. Conclusions and future work

AUVs have a lot of potential in the scientific and military use. With the development of technologies, such as accurate sensors and high density batteries, the use of AUVs will be more intensive in the future. In this book chapter, several subjects of an AUV have been reported. For every subject some of the techniques used in the past and the techniques used nowadays are described. For every aspect a suitable technique for an AUV is given. To show how the state-of-the-art technologies could be used in AUVs, an AUV prototype developed recently at the University of Canterbury has been detailed in design.

The AUV was specially designed and prototyped for shallow water tasks, such as inspecting and cleaning sea chests of ships. It features low cost and wide potential use for normal shallow water tasks with a working depth up to 20m, and a forward/backward speed up to 1.4m/s. Each part of the AUV is deliberately chosen based on a comparison of readily available low cost options when possible. The prototype has a complete set of components including vehicle hull, propulsion, depth control, sensors and electronics, batteries, and communications. The total cost for a one-off prototype is less than US$10,000. With these elements, a full range of horizontal, vertical and rotational control of the AUV is possible including computer vision sensing. The overall underwater vehicle will be a good platform for research, as well as for its specific applications, many of which are growing in importance like the sea chest inspection case noted here.

The controls of the vehicle are under development. The vertical motion control uses the feedback from the pressure sensor, while the horizontal motion control uses an inertial measurement unit (Microstrain GX2 IMU, VT, USA) to get information about the vehicle attitude and acceleration. The fluidic model (dynamic drag force) of the vehicle will be established by simulation and verified by experimental measurement. This model would be integrated in the control and navigation module of the vehicle.

9. References

Allmendinger, E. (1990). *Submersible Vehicle Systems Design*. Jersey City, NJ: Society of Naval Architects and Maringe Engineers, 1990.

Ballard, R. (1987). *The Discovery of the Titanic*. New York, NY: Warner/Madison Press Books, 1987.

Blidberg, D. (2001). The development of autonomous underwater vehicles (AUV): a brief summary, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2001)*, Seoul, Korea, May 2001.

Bradley, A.; Feezor, M.; Singh, H. & Sorrell, F. (2001). Power systems for autonomous underwater vehicles, *IEEE Journal of Oceanic Engineering*, Vol. 26, No. 4, 526–538.

Caccia, M. (2006). Autonomous surface craft: prototypes and basic research issues, *14th Mediterranean Conference on Control and Automation*, June 2006.

Cavallo, E. & Michelini, R. (2004). A robotic equipment for the guidance of a vectored thrustor AUV, *35th International Symposium on Robotics ISR 2004*, 2004.

Chardard, Y. & Copros, T. (2002). Swimmer: final sea demonstration of this innovative hybrid AUV/ROV system, *Proceedings 2002 International Symposium on Underwater Technology*, Tokyo, Japan, Apr. 2002, 17-23.

Curtin, T. & Bellingham, J. (2001). Autonomous ocean-sampling networks, *IEEE Journal of Oceanic Engineering*, Vol. 26, 421-423.

Evans, J. & Meyer, N. (2004). Dynamics modeling and performance evaluation of an autonomous underwater vehicle, *Ocean Engineering*, Vol. 31, 1835-1858.

Fauske, K.; Gustafsson, F. & Hegrenaes, O. (2007). Estimation of AUV dynamics for sensor fusion, *10th International Conference on Information Fusion 2007*, 1-7, July 2007.

Feng, Z. & Allen, R. (2004). Reduced order H1 control of an autonomous underwater vehicle, *Control Engineering Practice*, Vol. 12, 1511-1520.

Fossen, T. (1994). *Guidance and control of ocean vehicles*. New York: John Wiley and Sons Ltd., 2-nd ed., 1994.

Fryxell, D.; Oliveira, P.; Pascoal, A.; Silvestre, C. & Kaminer, I. (1996). Navigation, guidance and control of AUVs: an application to the MARIUS vehicle, *Control Engineering Practice*, Vol. 4, No. 3, 401-409.

Gaccia, M. & Veruggio, G. (2000). Guidance and control of a reconfigurable unmanned underwater vehicle, *Control Engineering Practice*, Vol. 8, 21-37.

Griffiths, G. & Edwards, I. (2003). AUVs: designing and operating next generation vehicles, *Elsevier Oceanography Series*, Vol. 69, 229-236.

Haberbusch, M.; Stochl, R.; Nguyen, C.; Culler, A.; Wainright, J. & Moran, M. (2002). Rechargeable cryogenic reactant storage and delivery system for fuel cell powered underwater vehicles, *Proceedings Workshop on Autonomous Underwater Vehicles*, 103-109, June 2002.

Horgan, J. & Toal, D. (2006). Review of machine vision applications in unmanned underwater vehicles, *9th International Conference on Control, Automation, Robotics and Vision*, Dec. 2006.

Hsu, C.; Liang, C.; Shiah, S. & Jen, C. (2005). A study of stress concentration effect around penetrations on curved shell and failure modes for deep-diving submersible vehicle, *Ocean Engineering*, Vol. 32. No. 8-9, 1098-1121.

Jalbert, J.; Baker, J.; Duchesney, J.; Pietryka, P.; Dalton, W.; Blidberg, D.; Chappell, S.; Nitzel, R. & Holappa, K. (2003). A solar-powered autonomous underwater vehicle, *OCEANS 2003*.

Jalving, B. (1994). The NDRE-AUV flight control system, *IEEE Journal of Oceanic Engineering*, Vol. 19, 497-501.

Kaminer, I.; Pascoal, A.; Khargonekar, P. & Coleman, E. (1995). A velocity algorithm for the implementation of gain-scheduled controllers, *Automatica*, Vol. 31, No. 8, 1185-1191.

Keary, A.; Hill, M.; White, P. & Robinson, H. (1999). Simulation of the correlation velocity log using a computer based acoustic model, *11th International Symposium Unmanned Untethered Submersible Technology*, 446-454, August 1999.

Kondoa, H. & Ura, T. (2004). Navigation of an AUV for investigation of underwater structures, *Control Engineering Practice*, Vol. 12, 1551-1559.

Lee, P.; Jun, B.; Kim, K.; Lee, J.; Aoki, T. & Hyakudome, T. (2007). Simulation of an inertial acoustic navigation system with range aiding for an autonomous underwater vehicle, *IEEE Journal of Oceanic Engineering*, Vol. 32, 327-345.

Leonard, J.; Bennett, A.; Smith, C. & Feder, H. (1998). Autonomous underwater vehicle navigation, *Proceedings IEEE ICRA Workshop Navigation Outdoor Autonomous Vehicle*, May 1998.

Leonessa, A.; Mandello, J.; Morel, Y. & Vidal, M. (2003). Design of a small, multi-purpose, autonomous surface vessel, *Proceedings OCEANS 2003*, Vol. 1, San Diego, CA, USA, 2003, 544–550.

Lygouras, J.; Lalakos, K. & Tsalides, P. (1998). THETIS: an underwater remotely operated vehicle for water pollution measurements, *Microprocessors and Microsystems*, Vol. 22, No. 5, 227–237.

Majumder, S.; Scheding, S. & Durrant-Whyte, H. (2001). Multisensor data fusion for underwater navigation, *Robotics and Autonomous Systems*, Vol. 35, 97-108.

Maurya, P.; Desa, E.; Pascoal, A.; Barros, E.; Navelkar, G.; Madhan, R.; Mascarenhas, A.; Prabhudesai, S.; Afzulpurkar, S.; Gouveia, A.; Naroji, S. & Sebastiao, L. (2007). Control of the Maya AUV in the vertical and horizontal planes: theory and practical results, *Proceedings MCMC2006 - 7th IFAC Conference on Manoeuvring and Control of Marine Craft*, 2007.

Modarress, D.; Svitek, P.; Modarress, K. & Wilson, D. (2007). Micro-optical sensors for underwater velocity measurement, *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 235-239, April 2007.

Monteen, B.; Warner, P. & Ryle, J. (2000). Cal poly autonomous underwater vehicle, California Polytechnic State University, 2000.

Paster, D. (1986). Importance of hydrodynamic considerations for underwater vehicle design, *OCEANS*, Vol. 18, 1413-1422, September 1986.

Ridao, P.; Batlle, J. & Carreras, M. (2001). Model identification of a low-speed AUV, In Control Applications in Marine Systems. *International Federation on Automatic Control*, 2001.

Rife, J. & Rock, S.M. (2002). Field experiments in the control of a jellyfish tracking ROV, in *MTS/IEEE Oceans '02*, Vol. 4, 2002, 2031-2038.

Ross, C. (2006). A conceptual design of an underwater vehicle, *Ocean Engineering*, Vol. 33, No. 16, 2087-2104.

Schubak, G. & Scott, D. (1995). A techno-economic comparison of power systems for autonomous underwater vehicles, *IEEE Journal of Oceanic Engineering*, Vol. 20, No. 1, 94-100.

Serrani, A. & Conte, G. (1999). Robust nonlinear motion control for AUVs, *IEEE Robotics & Autonomation Magazine*, Vol. 6, 33-38.

Smallwood, D. & Whitcomb, L. (2004). Model-based dynamic positioning of underwater robotic vehicles: theory and experiment, *IEEE Journal of Oceanic Engineering*, Vol. 29, No. 1, 169-186.

Smallwood, D.; Bachmayer, R. & Whitecomd, L. (1999). A new remotely operated underwater vehicle for dynamics and control research, *Proceedings UUST '99*, 1999.

Smith, P.; James, S. & Keller, P. (1996). Development efforts in rechargeable batteries for underwater vehicles, *Proceedings Autonomous Underwater Vehicle Technology Symposium, AUV'96*, 441-447, June 1996.

Stachiw, J. (2004). Acrylic plastic as structural material for underwater vehicles, *International Symposium on Underwater Technology*, 289-296, April 2004.

Stutters, L.; Liu, H.; Tiltman, C. & Brown, D. (2008). Navigation technologies for autonomous underwater vehicles, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, Vol. 38, 581-589.

Takagawa, S. (2007). Feasibility study on DMFC power source for underwater vehicles, *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 326-330, April 2007.

Tangirala, S. & Dzielski, J. (2007). A variable buoyancy control system for a large AUV, *IEEE Journal of Oceanic Engineering*, Vol. 32, 762-771.

Tivey, M.; Johnson, H.; Bradley, A. & Yoerger, D. (1998). Thickness of a submarine lava flow determined from near-bottom magnetic field mapping by autonomous underwater vehicle, *Geophysical Research Letters*, Vol. 25, 805-808.

Toal, D.; Flanagan, C.; Lyons, W.; Nolan, S. & Lewis, E. (2005). Proximal object and hazard detection for autonomous underwater vehicle with optical fibre sensors, *Robotics and Autonomous Systems*, Vol. 53, 214-229.

Uhrich, R. & Watson, S. (1992). Deep-ocean search and inspection: Advanced unmanned search system (AUSS) concept of operation, Naval Command, Control and Ocean Surveillance Center, RDT&E Division, San Diego, CA, Tech. Rep. NRaD TR 1530, Nov. 1992.

Valavanis, K.; Gracanin, D.; Matijasevic, M.; Kolluru, R. & Demetriou, G. (1997). Control architectures for autonomous underwater vehicles, *Control Systems Magazine*, Vol. 17, 48-64.

von Alt, C. (2003). Autonomous underwater vehicles, *Autonomous underwater Langrangian platforms and sensors workshop*, LaJolla, CA, March-April 2003.

Wasserman, K.; Mathieu, J.; Wolf, M.; Hathi, A.; Fried, S. & Baker, A. (2003). Dynamic buoyancy control of an ROV using a variable ballast tank, *OCEANS 2003*, Vol. 5, SP2888-SP2893, September 2003.

Wernli, R. (2001). Low cost AUV's for military applications: Is the technology ready? in *Pacific Congress on Marine Science and Technology 2001*, San Francisco, CA, July 2001.

Willcox, S.; Vaganay, J.; Grieve, R. & Rish, J. (2001). The bluefin BPAUV: An organic wide-area bottom mapping and mine-hunting vehicle, in *Proceedings UUST '01*, 2001.

Williams, C. (2004). AUV systems research at the NRC-IOT: an update, in 2004 *International Symposium on Underwater Technology*, 2004, 59-73.

Williams, S.; Newman, P.; Dissanayake, G.; Roseblatt, J. & Durrant-Whyte, H. (2006). A decoupled, distributed AUV control architecture, University of Sydney NSW, 2006.

Wilson. R. & Bales, J. (2006). Development and experience of a practical pressure-tolerant, lithium battery for underwater use, *OCEANS 2006*, 1-5, September 2006.

Winchester, C.; Govar, J.; Banner, J.; Squires, T. & Smith, P. (2002). A survey of available underwater electric propulsion technologies and implications for platform system safety, *Proceedings Workshop on Autonomous Underwater Vehicles*, 129-135, June 2002.

Wolf, M. (2003). *The design of a pneumatic system for a small scale remotely operated vehicle*, Bachalor's Thesis, MIT, May 2003.

8

General Concept of 3D SLAM

Peter Zhang, Evangelous Millos & Jason Gu
Dalhousie University
Canada

Simultaneous localization and mapping (SLAM) is a process that fuses sensor observations of features or landmarks with dead-reckoning information over time to estimate the location of the robot in an unknown area and to build a map that includes feature locations. In this chapter, a general model and its related solving algorithm for 3D SLAM are established. The method can be used for all of the situations in the mobile robot community. An underwater mobile robot is used as an example.

This chapter is organized as follows: Section 1 is the problem definition; Section 2 establishes all the models for 3D SLAM, including the robot process model, the landmark model, and the measurement model; Section 3 is the method for data association; Section 4 presents the algorithms to solve the SLAM; section 5 describes the multi-sensor related issues based on the underwater mobile robot cases; and Section 6 is the globally-consistent 3D SLAM for mobile robot in real environment.

1. Problem Definition

Assuming a 3D environment with randomly distributed landmarks and an autonomous mobile robot equipped with sensors (stereo camera, laser range finder, or sonar) which will move in this environment, by providing some proper input (robot speed and orientation), we need to determine the robot pose (position and orientation) and the position of detected landmarks during the robot navigation. Because of measurement noise and robot input noise, it is very difficult to compute a deterministic value for the robot pose and landmark position. We can only estimate their approximate value by using algorithms such as the Kalman filter, the Particle filter, and the Unscented Kalman filter. By using these algorithms, it is also possible to calculate the confidence of the estimation. In some areas of the robot's working environment, significant landmarks are sparse, especially in the underwater environment. A robot equipped with only one type of sensor may not obtain sufficient effective measurements, which would greatly affect the accuracy of the robot pose; therefore, more than one sensor will be used for the robot navigation in a real application. In this thesis, the SLAM problem in the 3D environment will be solved with multiple heterogeneous sensors. A general strategy will be proposed and related algorithms will be developed.

2. Models for 3D SLAM

2.1. Robot Process Model

Robot process model is a dynamic differential equation to describe the movement of a robot in a given environment and system input. It is related to the robot pose. The robot pose can be determined by its position and orientation. In a global coordinate system OXYZ, a robot position (p_v) is expressed by $(x, y, z)^T$, , and its orientation can be expressed by Euler angles, rotation matrix, axis and angle, or quaternion. From any one of the orientation representations, it is possible to compute the other representations. For simplicity, Euler angles are selected as a robot orientation state vector. Therefore, the state vector of the robot X_v can be expressed as

$$X_v = \begin{bmatrix} p_v^T \\ \theta_v^T \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \tag{1}$$

where T is the transpose of a matrix and assuming that the robot moves relative to its current pose with speed v and changes direction with Euler angles ($\delta\theta_x, \delta\theta_y, \delta\theta_z$), the input to the robot can be expressed by

$$U = \begin{bmatrix} v \\ \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} \tag{2}$$

where v is the robot speed in scalar, and the direction of the speed is always in the robot's forward pointing axis of its body. In order to simplify its implementation, the Euler angles need to be expressed in the form of a rotation matrix M_v

$$M_v = R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_z(\theta_z) \tag{3}$$

where Rz, Ry, and Rx are the rotation matrices which are the rotation around the z, y, x-axis, respectively, in right hand coordinate system with positive angle $\theta_z, \theta_y, \theta_z$, the positive angle is at counter-clockwise direction. Then, the robot process model can be expressed as

$$\theta_u(k+1) = \begin{bmatrix} \theta_x(k+1) \\ \theta_y(k+1) \\ \theta_z(k+1) \end{bmatrix} = \begin{bmatrix} f_1(\theta_x(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \\ f_2(\theta_y(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \\ f_3(\theta_z(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \end{bmatrix} \tag{4}$$

and

$$P_u(k+1)=\begin{bmatrix}x(k+1)\\ y(k+1)\\ z(k+1)\end{bmatrix}=\begin{bmatrix}x(k)\\ y(k)\\ z(k)\end{bmatrix}+M_u(k)\begin{bmatrix}\cos(\alpha)\\ \cos(\beta)\\ \cos(\gamma)\end{bmatrix}v\delta t \tag{5}$$

where δ_t is the sampling time, $M_v(k)$ is the rotation matrix, which corresponds to the Euler angles $\theta_x(k),\theta_y(k),\theta_z(k)$ at time k. In Equation(4), the angle $\theta_v(k+1)$ corresponds to the matrix Mv(k + 1), which has following equation

$$M_v(k+1)=M_v(\delta\theta)\cdot M_v(k) \tag{6}$$

where $M_v(\delta\theta)$ is a matrix which corresponds to the Euler angle $\delta\theta$ and in Equation.(5), And the α,β,γ are direction angles corresponding to the Euler angles, $\theta_x(k),\theta_y(k),\theta_z(k)$.

By combining the Equation(4) and (5), the process model can be written as a non-linear equation

$$X_u(k+1)=F(X_u(k),U(k)+\mu(k)+\omega(k) \tag{7}$$

where $\mu(k)$ the input is noise, and $\omega(k)$ is the process noise, at the sample time k. The noise is assumed to be independent for different k, white, and with zero mean and covariance $Q_v(k)$.

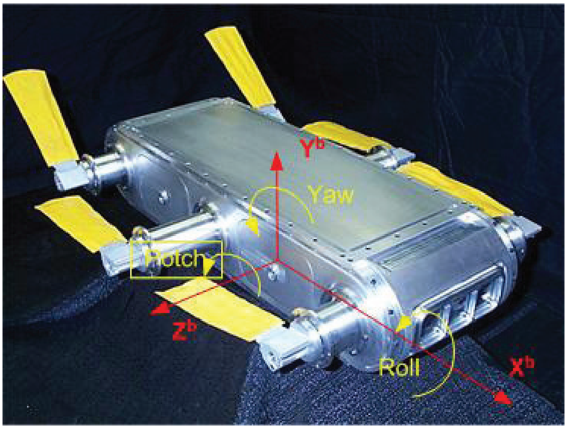


Fig. 1. Coordinate systems of an autonomous mobile robot.

2.2. Landmark Models

Landmarks can be classified into two types, artificial and natural. In a newly-visited natural environment, there is no artificial landmark for mobile robot navigation; therefore the natural landmarks are the only choice.

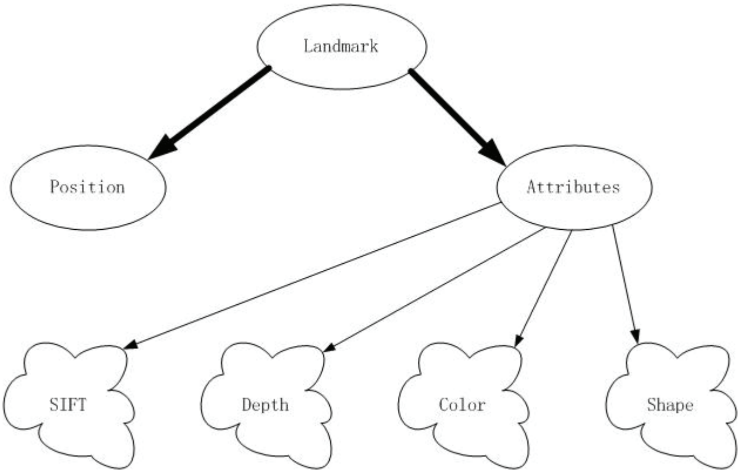


Fig. 2. General landmark expression.

A robot map consists of a set of landmarks. In order to provide enough information for robot navigation, every landmark should include position information and attribute information (Figure. 2). If the landmark position is known, a sensor's measurement of it can be used in the robot pose estimation with algorithms such as the extended Kalman filter or the particle filter. If the landmark position is unknown, an algorithm for SLAM will be applied to estimate the robot pose and landmark position by the aid of measurements. The attribute information will provide knowledge about the landmark which distinguishes it from other features, which is very useful for data association; therefore, landmark Li can be expressed as

$$L_i = \lfloor L_{i,position} \quad L_{i,attribute} \rfloor \tag{8}$$

where

$$L_{i,position} = L_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \tag{9}$$

During robot navigation, even though its environment is unknown, the landmarks for a map establishment are always assumed to have a static position. It is also assumed that attributes (or features) of a landmark will not change. In reality, features of a landmark may change with lighting conditions and sensor view point, therefore, the landmark i has the following evolution equation

$$L_i(k+1) = L_i(k) \tag{10}$$

where $i = 1,\dots,m$, which means there are m landmarks which will be used; *k* is the time which is used during the robot navigation.

2.3. Measurement Model

A mobile robot is always equipped with some type of sensors for its navigation. The sensors can obtain measurements of the relative location of the observed landmarks with respect to the robot. This observation can be expressed by a set of non-linear functions of the landmark's position relative to the robot position, which is called measurement model. Assuming the position of landmark *i* in the global coordinate system *OXYZ* is $(x_i, y_{i,} z_i)$. At time *k*, the robot has the pose $X_u(k)$. The measurement of the landmark i at this time can be computed by

$$Z_i = \begin{bmatrix} Z_{x_i}(k) \\ Z_{y_i}(k) \\ Z_{z_i}(k) \end{bmatrix} = M_v(k) \begin{bmatrix} x_i - x(k) \\ y_i - y(k) \\ z_i - z(k) \end{bmatrix} \cdot \tag{11}$$

The observation model in the non-linear equation is

$$Z_i = h(X_v(k), x_i, y_i, z_i) + \eta(k) = h(X_v(k), L_i(k)) + \eta(k) \tag{12}$$

where $\eta(k)$ is the observation noise, which is assumed with zero mean and covariance R_i and $h(\cdot)$ is the non-linear measurement function.

Without loss of generality, it is assumed that the measurement from every sensor is independent. If there are m features observed at time k, the measurement model is obtained by simply stacking Equation. (12) as

$$Z(k) = \begin{bmatrix} Z_1(k) \\ Z_2(k) \\ \cdots \\ Z_m(k) \end{bmatrix} = H(X_v(k), L(k)) + \eta(k) \cdot \tag{13}$$

An instance of a robot and five landmarks in 3D space is shown in Figure 3. When the robot moves in space, its sensor detects landmarks which are located in the view field of the sensor, then the robot pose and landmark position estimation can be performed. The estimated landmark position will be used to build a map which can be used by the robot for future navigation.

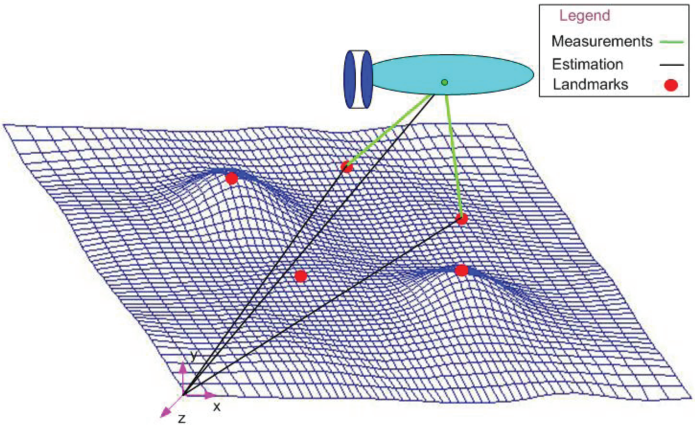


Fig. 3. An instance of a robot and features in 3D experiment case.

3. Data Association

There are two types of data associations - measurement between sensors received from multi-sensors, and measurement between adjacent times from a single sensor. In this thesis, only the data association between adjacent times from a single sensor will be addressed.

During robot navigation, if the sensor on the robot only observed one landmark, there would be no need for data association. Most sensors, such as camera, radar, laser, and sonar, will detect not only many real landmarks, but also many spurious landmarks; therefore, data association is a necessary step for landmark-based robot localization and object tracking.

A landmark defined in the previous Section 2.2. includes position and attributes (feature) components. If the attribute tuple is available, then data association can be implemented with this information; otherwise, maximum likelihood of measurement will be used.

3.1. Data Association by Feature Attribute

Data association by using the landmark's attribute is simple. A very good example is the image feature registration with SIFT (Scale Invariant Feature Transforms) features (Se et al., 2003). SIFT features in an associated image are among the best representations for the natural unstructured environment. The SIFT features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. The structure of the SIFT feature is as follows:

$$[\mathrm{u}, \mathrm{v}, \mathrm{gradient}, \mathrm{orientation}, \mathrm{descriptor1}, \cdots, \mathrm{descriptorM}]$$

where M is the number of the descriptor in SIFT feature. The position of landmark and the position of its related SIFT features (in an image) of a landmark have a non-linear relation. If a camera's physical position and orientation are given, the position of the landmark associated with the SIFT feature can be calculated by using its associated camera model (Sonka et al., 1999).

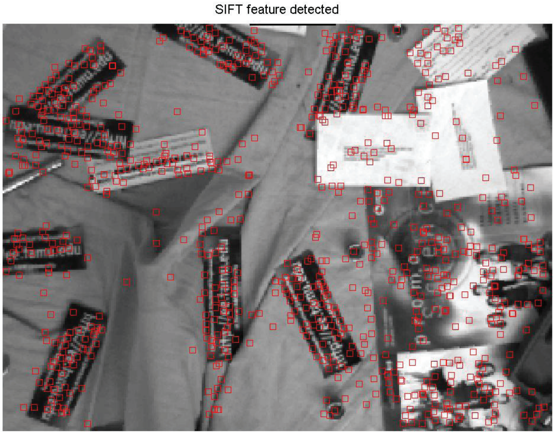


Fig. 4. SIFT features in an image.

An example of SIFT features in an image is shown in Figure 4. The data association for SIFT features can be carried out by using their feature descriptor directly. SIFT features correspondence between two adjacent images obtained from a moving camera at different view points after the implementation of data association is displayed in Figure 5.

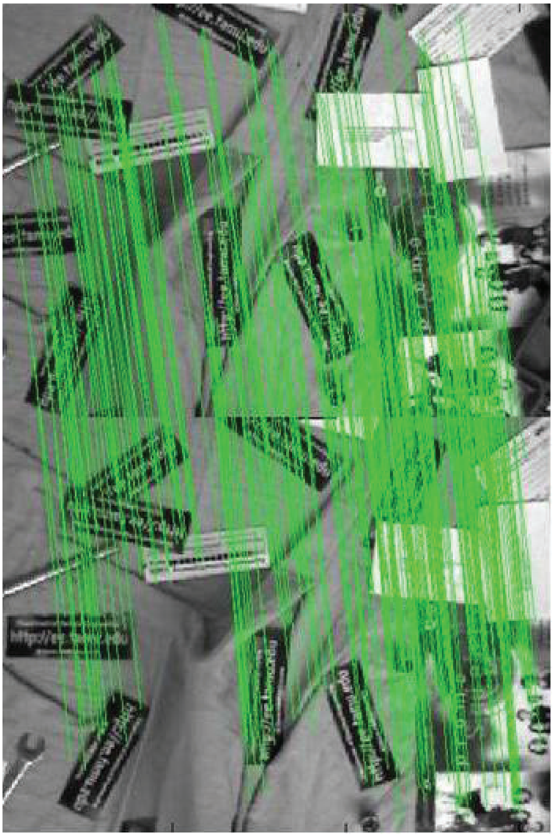


Fig. 5. Result of SIFT feature correspondence between two adjacent images obtained from a moving camera at different view points after data association.

3.2. Data Association by Maximum Likelihood of Measurement

If the measurements only provide a landmark's position information, but the landmark's attribute information is empty, the method presented by Bar-Shalom et al. (Bar-Shalom & Fortmann, 1998) for the data association with innovation will be used here. Their method can be briefly described as follows:

Innovation is the value of difference between measurement *Z(k)* and predicted measurement $\widehat{Z}(k)$, and expressed by $\nu(k)$

$$\nu(k) = Z(k) - \widehat{Z}(k)\,. \tag{14}$$

In order to define a measurement validation region, the innovation needs to be normalized as follows:

$$\varepsilon_v(k) = \nu(k)^T S_v(k)^{-1} \nu(k) \tag{15}$$

where $S_v(k)$ is the innovation covariance matrix, and it is defined as

$$S_v(k) = H(k) P_v(k) H(k)^T + R_v(k) \tag{16}$$

where $P_v(k)$ is the state vector's estimated covariance matrix at step *k*, and $\varepsilon_v(k)$ has a χ^2 distribution with n_z degrees of freedom (n_z is the dimension of the measurement *Z*). The validation technique is based on this innovation. If a measurement is inside a fixed region of a χ^2 distribution, then this observation is accepted; otherwise the observation is rejected.

4. Estimation of Robot Pose and Landmark Positions

In the SLAM problem, a robot pose and landmark positions at time *k*+1 are unknown. They need to be estimated by using input information U, measurement information Z, and robot pose and feature position information, at time *k*. Stacking Equation (4) and Equation (5), the system process model can be expressed as

$$X_s(k+1) = \begin{bmatrix} X_v(k+1) \\ L_1(k+1) \\ L_2(k+1) \\ \vdots \\ L_m(k+1) \end{bmatrix} = \begin{bmatrix} X_v(k+1) \\ L(k+1) \end{bmatrix} = \begin{bmatrix} F(X_v(k), U(k), \mu(k)) + \omega(k) \\ L(k) \end{bmatrix} \tag{17}$$

and the measurement model is

$$Z(k) = H(X_v(k), L(k)) + \eta(k)\,. \tag{18}$$

Both process model (Equation(17)) and measurement model (Equation(18)) are nonlinear equations. A straightforward method to solve this problem is the Extended Kalman Filter (EKF). Due to the high dimensions of the state vector and the need for linearization of the non-linear models, the EKF is not computationally attractive. The particle filter-based fast SLAM approach will be applied to solve this problem.

4.1. Particle Filter

From the view point of probability, the estimation of a robot pose and landmark positions involves computing their posterior probability density function (PDF), $p(X_v(k), L(k) \mid Z(k), X_v(k-1), U(k-1))$, based on the prior probability density function $p(Z(k) \mid X_v(k))$ and $p(X_v(k-1) \mid Z(k-1), X_v(k-2), U(k-2))$, where $X_v(k)$ is the robot state, *L(k)* is the landmark state, Z(k) is the measurement, *U(k)* is the input to the system, at time k. This is the well-known Bayesian approach. According to the definition of a robot model and landmark model in the previous section 3.2, the PDF for a SLAM problem $Bel(X_v(k+1))$ at time *k*+1 can be defined as (Thrun & Burgard, 1998)

$$Bel(X_v(k+1)) = p(X_v(k+1) \mid Z(k+1), X_v(k)), U(k)) . \tag{19}$$

The solution for the SLAM problem is to estimate the maximum of the $Bel(X_v(k+1))$ Bayes' formula can be used on Equation (19) to simplify its implementation.

$$\begin{aligned} Bel(X_v(k+1)) &= \frac{p(Z(k+1) \mid X_v(k+1), X_v(k), U(k))\, p(X_v(k+1) \mid X_v(k), U(k))}{p(Z(k+1) \mid X_v(k), U(k))} \\ &= \xi p(Z(k+1) \mid X_v(k+1), X_v(k), U(k))\, p(X_v(k+1) \mid X_v(k), U(k)) \end{aligned} \tag{20}$$

where ξ is the value of the inverse denominator and is assumed to be a constant. It is known that the measurement *Z(k + 1)* is only dependent on the current pose $X_v(k+1)$ and is not influenced by previous pose $X_v(k)$ and robot movement *U(k).* Therefore, Equation. (20) can be simplified into

$$Bel(X_v(k+1)) = \xi p(Z(k+1) \mid X_v(k+1), X_v(k), U(k))\, p(X_v(k+1) \mid X_v(k), U(k)) \tag{21}$$

By applying the total probability theorem to the second item of the right hand side of Equation (21), then

$$\begin{aligned} Bel(X_v(k+1)) &= \xi p(Z(k+1) \mid X_v(k+1) \\ &\quad \int p(X_v(k+1) \mid X_v(k), U(k))\, p(X_v(k) \mid Z(k), X_v(k-1), U(k-1))\, dX_v(k) \end{aligned} \tag{22}$$

$$Bel(X_v(k+1)) = \xi p(Z(k+1) \mid X_v(k+1) \int p(X_v(k+1) \mid X_v(k), U(k))\, Bel(X_v(k))\, dX_v(k) \tag{23}$$

where $p(Z(k+1) \mid X_v(k+1))$ is the sensor observation model, which can be calculated from Equation (18); $p(X_v(k+1) \mid X_v(k), U(k))$ is the system evolution model, which can be calculated from Equation (17). The integration in the Equation (23) is a difficult challenge to solve the SLAM problem efficiently; therefore, a new algorithm must be designed.

The Monte Carlo based particle filter can be used to overcome the implementation challenge in Equation (23). In the particle filter, $Bel(X_v(k))$ is expressed as a set of particles and every particle is propagated in time according to the state process model such as Equation

(17). The weight of every particle is calculated based on the observation model from the Equation (18). The robot pose and landmark position can be computed from the sum of the weighted samples. The particles should be re-sampled for the next step's estimation. Implementation of a particle filter is summarized in Algorithm.1.

```
Input: Robot movement $U(k)$, sensor measurement $Z(k+1)$ and sample number $N$
Output: Robot pose and features position

initialize state with $p(X_v(0))$
repeat
  for every particle $i$ do
    assign distribution using $p(X_v(k+1) \mid Xv(k),U(k)$
  end for
  for every particle $i$ do
    compute weight $w$, using $p(Z(k+1) \mid X_v(k+1)$
  end for
  calculate robot pose & landmark position from particles & associated weight
  re-sample the particles
until robot stop navigation
```

Alg. 1. Particle filter implementation for robot pose and feature position

Particle filtering can be used for any process and observation models. In the Kalman filter and the extended Kalman filter, the basic requirement is that the error of process model and observation model should be Gaussian distribution. In most cases, this requirement is too restrictive. The particle filter has been called bootstrap filter (Gordon, 1997), condensation (Isard & Blake, 1998), or Monte Carlo filter (Dellaert et al., 1999). In recent years, this method has been successfully used in problems of object tracking (Hue et al., 2002) and mobile robot localization (Dellaert et al., 1999) (Thrun et al., 2001).

4.2. Fast SLAM

Fast SLAM is an approach to separate the SLAM problem into a robot pose and landmark position estimation that is conditioned on the robot pose. The term was first introduced in (Montemerlo et al., 2002). The implementation of FastSLAM is an example of the Rao-Blackwellised particle filter (Doucet et al., 2000) (Murphy, 1999).

```
Input: Robot movement $U\{k)$, sensor measurement $Z(k + 1)$. and sample number $N$
Output: Robot pose and detected features position

initialize state with $p(X_v(0))$
repeat
  for every particle i do
    proposal distribution using $p(X_v(k+1) \mid X_v(k),U(k)$
  end for
  obtain observations $Z(k + 1)$
  data association for the observation data
  for every particle $i$ do
    compute weight using $p(Z(k+1) \mid X_v(k+1))$
  end for
  re-sample the particles
  if current observed feature exists in the map then
    for every particle i do
      for every observed feature do
        update the state of the robot
      end for
    end for
  end if
  if current observed feature is not in the map (new detected features) then
    for every particle $i$ do
      add the new detected features to the map based on the robot pose and observation
      to the features
    end for
  end if
until robot stop navigation
```

Alg. 2. FastSLAM implementation

From the previous definition of a SLAM problem, the system state estimation could be written as

$$Bel(X_v(k+1)) = p(X_v(k+1), L(k+1) \mid Z(k+1), X_v(k)), U(k)) \tag{24}$$

This expression can be factored into two parts according to [8].

$$Bel(X_v(k+1)) = p(X_v(k+1) \mid Z(k+1), X_v(k)), U(k)) \prod_{i=0}^{m} p(L_i(k+1) \mid X_v(k+1), Z(K+1), U(k)) \tag{25}$$

The estimate expression is decomposed into m+1 estimations. One of them is for the robot pose estimation, and m of them are for landmark estimation based on the estimated robot pose. The implementation of FastSLAM is summarized in Algorithm 2. In this thesis, this algorithm is applied for the SLAM problem. The particle filter is implemented to calculate its related conditional densities for robot and landmarks.

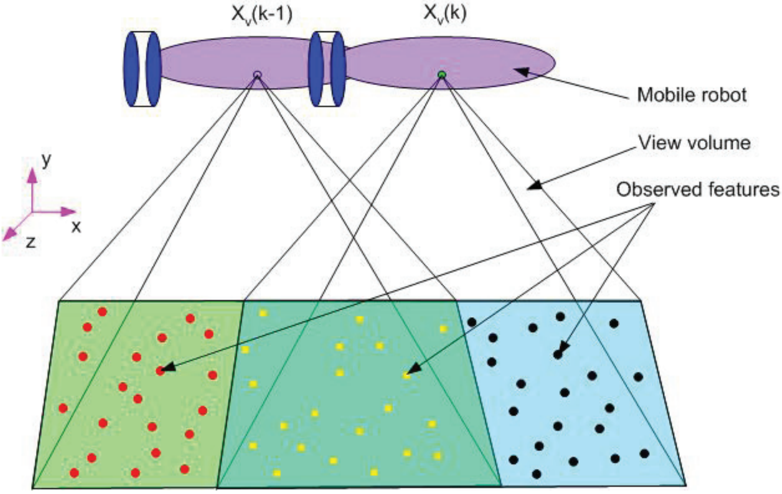


Fig. 6. SLAM implementation with features observed by a sensor.

The idea for FastSLAM can be obtained from Figure 6. We assume that all the observed landmarks by a sensor at robot position $X_v(k-1)$ exist in a map. Then, the observed landmarks at robot position $X_v(k)$ can be divided into two groups. Some of them are already in the map and are labeled as small yellow squares in Figure 6, which are called old landmarks; some of them are new landmarks and do not yet exist in the map and are expressed with small black dots in Figure 6. The measurements from the old landmarks are applied for the robot pose estimation at time k. The measurements from the new landmarks are used to estimate the new landmark positions in the global coordinate system based on the robot position. Then, all the new landmarks are added into the map.

In the fast SLAM approach, the factorizing assumption step turned the high dimension 6 + 3 $\cdot m$ of the SLAM problem into the low dimension (6 and 3) problem's combination, which greatly improves the computation efficiency, but it is assumed that the estimated robot pose has an accurate value. This assumption is not true and will cause errors in the step for the landmark position estimation in a global frame. This is the trade-off between efficiency and accuracy.

Another assumption in the fast SLAM is that the measurement of every detected landmark is independent of the other landmarks in the working area of the robot; therefore, the covariance between two landmarks will be zero. In other methods, such as the EKF or Particle filter, robot pose and all the detected features position are estimated in one state vector, and this may cause the covariance between two landmarks to be other than zero. In most of the cases, these values came from the algorithm design.

5. Multi-sensor Fusion for 3D SLAM

A mobile robot is usually equipped with many sensors which will work together. Fusing the measurements from more than one sensor will provide a more accurate estimation than by using only one sensor's measurement. An example of multi-sensor fusion is shown in Figure 7, where an underwater mobile robot is equipped with a stereo camera and communicated with a set of buoys on the surface.

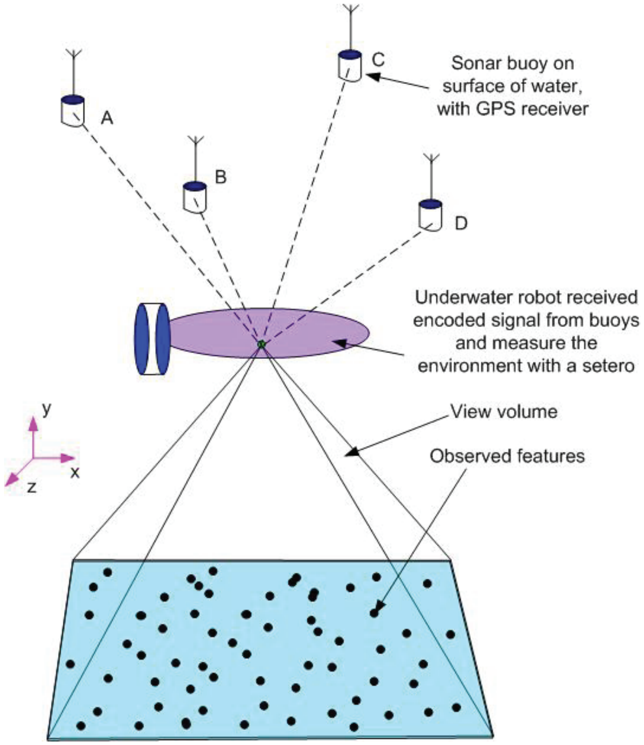

C
Sonar buoy on surface of water, with GPS receiver
A
B
D
Underwater robot received encoded signal from buoys and measure the environment with a setero
y
x
z
View volume
Observed features

Fig. 7. A sensor fusion example for a stereo camera and a set of buoys.

This kind of system structure has two advantages for the underwater robot. The buoys can provide a long range of measurements, while the measurements can be applied to estimate the robot position in a global frame. The stereo camera can provide detailed information of the immediate environment, which will be used for SLAM in a local frame. Integration of both can solve the SLAM problem in a large area for the underwater robot. The buoy system has been proposed using acoustic sensor by (Liu & Milios, 2005).

5.1. Synchronization for Multi-sensor Fusion

All the sensors in a system cannot work at the same speed or frequency, such as in Figure 7. Synchronization for multi-sensor fusion is an important issue. Usually, measurement frequency for each sensor is different, and their measurement time will not coincide. An instance of the estimated robot position from stereo camera and buoys with time stamp is shown in Figure 8. The buoys will provide a long range estimate in a global frame, which has less accumulated errors than that with a camera. Therefore, when the estimate from buoys at time $t_{b,k}$ is obtained, the robot position from buoys at t_k should be estimated by interpolation as

$$X_b(t_k) = \xi_1 X_b(t_{b,k-2}) + \xi_2 X_b(t_{b,k-1}) + \xi_3 X_b(t_{b,k}) \tag{26}$$

where *b* means buoy and k means time stamp; $X_b(t_{b,k-2})$, $X_b(t_{b,k-1})$ and $X_b(t_{b,k})$ are robot position estimated by buoys at time $t_{b,k-2}$, $t_{b,k-1}$ and $t_{b,k}$, respectively. And ξ_1, ξ_2 and ξ_3 are coefficients related to the corresponding time stamps.

$$\xi_1 = \frac{(t_k - t_{b,k-1})(t_k - t_{b,k})}{(t_{b,k-2} - t_{b,k-1})(t_{k,b-2} - t_{b,k})} \tag{27}$$

$$\xi_2 = \frac{(t_k - t_{b,k-2})(t_k - t_{b,k})}{(t_{b,k-1} - t_{b,k-2})(t_{k,b-1} - t_{b,k})} \tag{28}$$

$$\xi_3 = \frac{(t_k - t_{b,k-2})(t_k - t_{b,k-1})}{(t_{b,k-2} - t_{b,k-2})(t_{k,b} - t_{b,k-1})} \tag{29}$$

The uncertainty $P_b(t_k)$ of the position $X_b(t_k)$ from buoys can be obtained from Equation (26) based on the fact that all the estimated robot positions are independent variables. The robot position is estimated by using the measurement between the current robot position and buoys, instead of previous robot position. This is

$$P_b(t_k) = \xi_1^2 P_b(t_{b,k-2}) + \xi_2^2 P_b(t_{b,k-1}) + \xi_3^2 P_b(t_{b,k}) \tag{30}$$

where $P_b(t_{b,k-2})$, $P_b(t_{b,k-1})$ and $P_b(t_{b,k})$ are uncertainties of robot position estimated by buoys at time $t_{b,k-2}$, $t_{b,k-1}$ and $t_{b,k}$, respectively.

The estimated robot position from buoys at time t_k (see Figure 8.) can be obtained by using Equation (26) and Equation (30). At this time, fusing the robot estimation from the stereo camera and buoys is possible.

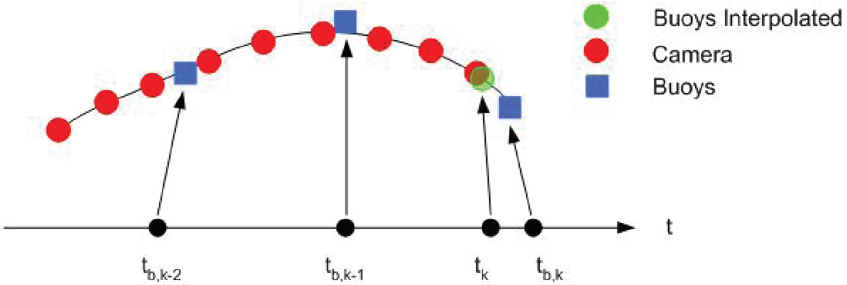


Fig. 8. A stereo camera and buoys estimation with time stamps.

It is very important to ensure that all the sensors installed in the same system have the same time signature. If there are more than two computers working for the same measurement system, it is necessary to calibrate them.
It must be pointed out that three points are used for the interpolation in this section. If there are more than three robot positions, using only three of them is applied for the interpolation; of course, higher order interpolation can be used. If is only two robot positions (points), linear interpolation could be applied.

5.2. General Sensor Fusion Mechanism

All the sensors in the system are applied for one purpose: reliable and accurate robot pose and map estimation. The sensors will not work properly all the time; therefore, a complementary fusion mechanism is designed in this thesis. Figure 9. is a sensor fusion architecture for a stereo camera and a set of buoys.

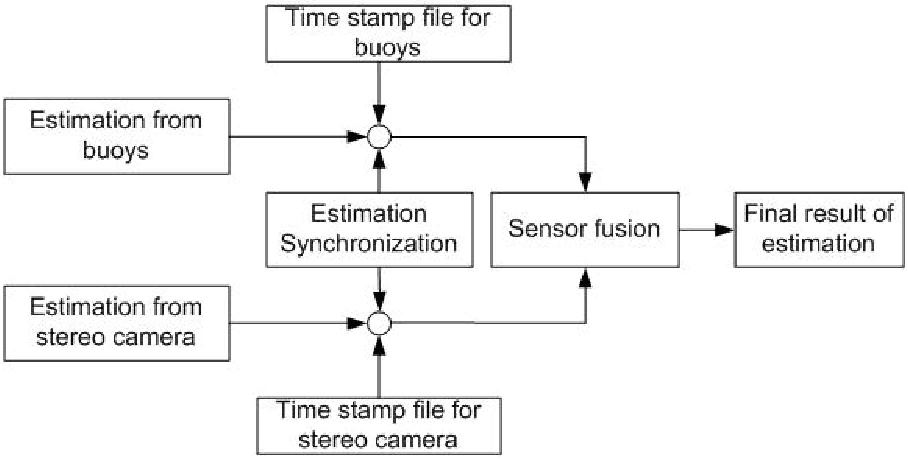

Time stamp file for buoys
Estimation from buoys
Estimation Synchronization
Sensor fusion
Final result of estimation
Estimation from stereo camera
Time stamp file for stereo camera

Fig. 9. Sensor fusion architecture for camera and buoys.

Before sensor fusion is applied, each sensor in the system estimates the robot pose and/or builds a map independently. When the synchronized estimate is obtained, sensor fusion will be performed. Assuming that the error from each sensor follows the Gaussian distribution with zero mean and known covariance, from [1], the fusion is carried out by

$$X_v(t_k) = \frac{P_b(t_k)}{P_b(t_k) + P_c(t_k)} X_c(t_k) + \frac{P_c(t_k)}{P_b(t_k) + P_c(t_k)} X_b(t_k) \tag{31}$$

$$P_v(t_k) = \frac{P_b(t_k) P_c(t_k)}{P_b(t_k) + P_c(t_k)} \tag{32}$$

where $X_c(t_k)$ and $X_b(t_k)$ are the estimated robot pose at time t_k by the stereo camera and buoys, and $P_b(t_k)$ and $P_c(t_k)$ are their associated covariance, respectively.

It must be pointed out that the relative drift of the sensor estimation has a big influence to the sensor fusion result. Ideally, both of the sensor estimation should be overlapped in their estimation uncertainty's boundary. If both of the sensor estimation did not overlap in their uncertainty boundary, the fusion result will not be reliable. To avoid big relative drift of the sensor fusion, we used multi-map mechanism, which means that a new estimation will start as soon as the estimation uncertainty reaches a threshold.

By using this sensor fusion mechanism and system designed for an underwater mobile robot, it is possible to estimate a globally-consistent robot position in a large working area. The following section discusses globally-consistent map building.

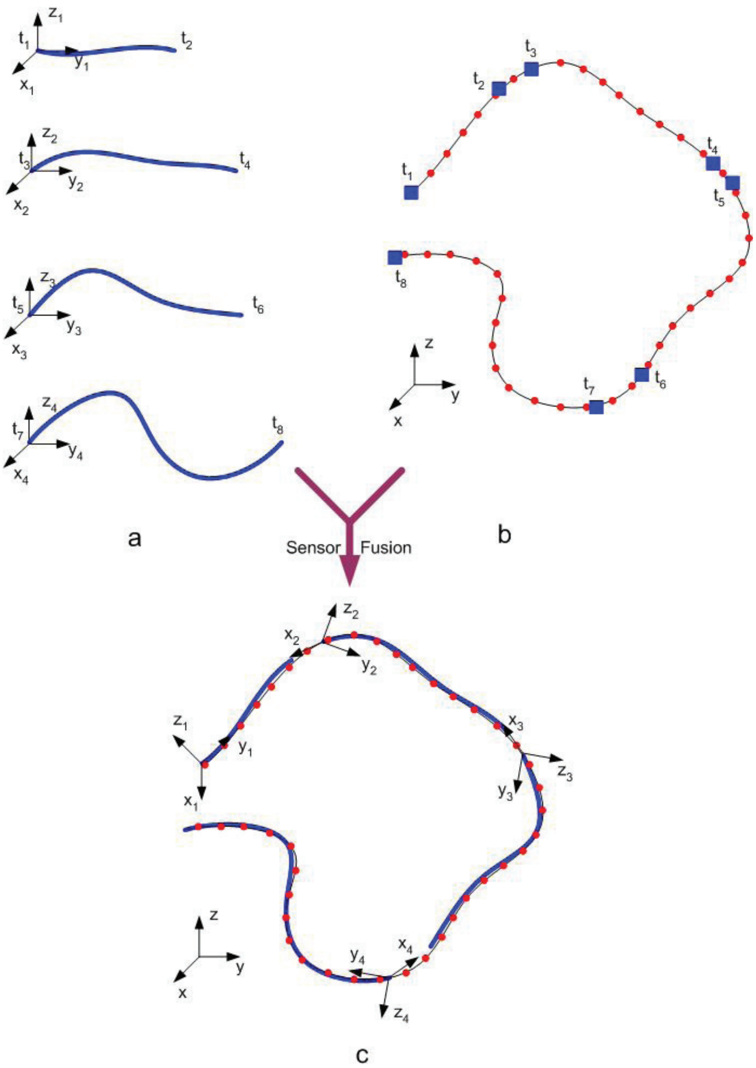


Fig. 10. 3D SLAM by fusion estimation from camera and buoys. (a) Local robot position and map (for simplicity, map is not displayed in this figure) from camera with time stamps. (b) Robot position estimated from buoys with time stamps. (c) Final results after fusing the information from both camera and buoys.

6. Globally-Consistent 3D SLAM for Mobile Robot in Application

The basic requirement for SLAM using stereo vision is that every two consecutive images have to provide enough overlapped features in all the images. In real applications, this requirement may be too strict. If this requirement can not be satisfied, the SLAM process based on image information will be stopped. In this case, even though the robot has another sensor installed (such as a GPS or buoys), it can not obtain a globally-consistent 3D path and a globally-consistent 3D map; therefore, it is necessary to design a new algorithm to solve the 3D SLAM problem in real application.

Assuming that two sensors (stereo camera and buoys or GPS) will be applied, both of which work separately, where the buoys will estimate a globally-consistent robot 3D path and the stereo camera will estimate a set of local maps and the robot's 3D paths in a local coordinate system. This means that when the two consecutive images provide enough overlapped features, the SLAM algorithm begins to work based on the local frames. The estimate results are also stamped with global time. The scenario of this processing is shown in Figure 10. There are four local robot path and associated maps (not shown) based on local frame which is estimated by the stereo camera's measurement with the algorithm of SLAM in Figure 10 (a). For simplicity, the maps are not drawn. These estimated path fragments are time stamped, which will be applied for sensor fusion. For the estimate by buoys measurement in Figure 10(b), the red dots are the robot's position at each time, and the blue squares are the time markers which are corresponding to the time stamps in Figure 10. (a). The final globally-consistent robot path and map are expressed in Figure 10. (c). There are several different steps required to obtain these: (1) robot path parameterizing; (2) robot position association; (3) transformation estimate from local coordinate system to global coordinate system; (4) globally-consistent map integration, and; (5) globally-consistent robot position.

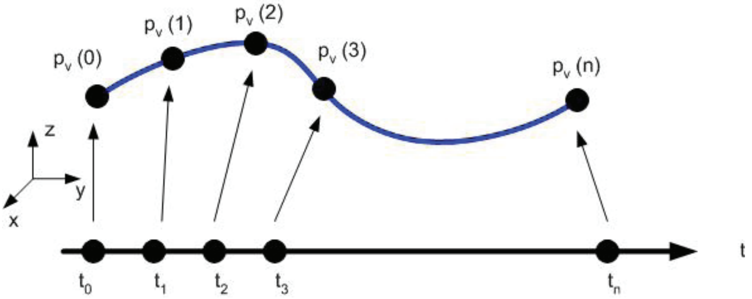


Fig. 11. Robot path parameterizing with B-spline.

6.1. Robot Path Parameterizing

For 3D SLAM, connecting the robot position over time will form a curve in 3D space. A widely-used method to construct a 3D curve to smoothly pass all the discrete position points is the B-spline interpolation (Piegl & Tiller, 1997). Assume that a sequence of robot 3D positions, $p_v(k)$, and its related covariance $P_{vv}(k)$ (k = 0, · · · , n), are obtained from a local coordinate system, and its corresponding time is t_k (Figure 11.). A parametric B-spline of degree 3 to pass these points is defined as

$$S(u)=\sum_{i=0}^{n} p_v(i)N_{i,3}(u) \qquad u\in[0,1] \tag{33}$$

where $N_{i,3}(u)$ are the B-spline basis functions of degree 3, defined with respect to the knot vectors $u = \{u_0, u_{1,} \cdots, u_{n+3+1}\}$, with $u_0 = u_1 = u_2 = u_3 = 0$ and $u_{n+1} = u_{n+2} = u_{n+3} = u_{n+3+1} = 1$, and its parameter *u* at any time can be obtained by

$$u = \frac{t - t_0}{t_n - t_0} \qquad t \in [t_0, t_n] \tag{34}$$

A B-spline base function $N_{i,p}(u)$ of degree p can be calculated by

$$N_{i,0}(u) = \begin{cases} 1 & if\, u_t \le u \le u_{t+1} \\ 0 & any\, others \end{cases} \tag{35}$$

$$N_{i,p}(u) = \frac{u - u_t}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u} N_{i+1,p-1}(u) \tag{36}$$

$$define \quad \frac{0}{0} = 0\,.$$

By using the B-spline function in Equation (33), for any time t between time t_0 and t_n, it is easy to obtain the position of the robot. Its related covariance can be obtained by

$$P_{vv}(u) = \sum_{i=0}^{n} p_v(i)(N_{i,3}(u))^2 \qquad u \in [0,1]\,. \tag{37}$$

6.2. Robot Position Association

From the buoys, the robot position is estimated during its navigation. The positions, $X_b(k)$ (*k* = *0, …, m*), are denoted by small black squares on the red curve in Figure 12, and their corresponding times are shown on the time axis with tb_k (*k* = *0, …, m*). The robot path estimated from the camera is expressed by the blue curve, which is constructed in the previous subsection. With the time information, the corresponding robot position, $X_v^{'}(k)$, based on the local coordinate system($x^{'}y^{'}z^{'}$), can be calculated from the Equation(33) and Equation(37).

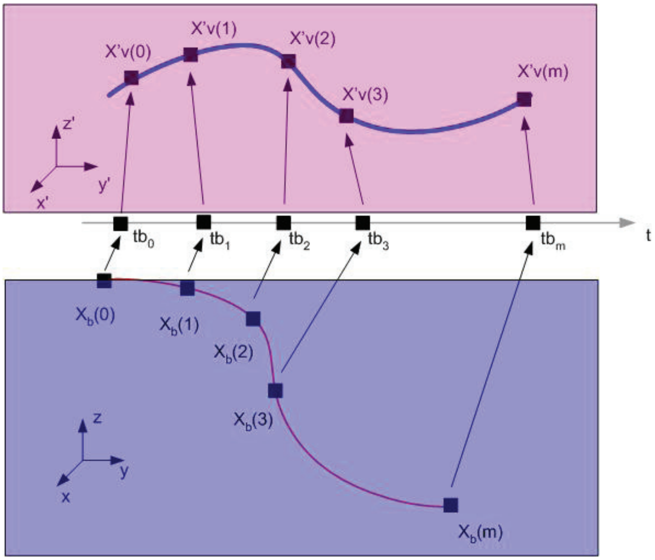


Fig. 12. Robot position association. The top is in local coordinate system, and the bottom is in global coordinate system.

6.3. Transformation between Frames

So far, two sets of corresponding 3D points, $X_b(k)$ and $X_v^{'}(k)$ (*k* = *0,…, m*), are obtained (Note: we only use $X_v^{'}(k)$ position information here since $X_b(k)$ only contain position information). If the number of corresponding points, m, is greater than 3 and all the points in each coordinate system are not in a straight line, it is then possible to estimate the transformation (rotation and translation) between these two data sets. If all the estimated 3D points in one of its local coordinate system are located on a line or are very close to a line, it is impossible to estimate the orientation of this transformation. In this situation, other information such as a robot direction from a digital compass will be needed.

6.4. Globally-Consistent Map Integration

Assuming that the transformation for a local map *k (k = 1, . . . , n)* from a local coordinate system to a global coordinate system is expressed by R_k and T_k, for rotation and translation, respectively, a map established in the local coordinate system can be then transformed to the global coordinate system by

$$L_k^g(i) = R_k L_k^l(i) + T_k \tag{38}$$

where $L_k^g(i)$ is the landmark *i*'s position in the global coordinate system, and $L_k^l(i)$ is the landmark *i*'s position in the local coordinate system *(i = 1,… ,M).*

```
Input: a sequence of images $I_k$, $k$=1,…,$N$, measurements from buoys, and a file of time stamps for each sensor
Output: a globally consistent map $G$ and robot path $p_v$

k=2, numMap=0, flag=new Map_start, Segment=[ ]
repeat
  robot path estimated from the measurements of buoys
  extract SIFT features from image $I_k$, $I_{k-1}$
  if matched features between image $I_{k-1}$ and $I_k$ are more than $m_I$ or the estimated
  position error is less than a certain threshold then
    if flag=new Map_start then
      numMap=numMap+l;
    end if
    SLAM estimation based on stereo camera's measurements
    Segment(k)=numMap
    flag=map_continue
  else
    flag=new Map_start
  end if
until robot stop navigation
robot path 3D curve from buoys estimation by B-spline
for each Segment do
  find corresponding point on 3D path curve
  estimate transformation parameters between local frame and global frame
  global map and path calculation
  integrate 3D map and path
end for
```

Alg. 3. Globally Consistent 3D SLAM based on Sensor Fusion

If there are several different local maps established on one working area, it will be necessary to transform all of them from each local coordinate system to a global coordinate system, and then integrate all of them to form a globally-consistent map.

6.5. Globally-Consistent Robot Position

Due to the bandwidth limitation of the communication or the nature of the sensing, the measurement frequency of the buoys is lower than the stereo camera's frequency. In order to obtain a continuously consistent 3D path, the robot path in a local coordinate system should be transformed to the global coordinate system according to the same method from the previous map transformation.

The algorithm to establish a globally-consistent 3D map is based on the sensor fusion shown in Algorithm 3. An overview of this algorithm is shown in Figure 13. The centre of this diagram belongs to Alg. 3, and the left part for local SLAM is obtained by a camera with the previous FastSLAM, and the global robot position is obtained with the EKF algorithms by buoys and GPS.

In the processing of local SLAM from a camera, there are two issues which need to be considered: estimation error and efficient map. In order to control the estimation error in the local SLAM result, we always check the estimation error for robot path. If the estimated path error grows above a certain threshold, the current local SLAM processing is also stopped, in which case a new local SLAM processing will be started. For efficient map, we will discuss it in the next chapter.

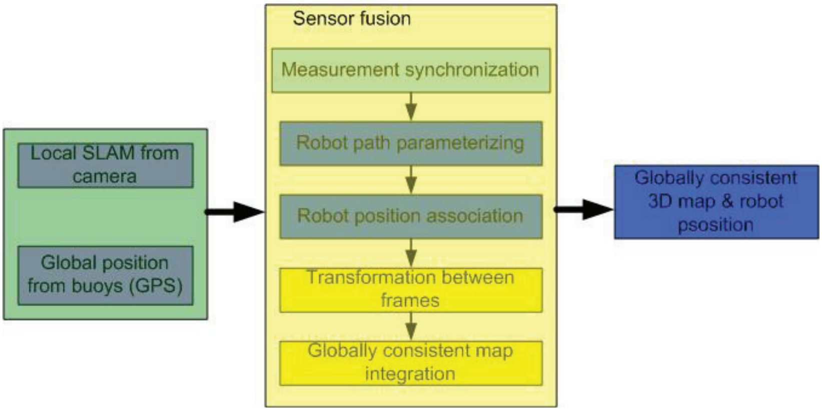

Local SLAM from camera
Global position from buoys (GPS)
Sensor fusion
Measurement synchronization
Robot path parameterizing
Robot position association
Transformation between frames
Globally consistent map integration
Globally consistent 3D map & robot psosition

Fig. 13. Diagram for the globally-consistent 3D SLAM.

7. Summary

This chapter established an approach to solve the full 3D SLAM problem, applied to an underwater environment. First, a general approach to the 3D SLAM problem was presented, which included the models in 3D case, data association and estimation algorithm. For an underwater mobile robot, a new measurement system was designed for large area's globally-consistent SLAM: buoys for long-range estimation, and camera for short-range estimation and map building. Globally-consistent results could be obtained by a complementary sensor fusion mechanism.

By carefully investigating all the algorithms used for SLAM, we designed a new sensor fusion algorithm for large area SLAM, which will be suitable for most of application environment. Two types of sensors are needed for this algorithm: stereo camera (or laser scanner, radar) for local SLAM, and a sensor for robot path in the global coordinate system. Both the local and global paths were expressed by B-spline, and their corresponding 3D points of the associate robot path could be extracted. Transformation (translation and rotation) values from local coordinate system to global coordinate system could be estimated from these matched points, and, the local maps could be integrated to the globally-coordinate system, and a globally-consistent map and path could be obtained.

8. Reference

Bar-Shalom, Y. & Fortmann ,T. E. (1998). Tracking and Data Association. Academic Press

Dellaert, F.; Foxt, D.; W. Burgard, & Thrun, S. (1999). Monte Carlo Localization for Mobile Robots. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, pages 1322–1328, Michigan

Doucet, A., Godsill, S. & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing, 10(3):197–208

Gordon, N. J. (1997). A hybrid bootstrap filter for target tracking in clutter. IEEE Transactions on Aerospace and Electronic Systems, 33:353–358

Hue, C.; Cadre, J. L. & Perez, P. (2002). Tracking multiple objects with particle filtering. IEEE Transactions on Aerospace and Electronic Systems, 38:791–812

Isard, M. & Blake, A. (1998). Condensation-conditional density propagation for visual tracking. Int. J. Computer Vision, 29:5–28

Liu, H. & Milios, E. (2005). Acoustic positioning using multiple microphone arrays. Journal of the Acoustical Society of America, 117(5):2772–2782

Montemerlo, M.; Thrun, S.; Koller, D. & Wegbreit, B. (2002). Fast SLAM: A factored solution to the simultaneous localization and mapping problem. In Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada

Murphy, K. (1999). Bayesian map learning in dynamic environments. In Proceedings (?) Neural Information Processing Systems (NIPS)

Piegl, L. A. & Tiller, W. (1997). The NURBS Book. Springer

Se, S.; Lowe, D. & Little, J. (2003). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. Int. Journal of Robotics Research, 21:735–758

Sonka, M.; Hlavac, V. & Boyle, R. (1999). Image Processing, Analysis, and Machine Vision. PWS Publishing

Thrun, S. & Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robot. Machine Learning, 31:29–53

Thrun, S.; Fox, D.; Burgard, W. & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. Artificial Intelligence, 128:99–141, 2001

9

Flight Dynamics Modelling and Experimental Validation for Unmanned Aerial Vehicles

X.Q. Chen[1], Q. Ou[1], D. R. Wong[1], Y. J. Li[1], M. Sinclair[1], A. Marburg[2]
[1]*University of Canterbury, New Zealand*
[2]*Geospatial Research Centre (NZ) Ltd, New Zealand*

1. Introduction

Unmanned Aerial Vehicles (UAVs) are a viable alternative to manned aircraft and satellites for a variety of applications, including environmental monitoring, agriculture, and surveying. They promise greater precision and much lower operating costs than traditional methods. Critical to the success of UAV systems is the auto-pilot system which keeps the vehicle in the air and in control in the absence of a human pilot. The development of auto-pilot systems for UAVs is an area undergoing intense research. The ability to test autopilot systems in a virtual (software) environment using a software flight dynamics model for UAVs is significant for development.

A reliable UAV simulation process which can be adapted for different aircrafts would provide a platform for developing autopilot systems with reduced dependence on expensive field trials. In many cases, testing newly developed autopilot systems in a virtual environment is the only way to guarantee absolute safety. Additionally the model would allow better repeatability in testing, with controlled flying environments.

Numerical modelling of flight dynamics has a long history in the aerospace industry, and is used in the development of all modern aircraft and satellites. A flight dynamics model is a mathematical representation of the steady state performance and dynamic response that is expected of the proposed vehicle, in this case a UAV (dcb.larc.nasa.gov/Introduction/models.html). The uses of flight dynamics models are diverse. Commercial, military, government organisations and academic sectors employ flight models to achieve their specific tasks (Chavez et al. 2001). Example applications include control algorithms testing, stability and flight characteristics evaluation of preliminary designs, onboard embedded autopilot systems, and onboard Inertial Navigation Systems (INS).

In the development of UAVs and auto-pilot systems, a flight dynamics model for flight simulations allows rapid and safe testing on a computer. However, a software model developed from first principles has unknown accuracy. For such a model to be of real use, its development process is necessary to include implementation, verification and validation. The approach for each stage of the development process is presented in this chapter.

2. Brief Review of Flight Model Implementation Methods

The fundamental goal of flight dynamics modelling is to represent the flight motion numerically for a given input, as close to the flight motion in the real world as the application requires. To accommodate a wide range of applications, various implementations of flight models in terms of assumptions and algorithms therefore exist. Nevertheless, all flight dynamics models are based on the mathematics model derived from Newtonian Physics. From Newton's second law, an aircraft's motion in its six degrees of freedom (DOF) can be described by a system of non-linear first order differential equations. These equations of motion served as the fundamentals for almost all flight dynamics models. With today's computing power, the processing time of solving these equations is trivial comparing to other signal processing algorithms (e.g. Kalman filter) that might be implemented as part of the flight model. A number of popular numerical techniques used to solve these non-linear systems range from highly efficient to precise but computational intensive: Euler, Heun, Bogacki-shampine, Runge-kutta and Dormand-Prince. As flight modelling becomes ever more sophisticated and more application specific, they are implemented to a vast number of different flavours.

In flight modelling, non-linear model is often used in personal computers where resources are abundant and speed is not a concern. It has been recognized that the significant improvements of dynamic performance of current and new generation of advanced airplanes is possible if flight systems design integrates nonlinear analysis, control, and identification (Lyshevski, 1997). There were several UAV flight control system design projects applied non-linear flight models to simulate the dynamic behaviour of their vehicles Buschmann et al., 2004; Guglieri et al., 2006). On the other hand, the non-linear mathematical model can be linearized to simplify the computation (Ye et al., 2006). In many applications, where the linearized model is considered 'close enough', the required computing power is significantly reduced. For instance, linearized model is widely used in embedded system of micro UAVs, with a wing span of less than 50 cm, to compromise the limited processing power of the onboard computer (Jackowski et al., 2004).

A flight dynamics model may also employ one of the two common orientation representations – Euler angle and quaternion. The most common way to represent the attitude of an aircraft is a set of three Euler angles. These are popular because they are easy to understand and easy to use. However, the main disadvantages of Euler angles are: (1) that certain important functions of Euler angles have singularities, and (2) that they are less accurate than unit quaternions when used to integrate incremental changes in attitude over time (Diebel, 2006). Nowadays, quaternion has been increasingly adapted in flight models because it is better able to avoid singularities and high data rates associated in Euler angle representation (Cooke et al., 1994).

For projects that require a generic flight model, there are some options available. Representative existing models available to the public domain are X-plane (Laminar Research), JSBSim (Berndt J.) and Flight Simulator X (Microsoft). Among these, only JSBSim is open source software, whereas the other two simulation programs are like a 'black box' which does not allow users to access the internal of the models. JSBSim is a 6-DOF non-linear flight model that adopts quaternion angle representation. The open source feature of JSBsim has gained a lot of attention from researchers because of the significant cost saving. A full 6-DOF simulator for flight simulation and pilot training was constructed at the University of Naples using JSBSim as its physics engine (Marco, 2006).

In most projects where time and budget permit, researchers favour to build their own flight models in Matlab and Simulink, or using other generic computer languages like C++ or C for embedded systems (Rasmussen & Chandler, 2002; Berndt, 2004; Buschmann et al., 2004; Jordan et al. 2006) . Apparently building a model from scratch is the best way to have full control over the model, and hence can effectively customize the model to meet their specific applications. Implementing a flight model in Matlab is generally the first step towards development. Matlab provides comprehensive modelling tools that allow concepts to be quickly validated and fine-tuned for improvements. Coding the flight model in C++ or C is normally the final stage when the accuracy of the model is validated and the performance has become a primary concern. The focus of flight model implementation here is on a rapid development approach by integrating Datcom, FlightGear with Matlab Simulink.

3. Integrated Flight Dynamics Modelling Methodology

3.1 Modelling Approach

The first stage of the work was focused on the development of a generic flight dynamics model (FDM). The role of the FDM in the whole simulation is a physics engine that process parameters from all input information. By manipulating input variables mathematically, an FDM predicts the future states of an aircraft. The generic FDM was developed in Matlab Simulink, using the Aerospace Blockset.

Aerodynamic coefficients (AC) characterize the response of the proposed vehicle based on its geometry. With a generic FDM implementation in mind, the ACs are not provided by the FDM and hence need to be determined by other means. As long as the aerodynamic coefficients are available, the FDM models the motion of any vehicle configuration, from a ball (potentially for debugging), to a transonic fighter. The FDM, like any other dynamics model, is a data driven program. Hence the accuracy of its outputs depends on the quality of the input information supplied. The FDM takes initial conditions of the vehicle, and other inputs including aircraft properties (e.g. inertia and gravity), aerodynamic coefficients, control inputs and relative wind conditions. It then outputs the vehicle dynamic responses. Fig. 1. illustrates the internal data flow of the FDM.

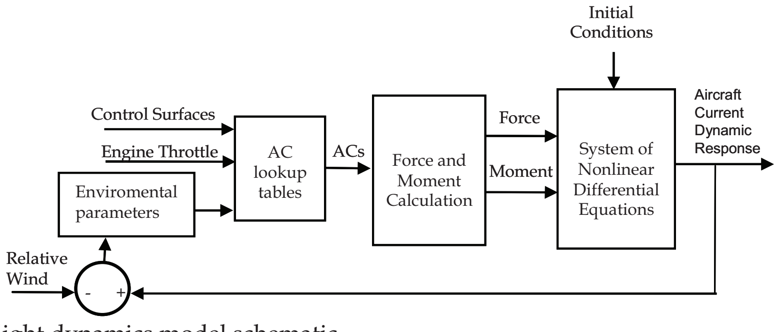


Fig. 1. Flight dynamics model schematic

To establish a model for a particular UAV, the determination of all UAV's aerodynamic coefficients is not a trivial task. Two methods to determine the coefficients are: i) experimental approach, and ii) mathematical approach. The former is generally very accurate, but time consuming and expensive. On the other hand, the mathematical approach is much faster, with little cost and very repeatable by the aid of computers. However, the inaccuracy in the mathematical approach is inevitably caused by the complex aerodynamics involved in aircraft modelling, and the uncertainty in the real environment conditions. The best compromise is to use a combination of these two methods, with the static ACs determined in a wind tunnel and the derivative ACs determined by software.

This work used the software Datcom (Galbraith, 2004) to calculate aerodynamic coefficients from first principles. By writing an input file containing all essential geometries of an aircraft, Datcom produces an output file with aerodynamic coefficients. The interface to Matlab is achieved by a Matlab script file that loads all the essential aerodynamic coefficients from the Datcom output file to the Matlab Workspace. The coefficients in the six degrees of freedom are drag, lift, side, pitching moment, rolling moment, and yawing moment coefficient. By interfacing Datcom with the FDM in the frontend, an aircraft model for any fix wing UAVs can be rapidly developed without wind tunnel testing. This feature significantly increases the repeatability of flight simulation and is found very useful for UAV preliminary designs, where only a rough estimate of the vehicle's stability is required.

Data visualization is another aspect considered while building the flight dynamics model. This has led to interfacing the FDM with an open source flight simulation program FlightGear (Barr, 2006), which can produce a 3D graphic animation in real time. The animation facility allows the UAV to be viewed from any angles, and provides absolute visual information on the UAV attitude and stability. With the FlightGear added in the back-end, the complete simulation environment with the FDM as the physics engine is shown in Fig. 2. This integrated flight dynamics model was developed in the earlier work (Ou, 2008).

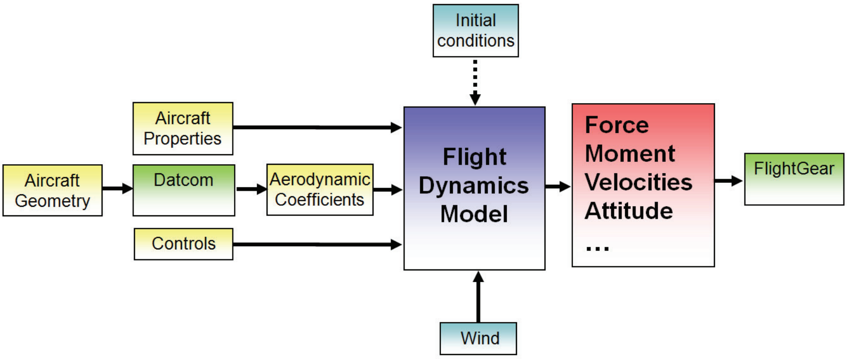


Fig. 2. Block diagram of integrated flight dynamics modelling

3.2 Mathematical Model

Depending on the tool used, the knowledge of underlying mathematics is not always necessary for constructing a flight model. For example, the presented approach in the previous section was to build the model using Aerospace Blockset in Matlab Simulink, where nearly all the functional blocks for rapid aircraft model development are provided. However, in applications that require onboard processing, the understanding of the mathematical model becomes prerequisite. With a mathematical model, the FDM can be implemented in any languages (VHDL, assembly or C language) for embedded systems. Furthermore, transfer functions can be derived from the mathematical model to optimize auto-pilot control strategies.

Essentially, a dynamics flight model can be represented as a system of first order non-linear Differential Equations (DEs). The simulation results are given by solving the system for all the state variables with respect to time. Traditionally, the system of DEs is established based on body coordinates. Response in other coordinate system of interest can be obtained by transformation matrices (e.g. Direction Cosine Matrix). A 6-DOF flight model consists of six fundamental state variables: *u v w* (body velocities), and *p q r* (body angular velocities).

Aircraft properties include aircraft's mass, moment of inertia and centre of gravity. For UAV modelling, it can be assumed the aircraft has constant mass over a flight. The moment of inertia along three axes is a 3×3 matrix denoted:

$$I = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{pmatrix}$$

For traditional aircraft, symmetry and uniformly distributed mass about the x-z plane can be assumed. As a result, the products of inertia $I_{xy} = I_{yx} = I_{zy} = I_{yz} = 0$. Since I_{xz} and I_{zx} are generally very much smaller than I_x, I_y and I_z, a further simplification can be made by neglecting them, so $I_{xz} = I_{zx} = 0$. This assumption is a very satisfactory approximation for UAV models. The accelerations in x, y and z-axis can be expressed in the following equations (Cook, 2007):

$$\dot{u} = rv - qw + \sum F_x / m \quad (1)$$

$$\dot{v} = pw - ru + \sum F_y / m \quad (2)$$

$$\dot{w} = qu - pv + \sum F_z / m \quad (3)$$

Where *r, p, q* are angular velocities about body coordinates x-, y-, and z-axis respectively. F_x, F_y, F_z are force in x-, y-, and z-axis respectively. *m* is the mass of the vehicle. The angular accelerations about x-, y- and z-axis can be expressed as (Cook 2007):

$$\dot{p} = \left[\sum M_x + (I_y - I_z)qr\right] / I_x \quad (4)$$

$$\dot{q} = \left[\sum M_y + (I_x - I_z)pr\right] / I_y \quad (5)$$

$$\dot{r} = \left[\sum M_z + (I_x - I_y)pq\right] / I_z \quad (6)$$

Transformation from body velocities to earth-fixed reference frame velocities is achieved by multiplying the inverse of Direction Cosine Matrix (DCM) (The MathWorks, Aerospace Blockset 3 User Guide).

$$\begin{pmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{Z}_e \end{pmatrix} = DCM^{-1} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

Where X_e, Y_e, Z_e are x-, y-, z-coordinates in the earth-fixed reference frame, and

$$DCM^{-1} = \begin{pmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix}$$

Where ϕ, θ, ψ are rolling, pitching and yawing angle in earth-fixed reference frame respectively.
Hence, the body velocities to earth-fixed reference frame velocities can be obtained:

$$\dot{X}_e = u\cos\psi\cos\theta + v(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) + w(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi) \quad (7)$$

$$\dot{Y}_e = u\sin\psi\cos\theta + v(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) + w(\sin\psi\sin\theta\cos\phi - \cos\phi\sin\phi) \quad (8)$$

$$\dot{Z}_e = -u\sin\theta + v\cos\theta\sin\phi + w\cos\theta\cos\phi \quad (9)$$

The rates of rolling, pitching and jawing angles are given as follows (The MathWorks, Aerospace Blockset 3 User Guide):

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\sin\theta / \cos\theta \quad (10)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad (11)$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)/\cos\theta \quad (12)$$

The angle of attack α is related to the body velocity u (in x-axis) and w (in z-axis):

$$\tan\alpha = \frac{w}{u}$$

Differentiating both sides with respect to time, and re-arranging the equation to obtain the rate of angle of attack:

$$\dot{\alpha} = \frac{\dot{w}u - w\dot{u}}{u^2}\cos^2\alpha \quad (13)$$

The sideslip angle β has the following relationship with the body velocities:

$$\sin\beta = \frac{v}{\sqrt{v^2 + u^2 + w^2}}$$

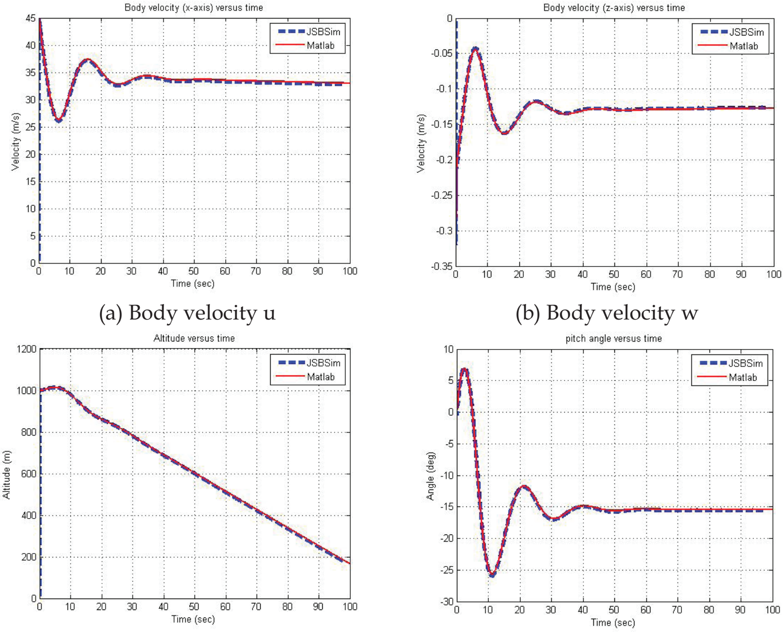
By differentiation, the rate of sideslip angle is obtained:

$$\dot{\beta} = \left[\dot{v}\left(u^2 + v^2 + w^2\right)^{-1/2} - \frac{1}{2}v\left(2u\dot{u} + 2v\dot{v} + 2w\dot{w}\right)\left(v^2 + u^2 + w^2\right)^{-3/2} \right] \frac{1}{\cos\beta} \quad (14)$$

The above ordinary differential equations (1 to 14) define a general aircraft mathematical model. The sum of forces and moments are not specified, because they vary according to aircraft geometries. Since forces and moments are obtained from aerodynamic coefficients, the summation terms are actually functions of flight conditions and aircraft state. Due to the inherent complexity of aerodynamics, the forces and moments are generally defined by lookup tables, which was the case throughout this project.

3.3 Flight Dynamics Model Verification

The FDM was verified by comparing its results with another FDM with exactly the same input information. The open source JSBSim model was chosen to make the comparison, because it is generally considered a very accurate FDM (Berndt, 2004). In the comparison, propulsion and control surface deflection angles inputs were set to be zero in both flight models. While this simplified the XML aircraft specification file required in JSBSim, it does not affect the comparison results because of the identical inputs in both FDMs. Fig. 3 shows the comparison results of body velocity u and w, altitude, and pitch angle for both FDMs.



(a) Body velocity u

(b) Body velocity w

(c) Altitude

(d) Pitch angle

Fig. 3. Verification of simulation results against JSBSim

The results verify that that the outputs produced by the developed flight dynamics model are consistent with the outputs of JSBSim. A new model was developed rather than using JSBSim because the Matlab interface is much easier to use and is more versatile than the complex JSBSim XML configuration system. The new FDM also allows rapid UAV simulation that is difficult in JSBSim.

3.4 Flight Dynamics Model Validation

To test and validate the model against real flights, an electronic system is employed for collecting all required flight data. Then the response of the real aircraft to environmental and control inputs can be compared to the response of the software model when the same input parameters are applied. This allows the flight model to be changed or fine tuned as required to prepare for reliable and representative virtual testing.

Fig. 4. shows an overview of the validation process. In the validation process, the flight simulation model requires wind data and servo motor input pulses to be recorded while the plane is in flight. Other electronic equipment includes GPS system (www.trimble.com), inertial navigation sensors (www.xbow.com) and data loggers (www.grcnz.com). This project focused on the Kuruwhengi series of UAVs, which includes the gas powered K100 with a 2.4 m wingspan and the K70 which is a 70% scaled down version.

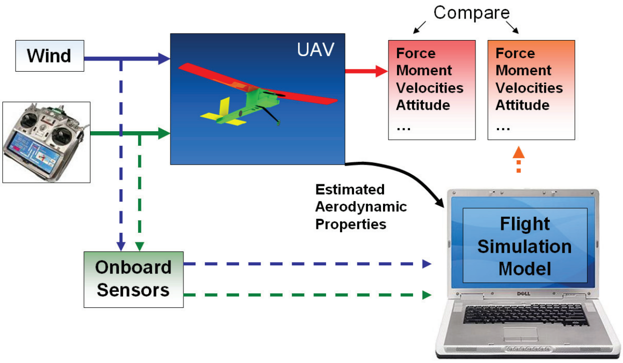


Fig. 4. Flight dynamic model validation process

The model also requires the aerodynamics coefficients and other mechanical properties of the K100 UAV. With all collected flight data from flight tests, the simulation model can be run with the same control inputs and wind conditions and the UAV behaviour compared. The major discrepancies between the two can then be identified so that changes to the simulation model for better accuracy can be made.

4. Development of Wind Speed Sensor and Data Acquisition System

The wind speed sensor was a key part of the validation project. The outcome of the project depended on the ability to measure the body velocities of the UAV in flight tests. This section presents the completion of the wind speed sensor design.

4.1 Wind Speed Sensor Requirements

The purpose of the wind speed sensor is to measure the Angle of Attack (AOA) and Angle of Sideslip (AOS) of the UAV during a flight. The data depend heavily on the wind conditions as well as the velocity and orientation of the aircraft. The requirements for the wind speed sensor design were identified as follows:

Must be light and portable to fit on the UAV.

- Must measure wind speed in three axes in order to determine angle of attack, angle of sideslip, and the longitudinal wind velocity along the body of the UAV.
- Price and complexity should be kept to a minimum.

Extensive research was carried out to determine the best method for measuring wind speed in three axes. Three main alternatives were identified and researched; namely, using ultrasound, a Pitot static system, and optical methods.

For this project the Pitot static system was chosen because of its compactness for fitting to a small aircraft and low cost. Pitot static systems work by measuring the pressure change associated with a moving volume of air. Pitot probes are used on commercial remote controlled aircraft for measuring longitudinal wind speed along the aircraft (Grasmeyer & Keennon, 2001). There are a few systems which measure angle of attack and angle of sideslip using a probe with five pressure ports at the tip (Porro, 2001; Marco, 2006). By measuring the differential pressure between opposing ports, the angle of attack, sideslip, and longitudinal wind speeds can be determined, and subsequently can be used directly to calculate a three dimensional wind vector.

4.2 Sensor System Design

The design of the wind speed sensor was split into three tasks: the mechanical design (casing), the electronics hardware, and the software design. An overview of the system is shown below in Fig. 5. The final design is a robust and compact system which is mounted in front of the leading edge of the UAV on a fibreglass rod, as shown in Fig. 6.

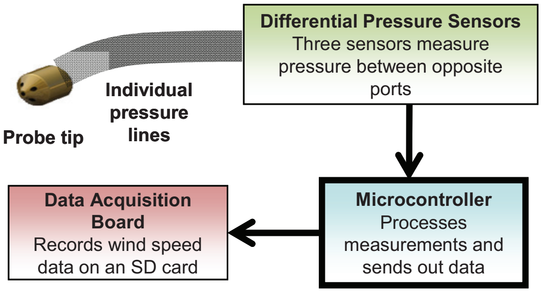

Differential Pressure Sensors
Three sensors measure pressure between opposite ports
Individual pressure lines
Probe tip
Microcontroller
Processes measurements and sends out data
Data Acquisition Board
Records wind speed data on an SD card

Fig. 5. Wind speed sensor design overview

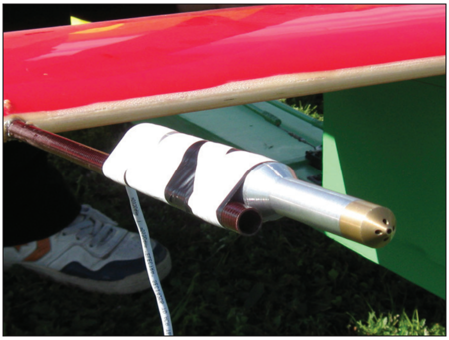


Fig. 6. Wind speed sensor mounting

4.3 Sensor Casing Design

The wind speed sensor probe casing was required to be reasonably light, and it was decided that all of the pressure sensors and electronics should be housed within the probe casing to avoid long pressure lines. The UAV could easily crash during testing, so the strength and ease of repair and assembly of the probe were also considerations. The material of the body is aluminium and the tip is made out of brass for high precision and better surface finish. The bigger diameter section of the probe body is for housing the sensor's printed circuit board (PCB). The brass probe tip is threaded onto the body so that it can be taken off the probe for ease of assembly. This also provides options to try different tip designs without changing the probe body. The tip is machined with a conical shape with a 45^0 angle to give a detecting angle range of ± 45^0. Plastic pipes connect the internal pressure sensors with the five pressure ports. The casing design shown in Fig. 7 was modelled in SolidWorks to produce drawings for fabrication and to check for the internal spacing with a PCB 3D model.

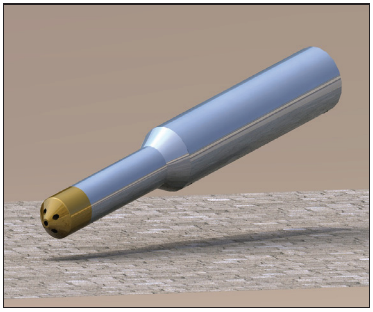


Fig. 7. Wind speed sensor casing modelling

4.4 Sensor Electronic Design

Based on first principle, the relationship between airspeed and change in pressure is given by Equation. (15). The flight speed range of a UAV is assumed to be from 30kph to 120kph, which will give a pressure difference of 42Pa and 665Pa respectively. These results showed that a sensor range from -2kPa to 2kPa is suitable for this application.

$$V = \sqrt{\frac{2RT}{M}\frac{P_\Delta}{P_s}} \tag{15}$$

Where V = Airspeed, R = Gas constant, T = Temperature, M = Molecular mass constant, P_Δ = Change in pressure, and P_s – Standard atmosphere pressure

Based on these pressure ranges, pressure sensors were chosen for the application. Three Freescale semiconductor piezoresistive pressure transducers are used (MPX7002DP); one measures the difference between vertical pressure ports (AOA), one the difference between horizontal pressure ports (AOS) and the other one the difference between the stagnant pressure in the central port and the static pressure inside the probe body (longitudinal wind velocity). These sensors have built in amplification circuitry and supply a 0 to 5V signal (which is 2.5V when the ports are at the same pressure). An Atmel ATMega 168 microcontroller receives the three analogue signals and converts the readings to a digital value by way of the built-in Analogue to Digital Converter (ADC). The digitalized values are sent out to the data acquisition board by way of a Serial Peripheral Interface (SPI). This interface is not typically used for long transmission lengths, but was tested with a 2 metres Cat-5 cable with no transmission errors. The PCB was modelled before manufacture and assembly to test the fit in the casing as shown in . Software development for the microcontroller was completed in C language using the WinAVR toolchain.

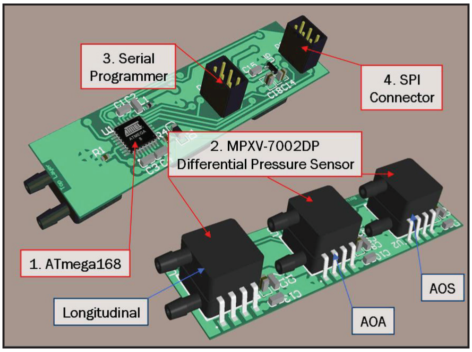


Fig. 8. Wind speed sensor PCB modelling

4.5 Sensor Calibration

The completed wind speed sensor required calibration so that the raw ADC values produced by the probe could be correlated to an actual wind vector. The closed high speed wind tunnel in the Mechanical Engineering Department at The University of Canterbury was used for the calibration. The wind speed sensor probe was mounted on a sting for accurate angle variation, as shown in . The wind tunnel was run at a range of wind speeds, and the AOA and AOS of the probe were varied and the ADC logged using a development board for a UART interface to a PC.

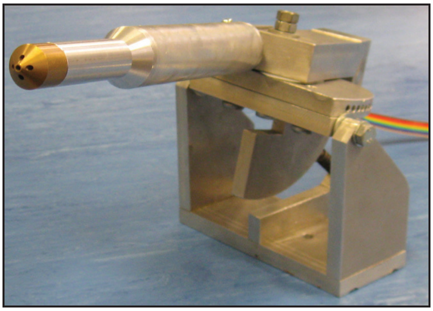


Fig. 9. Mount for wind tunnel calibration

The calibration results showed the inter-dependency of the two differential port ADC values and the central port ADC value at different speeds and angles. This is caused by the pressure drop in the central port as the angle of the probe with respect to the oncoming wind velocity increase. The inter-dependencies made generating an equation for converting ADC values into angles and velocities an extremely complex task. A sample of the calibration results for the AOA is shown in Fig. 10.

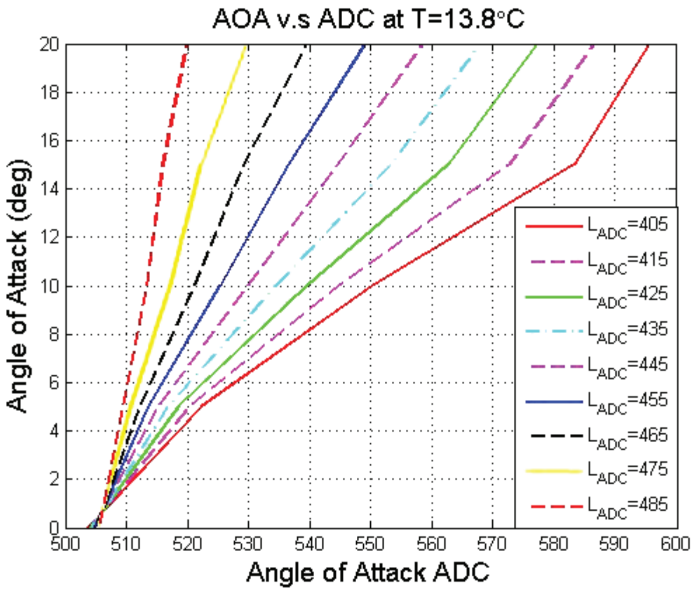


Fig. 10. Sample AOA calibration results

The calibration data were used to create a 2D look-up table for converting ADC values to angles and velocities. Matlab was used to create and use the tables, which also use a linear approximation between data points to allow reasonable estimation of all angles and velocities. The accuracy of the look-up method increases with more data points so running a calibration with as many data points as possible may increase the accuracy of the wind speed sensor significantly.

4.6 Data Acquisition Board

The Data Acquisition Board (DAB) measures the servo motor input pulse timing, which can be converted into an angle of deflection of the corresponding control surface/throttle position. The DAB also communicates with the wind speed sensor. It organizes all the data and sends them for logging by the Data Logger over a UART interface. Fig. 11 shows the functional block diagram of DAB interfacing with the other system components.

The DAB was required to achieve the following:

- Be powered off a power supply ranging from 5 to 12 volts.
- Recognize and read up to 8 servo motor channel pulses.
- Act as a gateway for the wind speed sensor and the Data Logger by communicating with the wind speed sensor via Serial Peripheral Interface (SPI) and the Data Logger via Universal Asynchronous Receiver Transmitter (UART).
- Interleave the servo signals and wind speed data; format the information and output the data via UART through an RS-232 serial interface.

Given the design requirements, the main consideration in the design process was component selection. Research was done into suitable hardware components and circuits. One circuit design consideration was whether or not to use a multiplexer to read the servo lines. The potential advantage of this approach was to save I/O pins. In addition, the software would be easier to develop because the task of interleaving the signals would be performed by hardware. The drawback was that the servos could only be connected to specific headers. This is because the RC receiver outputs control pulses one at a time to each servo sequentially. If the servo order is not specified and correctly arranged, no servo pulse may be recorded. Since only few I/O pins are required in this embedded application, it was decided to measure each servo channel directly by an I/O pin using external interrupt. This software multiplexing approach was able to deliver a more robust system.

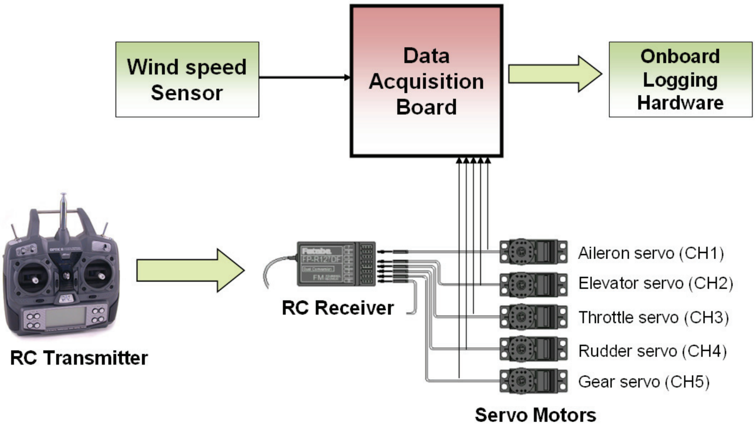


Fig. 11. Data acquisition board block diagram

The AVR ATmega168 was chosen for the main processor. This microcontroller is the same as the one used for the wind speed sensor. This simplified programming and interfacing between the two systems. The ATmega168 provides sufficient IO (input/output) and communication peripherals which are required for the Data Acquisition Board design - specifically UART and SPI.

The DAB taps off the servo control lines to read the signals. One concern was that this could affect the servo behaviour, which can potentially result in crashing the UAV. To ensure that any effect is minimal, the servo input circuitry has been designed for very high impedance. The CMOS CD4050B logic level converters from Texas Instruments were chosen to buffer the servo signals and their purpose is two-fold. Firstly, CMOS devices inherently have very high input impedance and this provides the necessary interference immunity for the servo signals. Secondly they perform the task of buffering the servo control readings to a suitable voltage level for the Atmega168 to read, allowing for a servo signal of up to 15V (while the Atmega168 is only 6V tolerant).

The final design was a success and performed to all the design specifications. It was able to reliably relay the servo signals and wind speed data via UART both during PC tests and in actual flight tests. Fig. 12. shows the final Data Acquisition Board.

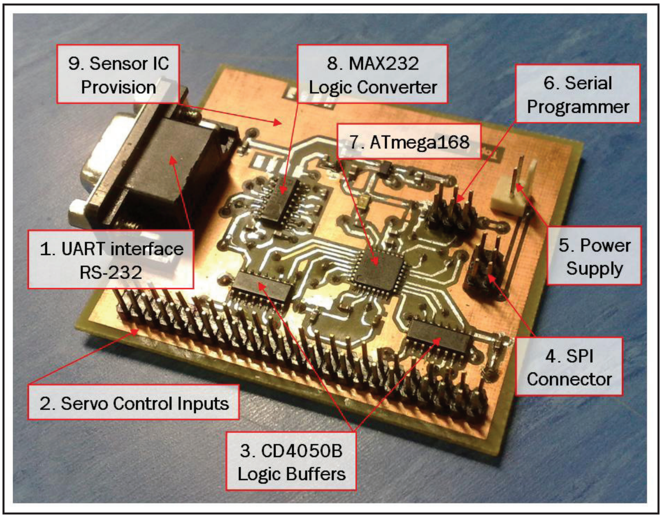


Fig. 12. Data acquisition board

A MAX-232 logic level converter was necessary to allow communication with a PC using the RS-232 standard. Although the Data Logger can handle the default logic levels of the ATmega168 IC, interfacing with a PC was required for development and testing. The MAX-232 was capable of buffering the CMOS logic levels to the necessary ±12 volt.

Noise immunity for analog pins from any possible high frequency noise was considered. The design has incorporated separate analog and digital grounds for analog and digital signals to limit the coupling between the two signals. Although there are no analog signals on the Data Acquisition Board in the current configuration, the final design incorporates footprints for the same pressure sensor IC as used in the wind speed sensor. This could provide extra functionality in development and testing. The data acquisition board is located within the body of the UAV so that the pressure sensor could be used to monitor internal 'cabin pressure', or provide a static pressure reference to compare with the static pressure in the wind speed probe.

5. Parameterisation of UAV

As discussed previously, the FDM requires the basic physical properties of the UAV to run an accurate simulation. The measurement of the K100 mechanical properties and aerodynamic coefficients is discussed in this section.

5.1 Geometric Properties of the K100 UAV

The geometry of the K100 UAV was carefully measured. The measurements were used to determine the UAV's aerodynamic coefficients using software packages. An accurate 3D model of the K100 UAV and engineering drawings were produced in SolidWorks.

The other important measurements required by the FDM are the moment of inertia along each axis, and the location of the centre of gravity. The 'compound pendulum' and 'multiple pendulums' experiments (Blaine, 1996) were carried out on the K100 to obtain the radii of gyration in each axis. These methods are commonly used on radio controlled airplanes and missiles. During the experiment, the periods of the plane swung on each axis were obtained, as well as some relevant lengths of the setup that were required for calculating the radii of gyration. Using this information as inputs to the software program Plane Geometry (Blaine, 1996), the moment of inertia along axis was determined. Table 1 summarizes the results.

	Pitch	Roll	Yaw
Period (s)	2.21	2.26	1.49
L (mm)	1027	1027	460
R (mm)	-	-	920
Radius of Gyration (m)	0.437134	0.498094	0.50368
Mass (Kg)	5	5	5
Moment of inertia (kgm2)	0.9554307	1.240488	1.26848

Table 1: Pendulum experiment results

5.2 Software Aerodynamics Coefficients

Two software packages, Datcom (Galbraith, 2004) and Tornado (www.redhammer.se/tornado) were used to estimate the aerodynamic coefficients of the K100, based on the geometric properties that were measured. Both packages perform computational fluid dynamics (CFD) calculations in order to produce their coefficient values. The CFD calculations use simplified geometry based on basic measurements only, so they do not give extremely realistic coefficient values.

Wind Tunnel Testing was required to compare the software aerodynamic coefficients with results found experimentally. Using the experimental results it was possible to determine which of the software packages produces better results for a particular coefficient and how much it differs from the experimental results. From these results it can be determined whether or not the CFD software can be used in future if aerodynamic coefficients for another UAV are to be obtained. If the software and experimental results agree well, coefficients can be accurately determined without use of a wind tunnel. Wind tunnel testing is time consuming and requires access to a wind tunnel facility, so it should be avoided if possible.

5.3 Wind Tunnel Testing

The Department of Mechanical Engineering, the University of Canterbury, has a large open wind tunnel that can be used with relatively large models. This tunnel has a 1500 mm wide nozzle that can produce 80 kph peak wind speeds. The K70 UAV (70% scale down of K100) has 1600 mm wide wings, which is slightly wider than the nozzle airflow width. Despite this, the K70 could be used in the open wind tunnel with the narrow nozzle without causing major inaccuracies. The one significant source of error when using the open wind tunnel is the limitation of its airflow speed and this had to be considered when analysing the results.

A wind tunnel test of the K70 UAV was carried out in order to determine the aerodynamic coefficients of the vehicle. These were used for inputs to the Flight Dynamic Model directly. They were also interesting for the comparison of different software packages for determining aerodynamic coefficients.

In preparation for wind tunnel testing a sting mount set up for mounting the K70 UAV was designed in SolidWorks, shown in . The mount was based on an existing sting (used in a previous wind tunnel experiment) which was fastened to a U-shaped clamp with an M12 bolt. Plates above and below where the UAV was positioned were fastened onto the clamp using M6 bolts. The UAV was clamped to the sting because it could not be drilled or modified in any other way. The sting was bolted to a force plate using a series of 3 mm OD super screws in order to minimise the damage to the force plate. The force place measured the forces and moments that the sting was subjected to during testing using a series of strain gauges. The wind tunnel setup and the UAV under test are shown in .

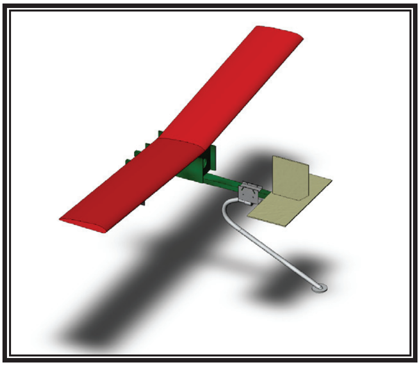


Fig. 13. SolidWorks UAV and sting mount

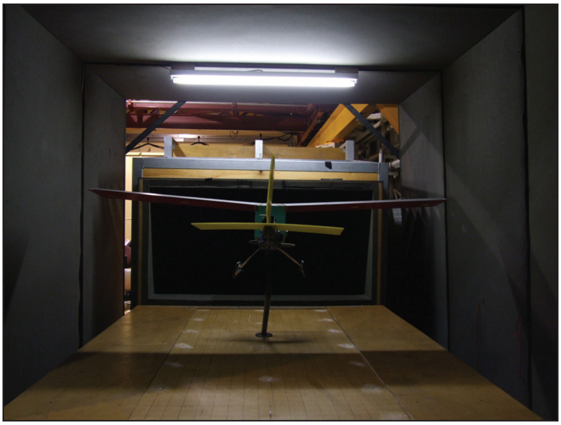
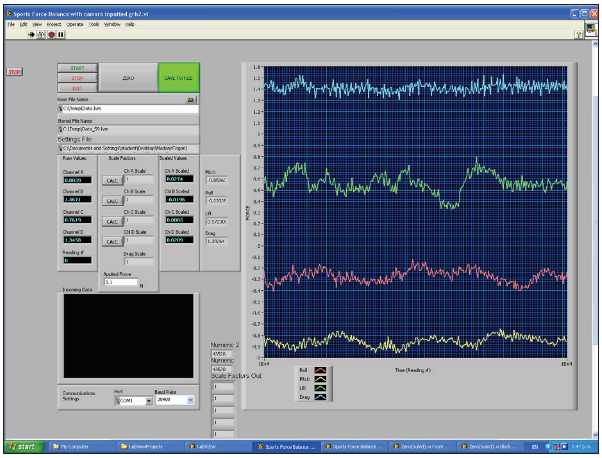


Fig. 14. Wind tunnel test setup

During Wind Tunnel testing, four load cell force readings were transmitted to LabVIEW (www.ni.com/labview) via a serial connection. LabVIEW was used to convert the raw force measurements into useful parameters – drag force, lift force, pitching moment and rolling moment. In order to be able to scale this data the software had to be calibrated. The software was reset at the start of each run. After each UAV test, known forces and moments were applied to the load cells which allowed the software to apply a scaling factor to the results. A sample readout of the LabView interface is shown in Fig. 15.



Base File Name: C:\Temp\Data.lvm

Stored File Name: C:\Temp\Data_59.lvm

Settings File: C:\Documents and Settings\student\Desktop\MadandTegan\

Raw Values	Scale Factors	Scaled Values	
Channel A 0.8039	Ch A Scale 1	Ch A Scaled 0.0274	Pitch -0.85060
Channel B 1.3671	Ch B Scale 1	Ch B Scaled -0.0196	Roll -0.23328
Channel C 0.7619	Ch C Scale 1	Ch C Scaled 0.0085	Lift 0.572300
Channel D 1.3458	Ch D Scale 1	Ch D Scaled 0.0209	Drag 1.39264
Reading # 0	Drag Scale 1		
	Applied Force 0.1 N		

Incoming Data

Communications Settings — Port: COM1 — Baud Rate: 38400

Numeric 2: 43520

Numeric: 43520

Scale Factors Out: 1, 1, 1, 1, 1

Fig. 15. LabVIEW wind tunnel output

For calibration an 18N (2 kg) weight was used. It was applied on the middle of the force balance to calibrate the lift force, pulled over the front using a pulley system to calibrate drag force, applied to the side a known distance to calibrate rolling moment (M = Fd) and applied to the front a known distance from the sting base to calibrate pitching moment. With two points (the zero value and the set value) LabVIEW can interpret the load cell data and hence calibrate itself since the relationship between force plate forces/moments and load cell transmission data is linear. Once the calibration was completed, the wind tunnel was started and the airflow was increased progressively from 0 kph to the maximum of 72 kph. The four load-cell series was plotted against time and the results exported to a spreadsheet using a LabVIEW application. The run was completed and the wind tunnel turned off once stable results were observed at maximum airflow speed. The conditions for the experiment are shown in Table 2.

Wind speed	72 ± 2kph	20 ± 0.5 ms-1
Pressure	985 ± 10 mbar	0.98 ± 0.01 atm
Temperature	16 ± 2 ^{o}C	289 ± K

Table 2. Experimental conditions

Runs were repeated for pitching and yawing angles -30 to 30 degrees inclusive with five degrees increments. In reality only pitching angles -10 to 20 degrees need to be considered because a plane will naturally stall outside this range (but the testing was done to show values outside this range nonetheless). Smaller increments of two degrees would have been optimal but because of mounting difficulties and error in the angle setup this was not easily achievable. The error in the angle setup was due to the UAV being mounted at a position away from its centre of gravity, at the back of fuselage. The UAV typically sagged forward and this could not be avoided because of mounting limitations (the mount could not be viced or clamped any more firmly without causing damage to the UAV) and flexing of the UAV airframe. As a result, during runs the UAV had some vibration which caused oscillations in the LabVIEW output. Therefore average values were used as opposed to maximum values.

5.4 Wind Tunnel and Software Coefficients Comparison

The LabVIEW experimental output was post-processed using Microsoft Excel and Matlab. The Lift Force, Drag Force, Pitching Moment and Rolling Moment were recorded for each trial. With images taken before each test the setup angle (pitching/yawing) could be analysed and the frontal area subjected to the airflow could be calculated using a pixel counting technique. With all of this data experimental coefficients were produced according to the following formulae:

$$C_x = \frac{F_x}{\overline{q}A} \tag{16}$$

$$C_x = \frac{M_x}{\overline{q}A} \tag{17}$$

$$\overline{q} = \frac{pV^2}{2} \tag{18}$$

where C = Aerodynamic Coefficient, F = Force (e.g. Lift force for coefficient of Lift), M = Moment (e.g. Pitching moment for Pitching coefficient), $\overline{q}$ = Dynamic pressure, A = Frontal area, p = Air density, V = Air velocity. (Since a 70% scaled model K70 was used, it is divided by the scale factor 0.7)

These results show that the existing software values are similar to what was determined using software with the exception of air drag. The reason the experimental drag coefficient is much greater than its software equivalents for the whole angle range is due to the software packages geometry limitations. Fig. 16 compares the coefficient profiles produced by the software packages and the wind tunnel results, and Fig. 17 shows how the coefficients change with yawing angle. It can be clearly seen that the experimental values display much more drag. For typical model aircraft the Datcom and Tornado drag coefficient estimates may be reasonably accurate, but for the K100, with its untypical blunt shape and large frontal area, the drag is obviously going to be much higher.

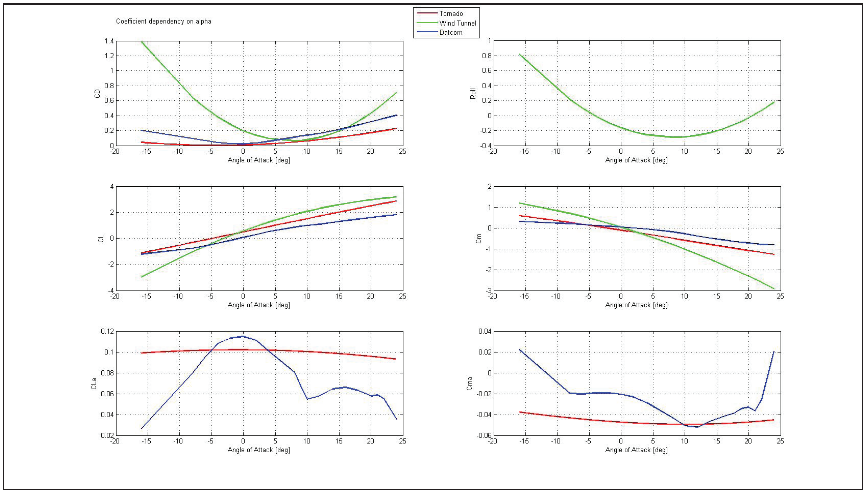


Fig. 16. Comparison of computed and experimental coefficients

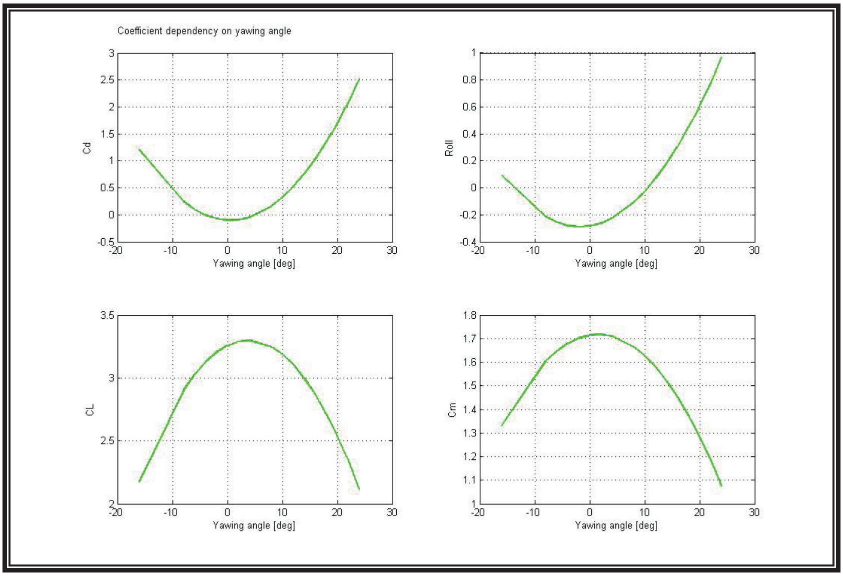


Fig. 17. Experimental coefficients for yawing angle changes

By substituting the software coefficients with the newly found experimental values, the wind speed velocity error observed in a flight simulation using the FDM is reduced and the flight dynamics model flight path improves. This verifies that the flight dynamics model can simulate a flight path similar to an actual flight given the control inputs are reasonably accurate. If the coefficients could be determined even more accurately the model may improve further.

6. Experimental Validation and Discussion

6.1 Flight Test Data

After careful preparation and organization, the flight tests using the model aeroplane K100 UAV were conducted. The flight test required the following field apparatus:

- K100 UAV – Perform flight tests, and record data.
- GPS Ground Station – Provide a stationary GPS reference.
- Camcorder – Record video of flights.
- Laptop – Perform data analysis in field.

Before a flight could be performed the GPS Ground station and Camcorder were set up. To ensure the data would be representative of a wide range of plane behaviour, it was necessary to gather data for all basic plane manoeuvres such as taxiing, take off, in flight movement and landing. Each flight lasted no less than five minutes. The data was downloaded to a laptop for analysis in between each flight to determine if it was useable.

The K100 was flown a number of times. The flight tests resulted in a 5 minute flight with good flight data for all of the parameters that required measurement. A considerable amount of post processing of data was undertaken for the flight test. All of the flight data that was logged in SD memory cards was needed to be interpreted in some way before any useful conclusions could be made. The attitude and position data were processed by combining information collected from the GPS base station and the onboard GPS and AHRS (www.xbow.com). The static position data provided by the GPS base station determined the errors of GPS signal. More precise navigation information can be obtained by subtracting the known errors.

Three ADC values were collected by the wind speed sensor in real time: differential pressure on angle of attack and sideslip, and the stagnation pressure in the central port. Even at a low sampling rate of 10Hz, the data captured by the wind speed sensor appeared to be quite noisy. By applying interpolation on the 2D lookup tables that were obtained in wind tunnel calibration, angle of attack α, angle of sideslip β, and body velocity along the x-axis u were derived. The other two body velocities v and w were easily derived using simple trigonometric relationships:

$$v = u \cdot tan(\beta) \tag{19}$$

$$w = u \cdot tan(a) \tag{20}$$

Since the flight model only accepts inputs for control surfaces in terms of deflection angles, conversions from servo pulse timing to the corresponding control surface deflection angles had to be made. Likewise, the thrust produced by the engine had to be correlated to the throttle servo pulses. The relationship between pulse servo signal pulse widths versus deflection angles and thrust were measured on the K100. The measurements are shown in Table 3 to Table 6. Interpolation was used to convert servo pulse widths into the mechanical inputs of the plane.

Servo Pulses (ms)	0.9	1.255	1.426	1.73	2.1
angle (deg)	16	8	0	-0.6	-1.3

Table 3: Elevator

Servo Pulses (ms)	0.9	1.125	1.25	1.45	1.565	1.60	1.78	1.9	2.1
Angle (deg)	18	14.4	10.8	6	0	-6	-10.8	14.4	-18

Table 4: Ailerons

Servo Pulses (ms)	0.8	0.97	1.33	1.544	1.72	1.91	2.1
Angle (deg)	12	10	5	0	-5	-10	-12

Table 5: Rudder

Servo Pulses (ms)	0.8	0.95	1.13	1.30	1.5	1.7	1.99	2.1
Thrust (N)	36.3	35.3	32.37	31.39	24.52	19.62	11.77	9.81

Table 6: Thrust

Fig. 18. and Fig. 19. show the servo pulse width signals and the converted control surface deflection angles and thrust. The collected signals covered the whole flight testing from taking off and landing. These results showed consistency with the UAV behaviour observed by watching the video recorded during the flight test.

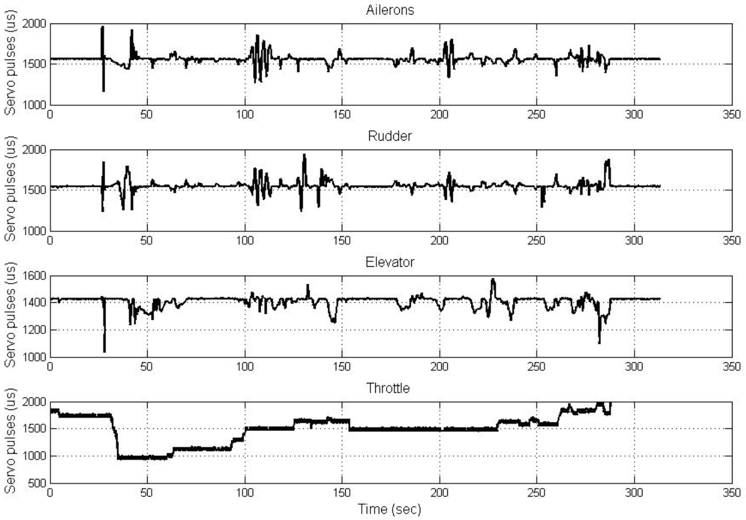


Fig. 18. Servo input signals

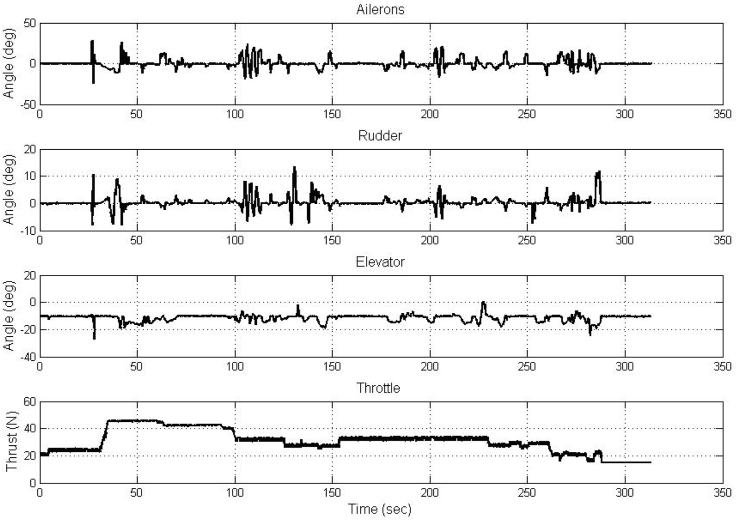


Fig. 19. Deflection angles and thrust

The vibrations from the two-stroke engine of the K100 caused problems for the wind speed sensor, which can be seen in Fig. 20. The vibrations of the sensor caused erroneous detection of angle changes, especially when the plane was stationary or moving slowly. This was caused by the probe tip movements resulting in a small pressure to be induced between opposite ports on the probe. Because of the slow speed, the longitudinal reading was low.

This condition is normally only present at very large angles. This theory was confirmed by the observation that only the AOA data was affected by this phenomenon. The sensor could only vibrate in the vertical direction because of how it was fixed on the plane wing. Note the UAV was stationary for the first 35 seconds of data recording. Vibration remained in 100 seconds after the throttle was turned down as can be seen in Fig. 20.

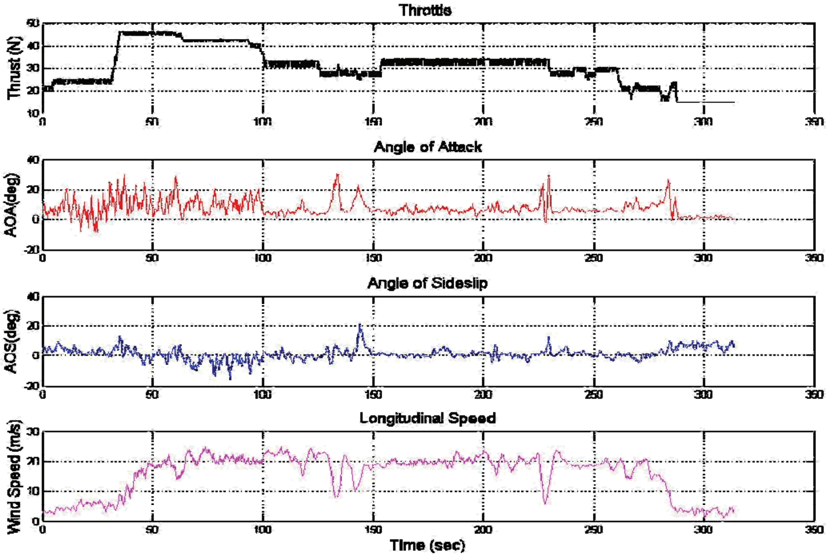


Fig. 20. Wind speed sensor data

6.2 Validation Results

Taking off and landing are generally much more difficult to model because of the more complicated environment. The added intricacies can be introduced by ground effect, low wind turbulence and the high non-linear flight response at low speed. As a result, only a section of the inputs were used to validate the flight dynamics model. Referring to Fig. 20, the chosen section for simulation was the period from 100 to 220 seconds. Fig. 21. shows the simulation results that used aerodynamic coefficients from Datcom exclusively. They were compared to the aeroplane responses that were measured by the onboard inertial reference system. The comparison was based on aeroplane attitude, altitude, flight path and body velocities. The roll angle, altitude change, and the vertical axis body velocity *w* generated by the FDM agree well with the actual flight response. In particular, the simulated roll angle gave the best match to the measured response. The simulated body velocity along x-axis *u* shows less resemblance to the experimental results. The much higher speed obtained from the model indicated that the drag coefficient used by the model is lower than the actual coefficient. The flight path is related to the integral of the velocities, so that both of flight path and body velocities exhibit a similar degree of inaccuracy.

The determined drag coefficient from the wind tunnel testing was used to replace the coefficient determined by Datcom. The simulation results given in Fig. 22. show a significant improvement for the body velocity parameters. The body velocity *u* shows that the flight model was not able to predict the velocity changes that occurred at around 30 and 40 second into the flight sample, but using the experimental coefficient for drag removed the offset error that can be seen in Fig. 22. In addition, the pitch angle was matched slightly better to the actual response. However, relatively large errors still exist on yaw angle and the flight path. The remaining experimental aerodynamic coefficients were not used for simulation because they produced an unstable flight response when used for the simulation.

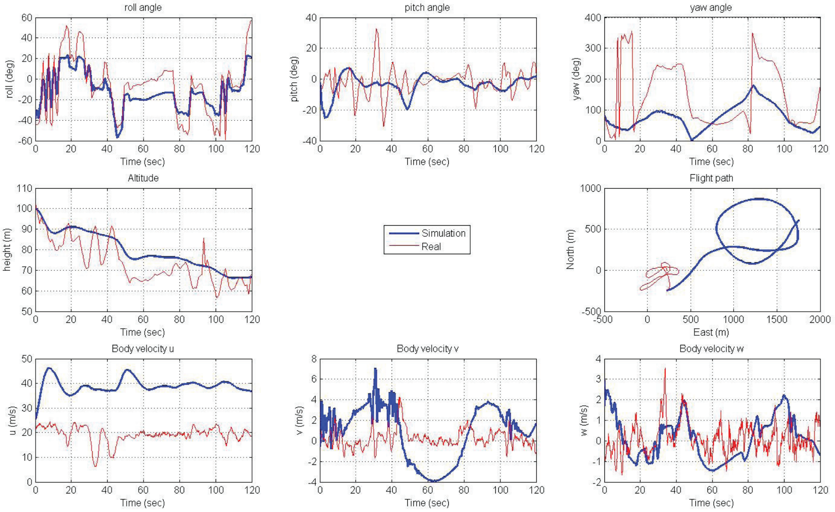


Fig. 21. Results based on coefficients generated by Datcom.

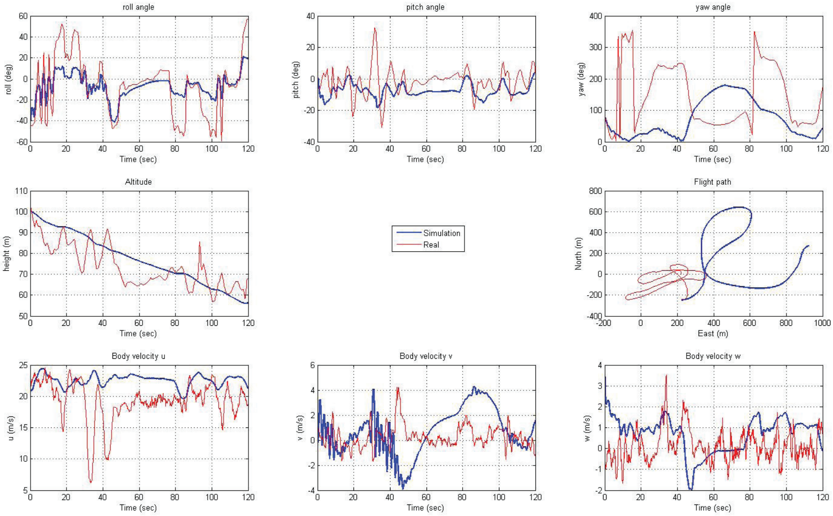


Fig. 22. Simulation results with wind tunnel determined drag coefficient

6.3 Model Errors and Improvements

The validation process has revealed the reliability of the simulation results given by the flight dynamics model with all the determined inputs. The modelling process is complex because of the variety of data that must be collected. There are many factors that can affect the simulated behaviour of the aircraft. Even though the simulation was able to predict the general trend of the aircraft motion, the large error found on the flight path and longitudinal velocity has limited its use for the application of dead-reckoning. The major sources of errors likely are as follows.

Control surfaces have a profound effect on the response of the aircraft. These effects are governed by the control surface aerodynamic coefficients, and they are normally non-linear and heavily dependent on the aircraft geometries. These coefficients are currently derived from Datcom, which determines their values from first principles. The nonlinearity of these coefficients implies that some errors must be involved from mathematical estimations.

Conversion between servo pulses and deflection angles were based on measurements taken when the aircraft was stationary. In flight, all control surfaces are subjected to high wind speed, which causes deflection and distortion. There is no easy way to measure the actual deflection angles. The angles were manually adjusted slightly to reduce this error.

Simulation errors were quantified by the measured aircraft responses. The measured responses involve uncertainties themselves due to noise, sensor limits and conversion inaccuracies. Quantifying the error in flight data instrumentation would allow an estimate of the effects of these errors on the simulation results.

Initial conditions affect the solution of a dynamic system. All initial conditions including linear and angular velocities, acceleration and position were measured by the onboard inertial navigation sensor system. This system has its own inaccuracies mostly caused by drift, which may have contributed to the flight model and experimental data discrepancies.

Wind condition inputs cause singularities when used in the current implementation of the flight model. The rapid change in wind data results in the model refusing to continue the calculation. This failure is caused by a combination of the limited quality of the wind speed data and a limitation of the model. A possible improvement on the flight model is to find out the cause and solution to this problem so that the model can include the measured wind vector in the calculation of the body velocities of the UAV. By doing so, the accuracy of the model would be improved significantly because ambient wind conditions can be taken into account. In addition, a higher sample rate in wind speed data collection would reduce the rapid rate of change in the data which causes the FDM to crash.

Effects of the control surfaces on the aircraft motion are significant. Determining the aircraft response with its control surfaces in a wind tunnel would greatly improve the simulation results.

In terms of the validation process, it was noted that a gas powered UAV can produce considerable vibrations, especially in the takeoff phase. These vibrations together with the turbulence behind the propeller caused significant noise to the wind speed sensor. This suggests a review of the mount position of the wind speed sensor and the selection of the UAV. A better position to place the wind speed sensor would be at the tip of the wing. With this position a counter weight has to be put on the other wing to cancel out the force and moment induced on the plane. An electric powered UAV would help to stop the wind speed sensor noise.

7. Conclusions

The objective of this project was to validate a flight dynamics model for the K100 UAV. This objective was achieved by determining the major aerodynamic coefficients of the K100 UAV and producing hardware for collecting flight data.

The wind tunnel testing was performed in a low speed wind tunnel using the K70 so accuracy was slightly compromised. However, the results were sufficient to show that for the unusually shaped K100 UAV, the aerodynamics coefficients determined by software packages (Datcom and Tornado) do not accurately represent the actual values. The experimental drag coefficients are higher than those predicted by the software model and this has a large affect on the accuracy of the flight dynamic model.

The sensor hardware developed during this project worked well during flight tests and allowed the collection of flight data which were used to assess the accuracy of the flight dynamics model. These sensors may be useful in other applications, such as aids for UAV navigation. The sources of instrumentation error were identified. The serious vibration generated by the K100 engines caused false AOA readings, particularly at low speeds. This could be overcome by improving the probe mounting location and method or using an electrically powered UAV.

The validation of the current software FDM has shown that it has two main limitations. It is unable to use some of the experimental aerodynamics coefficients because they produced unstable flight response. It was also unable to use the collected wind data because rapid changes caused the FDM to crash. Resolving these problems would improve the FDM, which otherwise represents the UAV flight reasonably well.

The aims of the model validation were met and a complete validation process for a flight dynamics model was presented. The current FDM has been assessed using this method and possible sources of inaccuracies identified. The presented validation process based on in-flight test and onboard instrumentation makes a significant step towards completing an accurate flight simulation system for auto-pilot development and design verification of UAVs.

8. Acknowledgement

The authors would like to thank Graeme Harris for his help with the wind tunnel testing, and Barry Lennox for his help with the flight test preparation and piloting. We would also like to thank the Geospatial Research Centre (NZ) Limited for their support throughout the project.

9. References

Barr, J. (2006). "FlightGear takes off", website: www.flightgear.org/.

Berndt, J. S. (2004). "JSBsim: An Open Source Flight Dynamics Model in C++." *AIAA Modeling and Simulation Technologies Conference and Exhibit AIAA* 2004-4923(AIAA).

Blaine K, B.-R. (1996). *Plane Geometry - Aircraft Geometry Measurement and Design Programs*.

Buschmann, M., Bange, J., Vörsmann, P. (2004). "MMAV - A Miniature Unmanned Aerial Vehicle (MINI-UAV) for Meteorological Purposes." *16th Symposium on Boundary Layers and Turbulence, American Meteorological Society*: Paper 6.7, 7pp.

Chavez, F. R., Bernard, J., et al. (2001). "Advancing the State of the Art in Flight Simulation via the Use of Synthetic Environments." *Iowa Space Grant Consortium*.

Cook, M. V. (2007). Flight dynamics principles. Amsterdam.

Cooke, J. M., Zyda, M. J., Pratt, D.R., McGhee, R.B. (1994). "NPSNET: Fight Simulation Dynamic Modeling using Quaternions." *In Presence 1*, No. 4: 404 - 420.

Diebel, J. (2006). "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors." (*Stanford University, Palo Alto, CA*).

Galbraith, B. (2004). "Datcom Predicted Aerodynamic Model." Holy Cows, www.holycows.net.

Grasmeyer, J.M., and Keennon, M. T. (2001). "Development of the Black Widow Micro Air Vehicle." *39th AIAA Aerospace Sciences Meeting and Exhibit* .

Guglieri, G., Pralio, B., Quagliotti, F. (2006). "Flight Control System Design for a Micro Aerial Vehicle." *Aircraft Engineering and Aerospace Technology* 78(2): 87 - 97.

Jackowski, J., Boothe, K., Albertani, R., Lind, R., Lfju, P. (2004). "Modeling the Flight Dynamics of a Micro Air Vehicle." *European Micro Air Vehicle Conference*.

Jordan, T. L., J. V. Foster, et al. (2006). "AirSTAR: A UAV Platform for Flight Dynamics and Control System." *NASA Langley Research Center,* Report Number: AIAA Paper 2006-3307: 8.

Lyshevski, S. E. (1997). "State-Space Identification of Nonlinear Flight Dynamics." *Proceedings of the 1997 IEEE International Conference on Control Applications*.

Marco, A. D. (2006). "A 6DoF Simulation Laboratory at the University of Naples." *The quarterly newsletter for JSBSim, an open source flight dynamics model in* C++ Volume 3(Issue 1).

Porro, A. R. (2001). "Pressure Probe Designs for Dynamic Pressure Measurements in a Supersonic Flow Field." *Glenn Research Center, Cleveland, Ohio*.

Ou, Q., Chen, X. C., Park, D., Marburg, A., Pinchin, J. (2008). "Integrated Flight Dynamics Modelling for Unmanned Aerial Vehicles." *Proc of the Fourth IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA08)*, ISBN: 978-1-4244-2368-2, Beijing, China, October 12-15, pp. 570-575.

Rasmussen, S. J., and Chandler, P. R. (2002). "Unmanned aerial vehicles: MultiUAV: a multiple UAV simulation for investigation of cooperative control." *Proceedings of the 34th conference on Winter simulation: exploring new frontiers. San Diego, California, Winter Simulation Conference*, pp. 869-877.

Ye, Z., Bhattacharya, P., Mohamadia, H., Majlesein, H., Ye, Y. (2006). “Equational Dynamic Modeling and Adaptive Control of UAV.” *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, CA, USA, April 2006, pp. 339 – 343.

10

Optimal Real-Time Estimation Strategies for a Class of Cyber-Physical Systems Using Networked Mobile Sensors and Actuators

Christophe Tricaud & YangQuan Chen
Center for Self-Organizing and Intelligent Systems (CSOIS)
Utah State University, Logan, UT84322-4160, USA
Emails: ctricaud@gmail.com; yqchen@ieee.org

1. Introduction

The combination of physical systems and networks has brought to light a new generation of engineered systems: Cyber-Physical Systems (CPS) (CPS, 2008). CPS is defined in (Chen, 2008) in the following way: ``*Computational thinking and integration of computation around the physical dynamic systems form Cyber-Physical Systems (CPS) where sensing, decision, actuation, computation, networking and physical processes are mixed*''. CPS is foreseen to become a highly researched area in the years to come with its own conferences (NSF, 2006; WCPS, 2008) and journals, e.g. (Gill *et al*, 2008).

``*Applications of CPS arguably have the potential to dwarf the 20-th century IT revolution*'' (Lee, 2007). CPS applications can be found in medical devices and systems, patient monitoring devices, automotive and air traffic control, advanced automotive systems, process control, environmental monitoring, avionics, instrumentation, oil refineries, water usage control, cooperative robotics, manufacturing control, buildings, etc.

The first step when considering a CPS is to determine the dynamics of its ``physical'' part, i.e. the environment in which the sensors and actuators are going to operate. First by defining a matching mathematical model, and then by retrieving the values of the parameters of this model. In this paper, the parameter estimation process constitutes a CPS in itself as we are using a mobile actuator-sensor network for that purpose.

The ``modeling-analysis-design (MAD)'' process in dynamic systems control is fundamental in control engineering practice. In both physical and mathematical modelling, the parameter estimation is essential in successful control designs.

A precise parameter estimation depends not only on ``relevant'' measurements and observations, but also on ``rich'' excitation of the system. These are all known concepts in system identification for finite dimensional systems (Ljung, 2008).

In control engineering practice, it is very common to estimate the parameters of a system given a mathematical model. Using observations or measurements, one can parameterize the model using different techniques. Sometimes, when the system to be modelled is spatially and temporally dynamic (i.e. the states depend on both time and space), common lumped parameter input-output relationships cannot characterize the system dynamics and instead, we must use partial differential equations (PDEs) for modelling.

However, making observations or measurements of the states of a distributed parameter system is not an easy and straightforward task. One needs to consider the location of the sensors so that the gathered information best helps model parameter identification. This problem of sensor location is not new but most of the work achieved so far is limited to stationary sensors in the context of wireless sensor networks (Kubrusly & Malebranche, 1985 ; Uciński, 2005).

Mobile sensors that are now available both via mobile robots or unmanned air/surface/underwater vehicles, offer a much more interesting alternative to stationary sensors for distributed parameter systems characterization. Indeed, when looking for optimal location of a stationary sensor, one only seeks the optimal average over time and space. However, the optimal location to gather sensible data about a given distributed parameter system is not necessarily static but in most cases, dynamic. It is therefore logical to expect a better estimation of the PDE-based system if mobile sensors are used than only considering stationary sensors.

Areas of application of such mobile sensing techniques include air pollutants monitoring using cars equipped with sensors on the ground and aircrafts in the air. In addition, low cost platforms for mobile sensors with wireless communications are now available and becoming cheaper and cheaper. As said, a set of such autonomous vehicles equipped with sensors can potentially improve the efficiency of the measurements. As technology evolves, it is necessary and practically meaningful to consider using mobile sensors for optimal measurement of distributed parameter systems with an objective of unknown parameters estimation.

In this chapter, we consider this type of research problem first introduced in (Walter & Pronzato, 1997), where the optimal observations of a DPS based on diffusion equations were made by two-wheeled differentially driven mobile robots equipped with sensors.

In the field of mobile sensor trajectory planning in a distributed parameter system setting, few approaches have been developed. So far, the available solutions are not quite practically appealing. For example, Rafajówicz (Rafajówicz, 1986) investigated the problem using the determinant of the Fisher Information Matrix (FIM) associated with the parameters to be estimated. The determinant of the FIM is used as a metric evaluating the accuracy of the parameters estimation. However, the results are more of an optimal time-dependent measure than a trajectory. In (Uciński, 2000) and (Uciński, 2005), Uciński reformulated the problem of time-optimal path planning into a state-constrained optimal control one which allows the addition of different constraints on the dynamics of the moving sensor. In (Uciński & Chen, 2005), Uciński and Chen tried to properly formulate and solve the time-optimal problem for moving sensors observing the state of a DPS for optimal parameter estimation.

In (Uciński & Chen, 2006), the Turing's Measure of Conditioning is used to obtain optimal sensor trajectories. The problem is solved for heterogeneous sensors (i.e. with different measurement accuracies) in (Tricaud et al., 2008). Limited power resource is considered in (Patan et al., 2008). In (Song et al., 2005), realistic constraints to the dynamics of the mobile sensor are considered when a differential-drive mobile robot in the framework of the MAS-net (mobile actuator and sensor networks) Project (Chen et al., 2004).

The framework was further extended in (Tricaud & Chen, 2008a) where the problem of optimal actuation or excitation to increase the relevance of the observations and measurements of the states of a distributed parameter system was introduced. Using similar methodology, an optimal mobile actuation policy was found for a class of distributed parameter systems.

It is pinpointed in (Song et al., 2005) that one of the fundamental problems in DPS-parameter estimation using mobile sensors is that the optimal paths for the DPS-parameter estimation are conditional on the very parameters' values that yet have to be estimated and which are in fact unknown. Given parameters, how to optimally plan the motion trajectories of the mobile sensors has been known in the literature (Uciński, 2005 ; Patan, 2004), where the purpose of mobile sensing is to best estimate the parameters. Clearly, there is a "chick-and-egg" problem regarding the optimal mobile sensor motion planning and parameter estimation for distributed parameter systems.

Tricaud and Chen (Tricaud & Chen, 2008b), for the first time, solved this problem by proposing optimal interlaced mobile sensor motion planning and parameter estimation. The problem formulation is given in detail with a numerical solution for generating and refining the mobile sensor motion trajectories for parameter estimation of the distributed parameter system. The basic idea is to use the finite horizon control (HFC) type of scheme.

First, the optimal trajectories are computed in a finite time horizon based on the assumed initial parameter values. For the following time horizon, the parameters of the distributed parameter system are estimated using the measured data in the previous time horizon, and the optimal trajectories are updated accordingly based on these estimated parameters obtained. Simulation results are offered to illustrate the advantages of the proposed interlaced method over the non-interlaced techniques. We call the proposed interlaced scheme "on-line" or "real-time" which offers practical solutions to optimal measurement and estimation of a distributed parameter system when mobile sensors are used. It should be mentioned that this "on-line" problem has been recognized in the last chapter of (Patan, 2004) as an "extremely important" research effort.

In what follows, we first present the problem formulation for optimal sensor location for parameter estimation in distributed parameter systems as in (Uciński, 2005). In Section 3, we introduce our approach for solving optimal actuation problems. In Section 4, we describe the method used to reformulate the optimal location problems into optimal control ones. In Section 5, we describe our developed scheme solving the "chicken-and-egg" problem described earlier. Finally, in Section 6, we illustrate our methods by applying them to a distributed parameter system governed by a diffusive partial differential equation.

2. Optimal Measurement Problem

2.1 Problem Definition

Consider a distributed parameter system (DPS), a class of CPS, described by the partial differential equation:

$$\frac{\partial y}{\partial t} = \mathfrak{F}\left(\mathbf{x}, t, y, \boldsymbol{\theta}\right) \quad \text{in } \Omega \times T, \tag{1}$$

with initial and boundary conditions:

$$\mathfrak{B}\left(\mathbf{x}, t, y, \boldsymbol{\theta}\right) = 0 \quad \text{on } \Gamma \times T, \tag{2}$$

$$y = y_0 \quad \text{in } \Omega \times \{t = 0\}, \tag{3}$$

where $y(\mathbf{x},t)$ stands for the scalar state at a spatial point $\mathbf{x} \in \bar{\Omega} \subset \mathbb{R}^n$ and time instant $t \in \bar{T}$. $\Omega \subset \mathbb{R}^n$ is a bounded spatial domain with sufficiently smooth boundary Γ, and $T = \left(0, t_f\right]$ is a bounded time interval. $\mathfrak{F}$ is assumed to be a known well-posed, possibly nonlinear, differential operator which includes first- and second-order spatial derivatives and includes terms for forcing inputs. $\mathfrak{B}$ is an known operator acting on the boundary Γ and $y_0 = y_0(\mathbf{x})$ is a given function.

We assume that the state y depends on the parameter vector $\theta \in \mathbb{R}^m$ of unknown parameters to be determined from measurements made by N static or moving pointwise sensors over the observation horizon T. We call $\mathbf{x}_s^j : T \to \Omega_{ad}$ the position/trajectory of the *j*-th sensor, where $\Omega_{ad} \subset \Omega \cup \Gamma$ is a compact set representing the domain where measurements are possible. The observations from the *j*-th sensor are assumed to be of the form:

$$z^j(t) = y\left(\mathbf{x}_s^j(t), t\right) + \varepsilon\left(\mathbf{x}_s^j(t), t\right), \quad t \in T, \quad j = 1, \ldots, N, \tag{4}$$

where ε represents the measurement noise which is assumed to be white, zero-mean, Gaussian and spatial uncorrelated with the following statistics:

$$E\left\{\varepsilon\left(\mathbf{x}_s^j(t), t\right)\varepsilon\left(\mathbf{x}_s^i(t'), t'\right)\right\} = \sigma^2 \delta_{ji} \delta(t - t'), \tag{5}$$

where σ^2 stands for the standard deviation of the measurement noise, δ_{ji} and $\delta(.)$ are the Kronecker and Dirac delta functions, respectively.

With the above settings, similar to (Uciński, 2005), the optimal parameter estimation problem is formulated as follows: Given the model (1)–(3) and the measurements z^j from the sensors $\mathbf{x}_s^j$, $j = 1, \ldots, N$, determine an estimate $\hat{\boldsymbol{\theta}} \in \Theta_{ad}$ (Θ_{ad} being the set of admissible parameters) of the parameter vector which minimizes the generalized output least-squares fit-to-data functional (Banks & Kunisch, 1989 ; Omatu & Seinfeld, 1989) given by:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\vartheta \in \Theta_{ad}} \sum_{j=1}^{N} \int_T \left[z^j(t) - y\left(\mathbf{x}_s^j(t), t; \vartheta\right) \right]^2 dt \tag{6}$$

where y is the solution of (1)–(3) with $\boldsymbol{\theta}$ replaced by ϑ.

By observing (6), it is possible to foresee that the parameter estimate $\hat{\boldsymbol{\theta}}$ depends on the number of sensors N and the mobile sensor trajectories $\mathbf{x}_s^j$. This fact triggered the research on the topic and explains why the literature so far focused on optimizing both the number of sensors and their trajectories. The intent was to select these design variables so as to produce best estimates of the system parameters after performing the actual experiment.

In order to achieve optimal sensor location, some quality measure of sensor configurations based on the accuracy of the parameter estimates obtained from the observations is required. Such a measure is usually related to the concept of the Fisher Information Matrix (FIM) (Sun, 1994), which is frequently referred to in the theory of optimal experimental design for lumped parameter systems (Fedorov & Hackl, 1997). Its inverse constitutes an approximation of the covariance matrix for the estimate of $\boldsymbol{\theta}$.

Let us write:

$$\mathbf{s}_s(t) = \left(\mathbf{x}_s^1(t), \ldots, \mathbf{x}_s^n(t)\right), \quad \forall t \in T, \tag{7}$$

and let $n = \dim\left(\mathbf{s}_s(t)\right)$. Given the assumed statistics of the measurement noise, the FIM has the following representation (Quereshi et al., 1980):

$$\mathbf{M}(\mathbf{s}_s) = \sum_{j=1}^{N} \int_T \mathbf{g}\left(\mathbf{x}_s^j(t), t\right) \mathbf{g}^T\left(\mathbf{x}_s^j(t), t\right) dt, \tag{8}$$

where

$$\mathbf{g}(\mathbf{x}, t) = \nabla_{\vartheta} y(\mathbf{x}, t; \vartheta)\Big|_{\vartheta = \boldsymbol{\theta}^0} \tag{9}$$

denotes the vector of the so-called sensitivity coefficients, $\boldsymbol{\theta}^0$ being a prior estimate to the unknown parameter vector $\boldsymbol{\theta}$ (Uciński, 2000).

However, the FIM can hardly be used in an optimization as is. Therefore, it is necessary to maximize some scalar function Ψ of the information matrix to obtain the optimal experiment setup. The introduction of the scalar criterion allows us to pose the sensor location problem as an optimization problem. Several choices for such a function can be found in the literature (Atkinson & Donev, 1992 ; Fedorov & Hackl, 1997 ; Walter & Pronzato, 1997) and the most popular one is the D-optimality criterion defined:

$$\Psi(\mathbf{M}) = -\log\det(\mathbf{M}). \tag{10}$$

Its use yields the minimal volume of the uncertainty ellipsoid for the estimates of the parameters. In this chapter, only the D-optimality criterion is considered.

2.2 Sensor Model

We assume that the sensors are mounted on vehicles whose dynamics are described by the following equation:

$$\dot{\mathbf{s}}_s(t) = f\left(\mathbf{s}_s(t), \mathbf{u}_s(t)\right) \quad \text{a.e. on } T, \quad \mathbf{s}_s(0) = \mathbf{s}_{s0} \tag{11}$$

where the function $f : \mathbb{R}^n \times R^r \to \mathbb{R}^n$ is continuously differentiable; $\mathbf{s}_{s0} \in \mathbb{R}^n$ represents the initial position of the sensors, and $u_s : T \to \mathbb{R}^r$ is a measurable control function satisfying the following inequality:

$$\mathbf{u}_{sl} \le \mathbf{u}_s(t) \le \mathbf{u}_{su} \quad \text{a.e. on } T \tag{12}$$

for some constant vectors $\mathbf{u}_{sl}$ and $\mathbf{u}_{su}$. We assume that all the vehicles have to stay within an admissible region Ω_{ad} (a given compact set) where measurements are possible. Ω_{ad} can be conveniently defined:

$$\Omega_{ad} = \left\{\mathbf{x} \in \Omega : b_{si}(\mathbf{x}) = 0, i = 1, \ldots I\right\}, \tag{13}$$

where the b_{si} are known continuously differentiable functions. That is, the following constraints have to be satisfied:

$$h_{sij}\left(\mathbf{s}_s(t)\right) = b_{si}\left(\mathbf{x}_s^j(t)\right) \le 0, \quad \forall t \in T, \tag{14}$$

where $1 \le i \le I$ and $1 \le j \le N$.

For simpler notation, we reformulate the conditions described in (14) in the following way:

$$\gamma_{sl}\left(\mathbf{s}(t)\right) \le 0, \quad \forall t \in T, \tag{15}$$

where γ_{sl} , $l = 1,\ldots,\nu$ tally with (14), $\nu = I \times N$.

It is possible to consider additional constraints on the path of each vehicle such as specific dynamics, collision avoidance and any other constrains. For example, we can restrict the minimum distance between the vehicles. Such constraint can be achieved by forcing the following condition:

$$\beta_{sij}\left(\mathbf{s}_s(t)\right) = R^2 - \left\|\mathbf{x}_s^i(t) - \mathbf{x}_s^j(t)\right\|^2 \tag{16}$$

where $1 \le i < j \le N$ and R is the minimum distance ensuring that the measurements taken by the sensors can be considered as uncorrelated, or ensuring that the vehicles will not collide each other during the experiment.

2.3 Problem Formulation

The optimal measurement problem consists in obtaining the steering of each mobile sensor, by minimizing the design criterion $\Psi(.)$ function of the FIMs of the form (8), which depend on the very trajectories of the sensors. Constraints (12) on the maximum admissible steering and state constraints (15) have to be satisfied. Initial sensors location $\mathbf{s}_{s0}$ will also be taken into account as a variable to be optimized in addition to the control function $\mathbf{u}_s$. The given problem can be reformulated as the following optimization: Find the pair $\left(\mathbf{s}_{s0}, \mathbf{u}_s\right)$ which minimizes the criterion:

$$J\left(\mathbf{s}_{s0}, \mathbf{u}_s\right) = \Psi\left[\mathbf{M}\left(\mathbf{s}_s\right)\right], \tag{17}$$

for all pairs of possible measurement:

$$P = \left\{\left(\mathbf{s}_{s0}, \mathbf{u}_s\right) \middle| \mathbf{u}_s : T \to \mathbb{R}^r \text{ is measureable, } \mathbf{u}_{sl} \le \mathbf{u}_s(t) \le \mathbf{u}_{su} \text{ a.e. on } T, \mathbf{s}_{a0} \in \Omega_{ad}^M \right\} \tag{18}$$

subject to constraints (15).

A methodology to solve this problem will be given in Section 4.

3. Optimal Actuation Problem

Note that, besides the explicit design variables there exists an implicit one that is the forcing input in (1). Therefore, for given sensor trajectories, our interest in this chapter focuses on designing the optimal forcing input so as to get the most accurate parameter estimates.

The optimal actuation problem is very close to the optimal measurement problem in the sense that both use the sensitivity coefficients as a measure of the quality of the parameter estimation. However, both problems differ in the following ways:

- The optimal measurement problem assumes that the forcing input in (1) is known whereas the optimal actuation problem attempts to optimize trajectories of mobile actuators constituting part of the entirety of the forcing input.
- In the optimal actuation problem, the sensors positions/trajectories are known beforehand and are not optimized, although it could be done jointly which is left as our future research effort.

We believe that when both sensors and actuators are movable, our framework presented in this chapter can solve the “smart sniffing and spraying” problem as outlined in (Chen et al.,

2004) in a model-based style, that is, we now have a solid basis on addressing the hard research question on "where to sniffing and where to spraying".

3.1 Actuator Model

Let us introduce:

$$\mathbf{s}_a(t) = \left(\mathbf{x}_a^1(t), \mathbf{x}_a^2(t), \ldots, \mathbf{x}_a^M(t)\right), \tag{19}$$

where $\mathbf{x}_a^k : T \to \Omega_{ad}$ is the trajectory of the *k*-th actuator. We assume that the actuators are mounted on vehicles whose dynamics are described by the following equation

$$\dot{\mathbf{s}}_a(t) = f\left(\mathbf{s}_a(t), \mathbf{u}_a(t)\right) \quad \text{a.e. on } T, \quad \mathbf{s}_a(0) = \mathbf{s}_{a0}, \tag{20}$$

Where the function $f : \mathbb{R}^M \times \mathbb{R}^r \to \mathbb{R}^M$ is continuously differentiable, $\mathbf{s}_{a0} \in \mathbb{R}^M$ represents the initial position of the actuators, and $\mathbf{u}_a : T \to \mathbb{R}^r$ is the control function satisfying the following inequality

$$\mathbf{u}_{al} \le \mathbf{u}_a(t) \le \mathbf{u}_{au} \quad \text{a.e. on } T, \tag{21}$$

for some constant known vectors $\mathbf{u}_{al}$ and $\mathbf{u}_{au}$.

We assume that all the vehicles are confined within an admissible region Ω_{ad} (a given compact set) where the actuation is possible. Ω_{ad} can be conveniently defined:

$$\Omega_{ad} = \left\{x \in \Omega : b_{ai}(\mathbf{x}) = 0, i = 1, \ldots, I\right\}, \tag{22}$$

where the b_{ai} functions are known continuously differentiable functions. That is to say that the following constraints have to be satisfied:

$$h_{aik}\left(\mathbf{s}_a(t)\right) = b_{ai}\left(\mathbf{x}_a^k(t)\right) \le 0, \forall t \in T, \tag{23}$$

where $1 \le i \le I$ and $1 \le k \le M$. For simpler notation, we reformulate the conditions described in (23) in the following way:

$$\gamma_{al}\left(\mathbf{s}_a(t)\right) \le 0, \forall t \in T, \tag{24}$$

where γ_{al}, $l = 1, \ldots, \nu$ tally with (23), $\nu = I \times M$. It would be possible to consider additional constraints on the path of the vehicles such as specific dynamics, collision avoidance and any other constraints.

The actuation function for the *k*-th mobile actuator is assumed to have the following form:

$$\mathfrak{F}_k(\mathbf{x}, t) = \mathfrak{G}_k\left(\mathbf{x}, \mathbf{x}_a^k, t\right). \tag{25}$$

3.2 Problem Definition

To define the considered problem, we reformulate (1):

$$\frac{\partial y}{\partial t} = \mathfrak{F}(\mathbf{x}, t, y, \boldsymbol{\theta}) + \sum_{k=1}^{M} \mathfrak{F}_k(\mathbf{x}, t) \quad in\ \Omega \times T, \tag{26}$$

with initial and boundary conditions remain unchanged. $\mathfrak{F}$ may still include forcing input terms.

For the framework of optimal actuation, the FIM is given by the following new representation:

$$\mathbf{M}(\mathbf{s}) = \sum_{k=1}^{M} \int_T \mathbf{h}\left(\mathbf{x}_a^k(t), t\right) dt, \tag{27}$$

where for the *k*-th actuator:

$$\mathbf{h}\left(\mathbf{x}_a^k(t), t\right) = \sum_{j=1}^{N} \mathbf{g}\left(\mathbf{x}_a^k(t), \mathbf{x}_s^j(t), t\right) \mathbf{g}^T\left(\mathbf{x}_a^k(t), \mathbf{x}_s^j(t), t\right), \tag{28}$$

and

$$\mathbf{g}\left(\mathbf{x}_a^k(t), \mathbf{x}(t), t\right) = \int_T \nabla_\vartheta \left(y\left(\mathbf{x}(\tau), \tau; \vartheta\right)\right)\Big|_{\vartheta=\boldsymbol{\theta}^0} d\tau. \tag{29}$$

In (29), y is the solution of (26) for $\mathfrak{F}_k(\mathbf{x}, \tau) = \mathfrak{G}_k\left(\mathbf{x}, \mathbf{x}_a^k, \tau\right)\delta(t-\tau)$ for all $k = 1, \ldots, M$.

The purpose of the optimal actuation problem is to determine the forces (controls) applied to each vehicle conveying an actuator, which minimize the design criterion $\Psi(.)$ defined on the FIMs of the form of (8), which are determined unequivocally by the corresponding trajectories, subject to constraints on the magnitude of the controls and induced state constraints. To increase the degree of optimality, our approach also considers $\mathbf{s}_{a0}$ as a control parameter vector to be optimized in addition to the control function $\mathbf{u}_a$.

Given the above formulation we can cast the optimal actuation policy problem as the following optimization problem: Find the pair $(\mathbf{s}_{a0}, \mathbf{u}_a)$ which minimizes

$$J(\mathbf{s}_{a0}, \mathbf{u}_a) = \Psi\left[\mathbf{M}(\mathbf{s}_a)\right] \tag{30}$$

over the set of feasible pairs

$$P = \left\{(\mathbf{s}_{a0}, \mathbf{u}_a) \middle| \mathbf{u}_a : T \to \mathbb{R}^r \text{ is measureable, } \mathbf{u}_{al} \le \mathbf{u}_a(t) \le \mathbf{u}_{au} \text{ a.e. on } T, \mathbf{s}_{a0} \in \Omega_{ad}^M\right\} \tag{31}$$

subject to the constraints .

The solution to this problem can hardly have an analytical solution. It is therefore necessary to rely on numerical techniques to solve the problem. A wide variety of techniques are available (Polak, 1997). However, the problem can be reformulated as a classical Mayer problem where the performance index is defined only via terminal values of state variables.

4. Optimal Control Problem Reformulation

In this section, both problems from Sections 2 and 3 are converted into canonical optimal control ones making possible the use of existing optimal control problems solvers.

To simplify our presentation, we define the function $\mathrm{svec}: \mathbb{S}^m \to \mathbb{R}^{m(m+1)/2}$, where $\mathbb{S}^m$ denotes the subspace of all symmetric matrices in $\mathbb{R}^{m\times m}$ that takes the lower triangular part (the elements only on the main diagonal and below) of a symmetric matrix $\mathbf{A}$ and stacks them into a vector $\mathbf{a}$:

$$\mathbf{a} = \mathrm{svec}(\mathbf{A}) = \mathrm{col}\left[\mathbf{A}_{11}, \mathbf{A}_{21}, \ldots, \mathbf{A}_{m1}, \mathbf{A}_{22}, \mathbf{A}_{32}, \ldots, \mathbf{A}_{m2}, \ldots, \mathbf{A}_{mm}\right]. \tag{32}$$

Reciprocally, let $\mathbf{A} = \mathrm{Smat}(\mathbf{s})$ be the symmetric matrix such that $\mathrm{svec}(\mathrm{Smat}(\mathbf{a})) = \mathbf{a}$ for any $\mathbf{a} \in \mathbb{R}^{m(m+1)/2}$.

Consider the matrix-valued function:

$$\Pi_s\left(s_s(t),t\right)=\sum_{j=1}^{N}\mathbf{g}\left(\mathbf{x}_s^j(t),t\right)\mathbf{g}^T\left(\mathbf{x}_s^j(t),t\right), \tag{33}$$

for the optimal sensor trajectory problem, and the function:

$$\Pi_a\left(\mathbf{s}_a(t),t\right)=\sum_{k=1}^{M}h\left(\mathbf{x}_a^k(t),t\right), \tag{34}$$

for the optimal actuator trajectory problem.

Setting $r_x : T \to \mathbb{R}^{m(m+1)/2}$, x being s or a, as the solution of the differential equations:

$$\dot{\mathbf{r}}_x(t)=\operatorname{svec}\left(\Pi_x\left(\mathbf{s}_x(t),t\right)\right),\quad \mathbf{r}_x(0)=0, \tag{35}$$

we can obtain:

$$\mathbf{M}(\mathbf{s}_x)=\operatorname{Smat}\left(\mathbf{r}_x\left(t_f\right)\right), \tag{36}$$

Minimization of $\Psi\left[\mathbf{M}(\mathbf{s}_s)\right]$ thus reduces to minimization of a function of the terminal value of the solution to (35). Introducing an augmented state vector:

$$\mathbf{q}_x(t)=\begin{bmatrix}\mathbf{s}_x(t)\\ \mathbf{r}_x(t)\end{bmatrix}, \tag{37}$$

we obtain:

$$\mathbf{q}_{x0}=\mathbf{q}_x(0)=\begin{bmatrix}\mathbf{s}_{x0}\\ \mathbf{0}\end{bmatrix}. \tag{38}$$

Then the equivalent canonical optimal control problem consists in finding a pair $(q_0,u)\in\bar{P}$ which minimizes the performance index:

$$\bar{J}\left(\mathbf{q}_{x0},\mathbf{u}_x\right)=\phi\left(\mathbf{q}_x\left(t_f\right)\right) \tag{39}$$

subject to:

$$\begin{cases}\dot{\mathbf{q}}_x(t)=\phi\left(\mathbf{q}_x(t),\mathbf{u}_x(t),t\right)\\ \mathbf{q}_x(0)=\mathbf{q}_{x0}\\ \bar{\gamma}_{xl}\left(\mathbf{q}_x(t)\right)\le 0\end{cases} \tag{40}$$

where:

$$\bar{P}=\left\{\left(\mathbf{q}_{x0},\mathbf{u}_x\right)\middle|\,\mathbf{u}_x:T\to\mathbb{R}^r \text{ is measurable, } \mathbf{u}_{xl}\le\mathbf{u}_x(t)\le\mathbf{u}_{xu} \text{ a.e. on } T,\ \mathbf{s}_{x0}\in\Omega_{ad}^M\right\} \tag{41}$$

and

$$\phi\left(\mathbf{q}_x,\mathbf{u}_x,t\right)=\begin{bmatrix}f\left(\mathbf{s}_x(t),\mathbf{u}_x(t)\right)\\ \operatorname{svec}\left(\Pi_x\left(\mathbf{s}_x(t),t\right)\right)\end{bmatrix}, \tag{42}$$

$$\bar{\gamma}_{xl}\left(\mathbf{q}_x(t)\right)=\gamma_{xl}\left(\mathbf{s}_x(t)\right). \tag{43}$$

The above problem in canonical form can be solved using one of the existing packages for numerically solving dynamic optimization problems, such as RIOTS_95 (Schwartz et al. 1997), DIRCOL (Stryk, 1999) or MISER (Jennings et al., 2002). We choose RIOTS_95, which is designed as a MATLAB toolbox written mostly in C and runs under Windows 98/2000/XP and Linux. The theory behind RIOTS_95 can be found in (Schwartz, 1996).

5. Real-Time Interlaced-Scheme

5.1 Measurements and Parameters Estimation

Once the optimal trajectories have been computed, the measurements are done as described in Section 1. However, the observations are completed until the end of the finite horizon for which the trajectory was computed. Instead, after a fraction of the horizon, the data gathered so far are used to refine the estimation of the parameters values.

In order to determine refined values of the parameters, we use the Matlab command “`lsqnonlin`", a routine for solving non-linear least squares problems and especially for our case, the least squares fitting problems. “`lsqnonlin`" allows the user to incorporate his/her own function to compute. In our problem, the input of the function is a set of parameters as well as the measurements and the output is the error between the measurement and the simulated value of the measurement for the set of parameters.

$$\min_{\boldsymbol{\theta}} = \frac{1}{2}\sum_{i=1}^{N} f_i(\boldsymbol{\theta})^2 \tag{44}$$

with

$$f_i(\boldsymbol{\theta}) = z^i(t_0,\ldots t_k) - \hat{y}\left(\mathbf{x}_s^i(t_0,\ldots,t_k),t_0,\ldots,t_k;\boldsymbol{\theta}\right). \tag{45}$$

Prior to the experiment, we determine the value of the state $\hat{y}(\mathbf{x},t;\boldsymbol{\theta})$ for a set of parameter value $\boldsymbol{\theta} \in \Omega_{ad}$ in an offline manner. We assume that the state variations between two values of a parameter are linear enough to allow interpolation. Using this database obtained “offline" allows faster computation of the function to be called by the optimization algorithm.

5.2 The Interlaced Scheme

Let us summarize the interlaced strategy step by step:

1. Given a set of parameters $\widehat{\boldsymbol{\theta}}$ for the DPS (its initial value being given prior to the first iteration), we design an optimal experiment, i.e., optimal trajectories for the mobile sensors to follow.
2. The sensors takes measurements along their individually assigned trajectories. Measurements are simulated taking the real value of the state along the trajectory and adding zero-mean white noise.
3. Measurement data are used to refine the estimate of the parameters using an optimization routine such as “`lsqnonlin`". The optimization routine computes the parameters such that the difference between the measurements and the simulated values of the state along the trajectory is minimized. Go back to Step-1.

The above algorithm is illustrated in Figure 1.

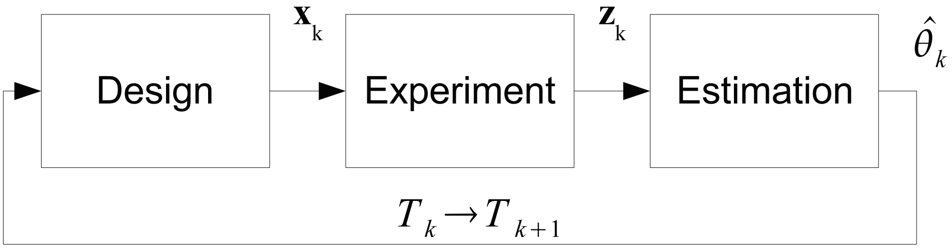


Fig. 1. The interlaced scheme illustrated

6. An Illustrative Example

6.1 Optimal Sensor Trajectories (Offline Results)

The model used for a specific diffusion process is the same as in (Uciński, 2005) except that the parameter values are different. The considered system is governed by the following diffusive partial differential equation:

$$\frac{\partial y}{\partial t} = \nabla \cdot (\kappa \nabla y) + \mathfrak{F}, \tag{46}$$

for $\mathbf{x} = [x_1, x_2]^T \in \Omega = (0,1)^2$ and $t \in [0,1]$, subject to homogeneous zero initial and Dirichlet boundary conditions. The spatial distribution of the diffusion coefficient is assumed to have the form:

$$\kappa(x_1, x_2) = \theta_1 + \theta_2 x_1 + \theta_3 x_2. \tag{47}$$

In our example, we select the initial estimates of the parameter values as $\theta_1^0 = 0.1$, $\theta_2^0 = 0.6$ and $\theta_3^0 = 0.8$, which are assumed to be nominal and known prior to the experiment. The forcing input $\mathfrak{F}$ is defined as:

$$\mathfrak{F}(\mathbf{x}, t) = 20 \exp\left(-50(x_1 - t)^2\right) \tag{48}$$

The sensitivity function is obtained "offline" or beforehand using the Matlab PDE Toolbox, prior to the function call of RIOTS by Matlab. The computation of the sensitivity function requires solutions of the followings equations:

$$\begin{cases} \frac{\partial y}{\partial t} = \nabla \cdot (\kappa \nabla y) + 20 \exp\left(-50(x_1 - t)^2\right) \\ \frac{\partial g_1}{\partial t} = \nabla \cdot \nabla y + \nabla \cdot (\kappa \nabla g_1) \\ \frac{\partial g_2}{\partial t} = \nabla \cdot (x_1 \nabla y) + \nabla \cdot (\kappa \nabla g_2) \\ \frac{\partial g_3}{\partial t} = \nabla \cdot (x_2 \nabla y) + \nabla \cdot (\kappa \nabla g_3) \end{cases} \tag{49}$$

where $\nabla = \partial/\partial x_1 + \partial/\partial x_2$. Note that there are three sensitivity equations since there are 3 parameters θ_1 , θ_2 , θ_3 .

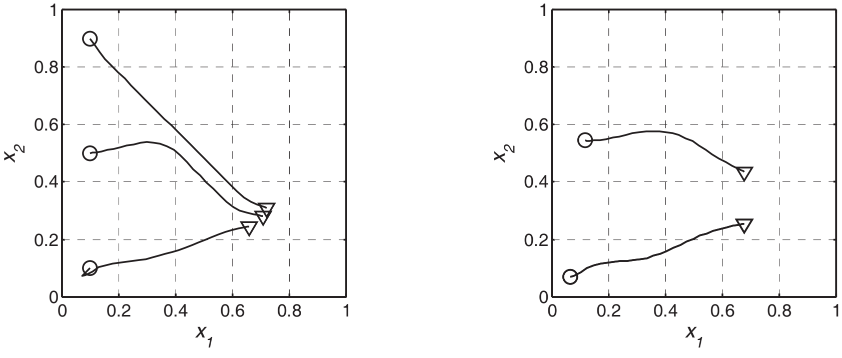
The dynamics of the mobile sensors follow the simple model:

$$\dot{\mathbf{x}}_s^j(t) = \mathbf{u}^j(t), \quad \mathbf{x}_s^j(0) = \mathbf{x}_{s0}^j \tag{50}$$

with additional constraints:

$$\left|\mathbf{u}(t)\right| \le 0.7, \quad \forall t \in T . \tag{51}$$

The optimal trajectories obtained are given in Figures 2 and 3. In Figure 2, all three sensors have fixed initial positions ($x_s^1(0) = (0.1,0.1)$, $x_s^2(0) = (0.1,0.5)$ and $x_s^3(0) = (0.1,0.9)$). In Figure 3, sensors initial positions are left to be optimized too.



(a) (b)

Fig. 2. (a): Optimal sensor trajectories with fixed initial positions, (b): Optimal sensor trajectories with optimal initial positions

It is important to mention that for Figure 2.(b), two sensors have the same trajectory. This result can be explained by the assumption that the noise is uncorrelated.

6.2 Optimal Actuator Trajectories

In this part, we use a similar example to illustrate our method. We consider the two-dimensional diffusion equation:

$$\frac{\partial y}{\partial t} = \nabla \cdot (\kappa \nabla y) + \sum_{i=1}^{M} \mathfrak{F}_i, \tag{52}$$

for $\mathbf{x} = [x_1, x_2]^T \in \Omega = (0,1)^2$ and $t \in [0,1]$, subject to homogeneous zero initial and Dirichlet boundary conditions. The spatial distribution of the diffusion coefficient is also assumed to have the form:

$$\kappa(x_1, x_2) = \theta_1 + \theta_2 x_1 + \theta_3 x_2. \tag{53}$$

In our example, we select the initial estimates of the parameter values as $\theta_1^0 = 0.1$, $\theta_2^0 = -0.05$ and $\theta_3^0 = 0.2$, which are assumed to be nominal and known prior to the experiment. The actuation function is defined:

$$\mathfrak{F}_i(\mathbf{x}, \mathbf{x}_a^i, t) = 1000 \exp\left(-50\left(\left(x_{a1}^i(t) - x_1\right)^2 + \left(x_{a2}^i(t) - x_2\right)^2\right)\right), \tag{54}$$

where $\mathbf{x}_a^i = \left[x_{a1}^i, x_{a2}^i\right]^T$. The dynamics of the mobile actuators follow the simple model:

$$\dot{\mathbf{x}}_a^j(t)=\mathbf{u}^j(t), \quad \mathbf{x}_a^j(0)=\mathbf{x}_{a0}^j \tag{55}$$

and additional constraints:

$$|\mathbf{u}(t)|\le 0.7, \quad \forall t\in T\ . \tag{56}$$

Our goal is to design their trajectories so as to obtain possibly the best estimates of θ_1, θ_2 and θ_3.

The determination of the FIM follows the same way used in Section 4.1.

Five different given sensor configurations or setups are considered, and for each setup optimal actuation trajectories of different number of actuators (1, 2 and 3) are compared:

1. One static sensor located in the centre of the domain (0.5, 0.5),
2. One static sensor located near one of the corners of the domain (0.2, 0.8),
3. Three static sensors located throughout the domain ((0.1, 0.7), (0.5, 0.2), (0.6, 0.4)),
4. One moving sensor with a linear motion (0.1, 0.2) → (0.6, 0.7),
5. Two moving sensors. One moving sensor with a linear motion (0.1, 0.2) → (0.6, 0.7) and the other one moving along an arc.

Results for the different cases are summarized in Table 1, and the resulting trajectories can be observed in Figures 4-8. In the figures, static sensors locations are represented by a red **x**, mobile sensors trajectories are in red and actuator trajectories are in blue (○ locates the starting point and ∇ the ending point).

	Case 1	Case 2	Case 3	Case 4	Case 5
1 actuator	15.991	18.051	10.904	14.465	12.547
2 actuators	12.582	14.273	7.36	11.095	7.4806
3 actuators	11.28	13.022	5.8136	9.8976	6.4512

Table 1. Values of the D-optimality criterion $\Psi(\mathbf{M})$ for the different test cases

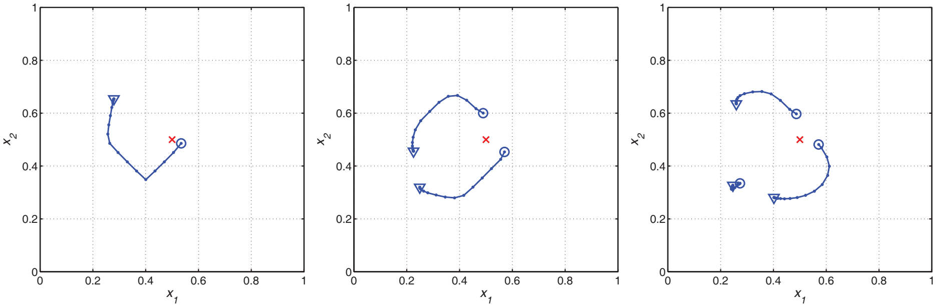


Fig. 4. D-Optimum trajectories of mobile actuators for one stationary sensor

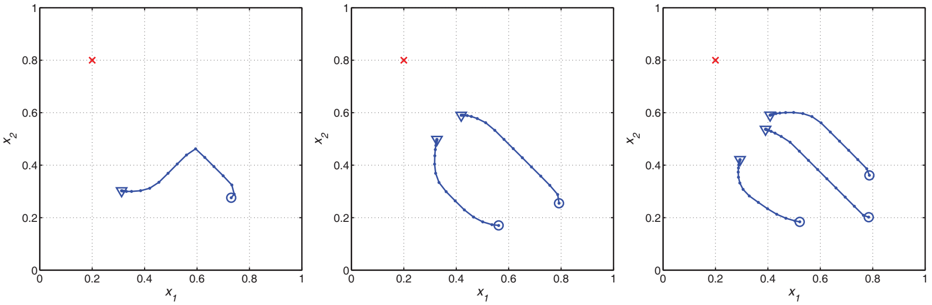


Fig. 5. D-Optimum trajectories of mobile actuators for one stationary sensor

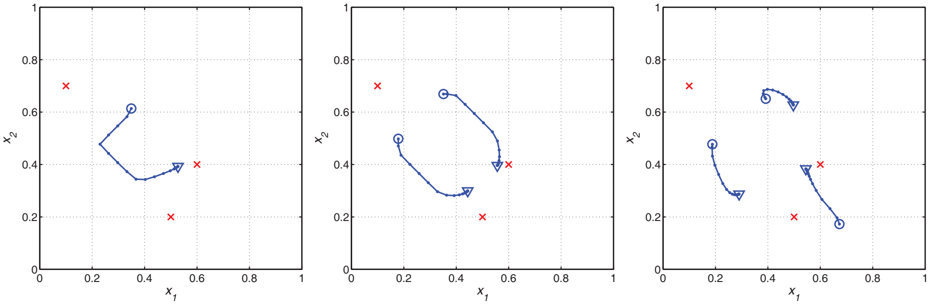


Fig. 6. D-Optimum trajectories of mobile actuators for three stationary sensors

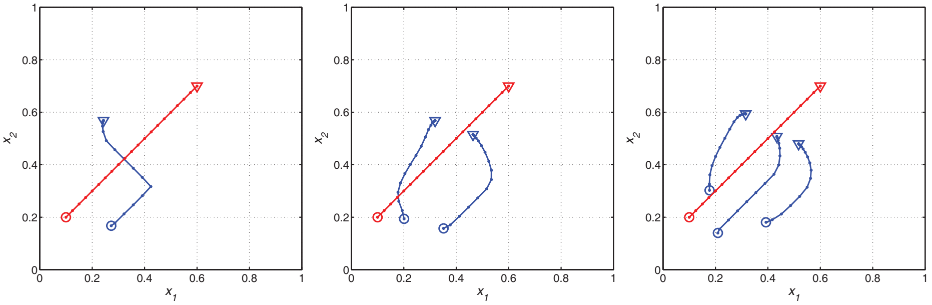


Fig. 7. D-Optimum trajectories of mobile actuators for one moving sensor

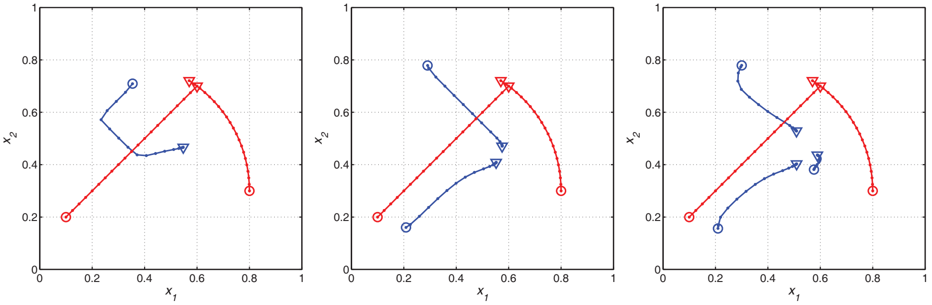
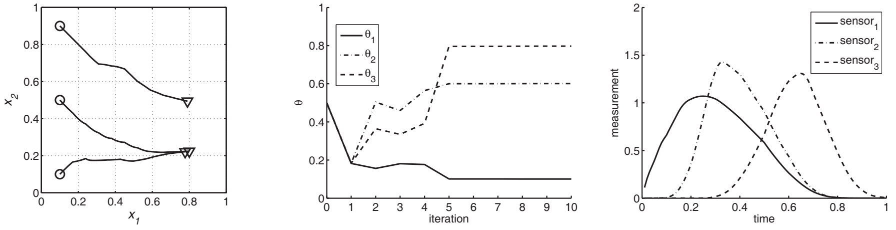


Fig. 8. D-Optimum trajectories of mobile actuators for two moving sensors

As expected, for all cases, the performance criterion value decreases as the number of actuator increases. We can also notice that both the mobility, population and location of the sensors have a direct impact on the performance of the strategy. Therefore, we can suppose the existence of an optimal combination of sensor and actuator trajectories.

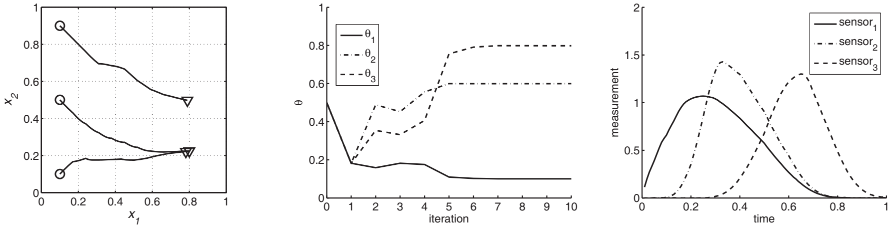
6.3 Optimal Sensor Trajectories (Online Results)

In this section, we focus our attention on the performance of the online methodology described in Section 5. The experiment is run for different noise statistics and for each case results are given in the form of sensor trajectories and parameter estimates. For case 1, $\sigma = 0.0001$, for case 2, $\sigma = 0.001$, and for case 3, $\sigma = 0.01$. In all cases, we consider 3 mobile sensors. The control of the mobile sensors u is limited between -0.7 and 0.7 . All three sensors have fixed initial positions ($x_s^1(0) = (0.1, 0.1)$, $x_s^2(0) = (0.1, 0.5)$ and $x_s^3(0) = (0.1, 0.9)$). The results for the previously defined case are respectively given in Figure 9. for Case 1, in Figure 10 for Case 2 and in Figure 11 for Case 3. For each figure, subfigure (a) gives the sensor trajectories, the evolution of the estimates is shown in (b) and the measurements are given in (c).



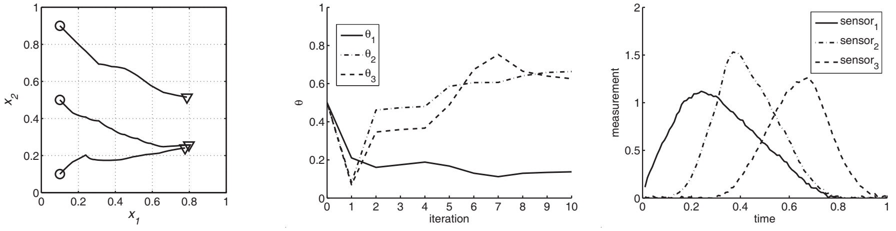
(a) (b) (c)

Fig. 9. Closed-loop D-Optimum experiment for $\sigma = 0.0001$. From left to right (sensor trajectories, parameter estimates and sensor measurements)



(a) (b) (c)

Fig. 10. Closed-loop D-Optimum experiment for $\sigma = 0.001$. From left to right (sensor trajectories, parameter estimates and sensor measurements)



(a) (b) (c)

Fig. 11. Closed-loop D-Optimum experiment for $\sigma = 0.01$. From left to right (sensor trajectories, parameter estimates and sensor measurements)

From these figures, we have the following observations:

- In all the cases, the sensors have similar trajectories as they try to follow the excitation wave along the x_1 axis $20\exp\left(-50\left(x_1 - t\right)^2\right)$.
- For low noise amplitude (cases 1 and 2), the experiment is long enough to obtain good estimates of the parameters. In case 3, the experiment is not long enough to obtain convergence.
- In all cases, we can clearly observe that the trajectories of the mobile sensors change as the estimated values of the parameters are getting closer to the real values.

7. Conclusions

In this chapter, we described a numerical procedure for optimal sensor-motion scheduling of diffusion systems for parameter estimation. The state of the art problem formulation was presented so as to understand our contribution to the field. The problem was formulated as an optimization problem using the concept of the Fisher information matrix.

We then introduced the optimal actuation framework for parameter identification in distributed parameter systems. The problem was reformulated into an optimal control one.

Later, using our developed “online” scheme, mobile sensors find an initial trajectory to follow and refine the trajectory as their measurements allow finding a better estimate of the system’s parameters. Using the Matlab PDE toolbox for the PDE system simulations, RIOTS_95 Matlab toolbox for solving the optimal path-planning problem and Matlab Optimization toolbox for the estimation of the system’s parameters, we were able to solve this parameter identification problem in an interlaced manner successfully.

With the help of the Matlab PDE toolbox for the system simulations and RIOTS_95 Matlab toolbox for solving the optimal control problem, we successfully obtained the optimal solutions of all the introduced methods for illustrative examples. We believe, this chapter has for the first time laid the rigorous foundation for real-time estimation for a class of cyber-physical systems (CPS).

8. Future Work

Our future efforts will go towards combining all of the techniques described here into a single framework. Obtaining optimal trajectories for both moving actuators and moving sensors is a challenging but very exciting research topic.

It should be emphasized that the "online" estimation methodology sets the basis for exciting future research. Indeed, we can now investigate problems related to communication between mobile nodes such as time-varying information sharing topology, communication range, information loss and other well-known problems in the scope of task-oriented mobile multi-agent systems.

9. Acknowledgements

This work is supported in part by the NSF International Research and Education in Engineering (IREE) grant #0540179. Christophe Tricaud was supported by Utah State University Presidential Fellowship (2006-2007).

The authors are grateful to Professors D. Uciński and M. Patan of the Institute of Control and Computation Engineering University of Zielona Góra for constructive research collaborations over the years. Christophe Tricaud would like to thank Professors D. Uciński and M. Patan for hosting his summer research visit in the Institute of Control and Computation Engineering University of Zielona Góra, in 2007 sponsored by an NSF IREE grant. YangQuan Chen appreciates the visit of Professor D. Uciński to CSOIS in 2006 and his invited lecture at the International Mini-Workshop on DDDAS (http://mechatronics.ece.usu.edu/mas-net/dddas/ Dynamic Data Driven Application Systems sponsored by an NSF DDDAS/SEP grant.

10. References

Atkinson, A. C. & Donev, A. N. (1992). *Optimum Experimental Designs*, Clarendon Press, Oxford

Banks, H. T. & Kunisch, K. (1989). *Estimation Techniques for Distributed Parameter Systems*, Systems & Control: Foundations & Applications, Boston: Birkhäuser

CPS (2008). C. S. Group, "Cyber-physical systems executive summary," in *Cyber-Physical Systems Summit*, 2008. [Online]. Available: http://varma.ece.cmu.edu/summit/CPS-Executive-Summary.pdf

Chen Y. Q., "Mobile actuator/sensor networks (MAS-net) for cyber-physical systems," USU ECE 6800 Graduate Colloquium, September 2008. [Online]. Available: http://www.neng.usu.edu/classes/ece/6800/

Chen, Y. Q. ; Moore, K. L. & Song, Z. (2004). Diffusion boundary determination and zone control via mobile actuator-sensor networks (MAS-net) – challenges and opportunities. *Proceedings of SPIE Conference on Intelligent Computing: Theory and Applications II, part of SPIE's Defense and Security*. SPIE, April 2004, Orlando, FL., USA

Fedorov, V. V. & Hackl, P. (1997). *Model-Oriented Design of Experiments*, Lecture Notes in Statistics, Springer-Verlag, New York

Gill H., Zhao W., and Znati T. (2008) "Call for papers for a special issue of on distributed cyber physical systems," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 19, no. 8, pp. 1150 – 1151, August 2008.

Jennings, L. S. ; Fisher, M. E. ; Teo, K. L. & Goh, C. J. (2002). *MISER 3: Optimal Control Software, Version 2.0. Theory and User Manual*, Department of Mathematics, University of Western Australia, Nedlands

Lee E. A. (2008) "Computing foundations and practice for cyber-physical systems: A preliminary report," University of California, Berkeley, Tech. Rep. UCB/EECS-2007-72, May 2007. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/306.html

Ljung L. (2008) *System Identification Toolbox TM7 User's guide*, The MathWorksTM, 2008. [Online]. Available: http://www.mathworks.com/products/sysid/

Kubrusly, C. S. & Malebranche, H. (1985). Sensors and controllers location in distributed systems — a survey. *Automatica*, 21, 2, (117–128)

Quereshi, Z. H. ; Ng, T. S. & Goodwin, G. C. (1980). Optimum experimental design for identification of distributed parameter systems. *International Journal of Control*, 31, 1 (21–29)

NSF (2006). "NSF workshop on cyber-physical systems," October 2006. [Online]. Available: http://warma.ece.cmu.edu/cps/

Omatu, S. & Seinfeld, J. H. (1989). *Distributed Parameter Systems: Theory and Applications*. Oxford Mathematical Monographs, New York: Oxford University Press

Patan, M. (2004). *Optimal Observation Strategies for Parameter Estimation of Ditributed Systems*. Ph.D. Dissertation, University of Zielona Góra Press

Patan, M. ; Tricaud C. & Chen Y. Q. (2008). Resource-constrained sensor routing for parameter estimation of distributed systems. *In Proceedings of the 17th IFAC World Congress*, July 2008, Seoul, Korea

Polak, E. (1997). *Optimization, Algorithms and Consistent Approximations.* Applied Mathematical Sciences, Springer-Verlag, New York

Rafajówicz, E. (1986). Optimum choice of moving sensor trajectories for distributed parameter system identification. *International Journal of Control*, 43, 5, (1441–1451)

Schwartz, A. L. (1996). *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*. PhD thesis, University of California, Berkeley

Schwartz, A. ; Polak, E. & Chen, Y. Q. (1997). *RIOTS - A Matlab Toolbox for Solving Optimal Control Problems*, May 1997. http://www.accesscom.com/~adam/RIOTS/

Song, Z. ; Chen, Y. Q. ; Liang, J. & Uciński, D. (2005). Optimal mobile sensor motion planning under nonholonomic constraints for parameter estimation of distributed parameter systems. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada

Stryk, O. von (1999). *User's Guide for DIRCOL, a Direct Collocation Method for the Numerical Solution of Optimal Control Problems*. Version 2.1. Fachgebiet Simulation und Systemoptimierung, Technische Universität Darmstadt, November 1999

Sun, N. Z. (1994). *Inverse Problems in Groundwater Modeling. Theory and Applications of Transport in Porous Media*. Dordrecht, The Netherlands: Kluwer Academic Publishers

Tricaud, C. ; Patan M. ; Uciński, D. & Chen Y. Q. (2008). D-optimal trajectory design of heterogeneous mobile sensors for parameter estimation of distributed systems. *Proceedings of 2008 American Control Conference*, June 2008, Seattle, Washington, USA

Tricaud, C. & Chen Y. Q. (2008a). Optimal Mobile Actuation Policy for Parameter Estimation of Distributed Parameter Systems, *Proceedings of Eighteenth International symposium on Mathematical Theory of Networks and Systems (MTNS2008),* July 2008, Virginia Tech, Blacksburg, Virginia, USA

Tricaud, C. & Chen Y. Q. (2008b). Optimal Mobile Sensing Policy for Parameter Estimation of Distributed Parameter Systems: Finite Horizon Closed-loop Solution, *Proceedings of Eighteenth International symposium on Mathematical Theory of Networks and Systems (MTNS2008),* July 2008, Virginia Tech, Blacksburg, Virginia, USA

Uciński, D. (2005). *Optimal Measurement Methods for Distributed-Parameter System Identification*. CRC Press, Boca Raton, FL,

Uciński, D. (2000). Optimal sensor location for parameter estimation of distributed processes. *International Journal of Control*, 73, 13, (1235–1248)

Uciński, D. & Chen Y. Q. (2005). Time-optimal path planning of moving sensors for parameter estimation of distributed systems. *Proceedings of 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, Seville, Spain

Uciński, D. & Chen, Y. Q. (2006). Sensor motion planning in distributed parameter systems using Turing's measure of conditioning. *Proceedings of 45th IEEE Conference on Decision and Control*, 2006, San Diego, CA

Walter, É. & Pronzato, L. (1997). *Identification of Parametric Models from Experimental Data. Communications and Control Engineering*. Springer-Verlag, Berlin

WCPS (2008). The first international workshop on cyber-physical systems," June 2008. [Online]. Available: http://www.qhdctc.com/wcps2008/

Effective Heuristics for Route Construction of Mobile Data Collectors

Samer Hanoun & Saeid Nahavandi
Centre for Intelligent Systems Research Deakin University
Australia

1. Introduction

Wireless sensor networks (WSN) are composed of large numbers of small sensing self-powered nodes which are densely deployed either inside the phenomenon or very close to it (Akyildiz et al., 2002; Culler et al., 2004). The capabilities of these inexpensive, low-power communication devices ensure serving in a wide range of application domains (Chong and Kumar, 2003; Haenggi, 2004). While their potential benefits are clear, a number of open problems must be solved in order for wireless sensor networks to become viable in practice. These problems include issues related to deployment, security, calibration, failure detection and power management.

Recently, significant advances have been accomplished in the field of mobile robotics (Engelberger, 1999), and robots have become increasingly more feasible in practical system design. Therefore, a number of problems with wireless sensor networks can be solved by including a mobile robot as an integral part of the system. Specifically, the robot can be used to deploy and calibrate sensors, detect and react to sensor failure, deliver power to sensors, and otherwise maintain the overall health of the wireless sensor network.

Energy consumption is one of the most important requirements for many WSN applications. Since energy resources are scarce and battery replacement is not an option for networks with thousands of physically embedded sensor nodes, energy optimization for individual nodes is required as well as the entire network. In static wireless sensor networks, it is observed that, as data traffic must be concentrated towards the sink (base station), the nodes around that sink have to forward data for other nodes whose number can be very large; this problem always exists, regardless of what energy conserving protocol is used for data transmission. As a result, those bottleneck nodes around the sink deplete their batteries much faster than other nodes and, therefore, their lifetime upper bounds the lifetime of the whole network.

Mobile collectors (mobile robots) are utilized to act as mechanical data carriers taking advantage of mobility capacity (Grossglauser and Tse, 2002) and physically approaching the sensors for collecting their data using single hop communication. This approach trades data delivery latency for the reduction of energy consumption of sensors; however, it shows remarkable enhancement in the network lifetime. Still, the data delivery latency depends mainly on the mobility regime applied by the collector.

In random mobility regimes (Burrell et al., 2004; Jain et al., 2006; Shah et al., 2003), random moving humans and animals act as "data mules", collect data opportunistically from sensor nodes when entering their communication ranges. A grid type topology is assumed for the network deployment with sensors located at the grid intersection points, where the mobile entities (data mules) can move in any of the four directions with equal probability (Shah et al., 2003) or move up and down the rows (Burrell et al., 2004). The random mobility regime shows improved data capacity (Grossglauser and Tse, 2002), however, in all cases, the worst-case latency of data delivery cannot be bounded. This unbounded latency may lead to excessive data caching at mobile entities, resulting in buffer overflows and data in transit may have to be dropped before being delivered to the destination, making it harder to provide transport layer reliability.

In predictable mobility regimes (Baruah et al., 2004; Chakrabarti et al., 2003), vehicles moving on a predesigned path collect sensors data when they move near them. The sensor nodes learn the times at which they have connectivity with the vehicle, and wake up accordingly to transfer their data. The trajectory of the mobile vehicle is known to the sensor nodes, which helps the sensors to save energy by sleeping until the predicted time of data transfer comes. A queuing model (Chakrabarti et al., 2003) is introduced to accurately model the data collection process. Using this queuing system model, the success rate of data collection and power consumption are analysed. Also, applying reinforcement learning to locate the vehicle efficiently at any point of time along its path is studied in (Baruah et al., 2004). The predictable mobility regime provides efficient solutions for saving the sensors energy consumption; however, it lacks flexibility and scalability as the sensors have to relearn the new data transfer times if the vehicle's path change and the need for redesigning the vehicle's path when transplanted to other networks.

In contrast, the controlled mobility regime adapts the motion strategy of the mobile collector according to the network runtime conditions to balance the sensors energy consumption and the data delivery latency. The message ferrying approach introduced in (Kansal et al., 2004; Zhao and Ammar, 2003; Zhao et al., 2004, 2005) controls the motion of a mobile relay to route messages between nodes in sparse networks. The idea is studied on networks with stationary sensor nodes (Kansal et al., 2004; Zhao and Ammar, 2003) and networks with mobile nodes (Zhao et al., 2004, 2005). The message ferry moves proactively to meet nodes wishing to send or receive packets and most communication involves short range radios for avoiding excessive energy consumption. The approach proves its strength in improving data delivery and energy efficiency; however, the collector acts only as a mobile relay between sparse sensor nodes.

Controlling the mobile collector motion for efficient data collection is presented in (Gu et al., 2005; Ngai et al., 2007; Somasundara et al., 2004; Tirta et al., 2006). The motion strategy of the mobile collector (element) is formulated as a scheduling problem based on knowing in advance the sensors sampling intervals and the rate by which the events in the environment occur. Offline solutions are provided in (Gu et al., 2005; Ngai et al., 2007; Tirta et al., 2006) based on having advance knowledge regarding the deployment locations of the sensors, their data generation rates, and buffer sizes. The solutions presented minimize the data latency by optimizing the collector inter-arrival times while using single hop communication for data transfer to minimize the sensor's energy consumption, remains that all operate offline and maladaptive to the network operational conditions. An online solution is presented in (Somasundara et al., 2004) that overcomes some of the former problems. A mobile data collector is scheduled in real time to visit sensors such that no sensor buffer overflow occurs. The algorithm considers buffer overflow deadlines as well as distances between nodes in determining the visiting schedule. The new deadline for the node's future visit is updated, once it is visited and depends mainly on knowing the node's buffer size and sensing rate to compute its next overflow deadline. The solution works online; still, there is a requirement on having full knowledge about the operational parameters of the sensor node.

This work is concerned with the controlled mobility regime. The former presents the solutions lying in this domain of research; however, more research remains and requires deep attention and addressing. The following are among these research points:

- Offline solutions (Gu et al., 2005; Ngai et al., 2007; Tirta et al., 2006) produce optimized results but depend mainly on having in advance full knowledge about the network operational parameters, which may not be always feasible. Solutions working online and in real-time are needed to accommodate changes in the network operation.
- The time required for the sensor's buffer to become full (overflow time) is assumed to be fixed and constant with time (Somasundara et al., 2004, 2007). Intelligent algorithms running locally on the sensor, performing local fusion and compression, impact the sensor's overflow time as this depends mainly on the data sensed and the quality threshold measures applied by the fusion and compression algorithms. Additionally, the overflow time can change with time according to the dynamics in the phenomenon which the sensors are sensing. Therefore, the change in the buffer overflow time with time has to be encountered in the solution.
- In scenarios where sensor nodes form clusters (Younis and Fahmy, 2004), the mobile data collector can visit the centroids of these clusters (cluster heads). An early arrival would force the mobile collector to wait until enough data is aggregated at the cluster head, and a late arrival may cause missing some of the aggregated data. Adaptive schedule considering the runtime conditions of the sensors is required to handle such cases.
- In event-driven sensor network applications, the dynamics in the phenomenon which the sensor nodes are sensing changes with time and the rate and times the events occur are not known ahead and even unpredictable. Moreover, in query-driven applications, the network operator may query the network to perform certain tasks, in irregular patterns. This requires that the motion strategy of the mobile collector to act based on these unpredictable situations.
- Deploying thousands of sensors manually and setting their locations is not a feasible process and in most situations the sensors are deployed randomly in the environment. Moreover, some sensors may be carried by low mobile platforms and their locations change from time to time. These factors impact the mobile collector schedule and should be adopted properly. The network performance is considered to be dependent only on the sensor nodes in the network. However, the embedded capabilities of the mobile collector (i.e speed) can be utilized to enhance the network performance. Also, using multiple mobiles with appropriate cooperation strategies can add more benefit to the data collection operation. Adaptive decentralized solutions are required to take advantage of these capabilities.

Accordingly, in this work a dynamic scheme for data collection in wireless sensor networks is presented. The scheme aims to consider most of the research points described above. First of all, the mobile element utilized for collecting the sensors' data buffers works online and constructs its collection route dynamically according to the collection requests received from the sensors requiring the collection service. Secondly, the mobile collector does not require any prior knowledge about the sensors deployment positions, their data generation rates and buffer size. This is achieved by making the sensor node disseminates its deployment position in the collection request sent to the mobile collector. Additionally, when the sensor's buffer becomes full and after disseminating the collection request, it sleeps and waits the arrival of the collector, which relieves the collector the requirement of knowing ahead the sensor's data generation rate and buffer size to schedule its collection before the buffer overflow time comes. Therefore, each sensor in the network can operate with different sensing frequencies and its buffer size may vary according to the type of application the network is serving (i.e. event-driven and query-driven sensor network applications). However, this achieves no dependency between the sensor network and the mobile collector; it requires optimizing the sensor sleeping time waiting the arrival of the collector to avoid missing changes in the observed phenomenon and to increase the network activity.

Experimental testing and simulations are performed to evaluate the presented heuristics with real world data parameters. Performance metrics are selected for comparing the proposed heuristics. The results show that the presented heuristics are capable of reducing the data collection time and ensure high network activity along its operational lifetime. Also, online and dynamic solutions can provide high flexibility and scalability and lose coupling between the sensor network and the mobile collectors can be achieved.

2. Related Work

This section presents some relevant literature in routing and scheduling theory.

2.1 Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is one of the classical challenging combinatorial optimization problems. The objective of the TSP is to minimize the total distance travelled by visiting all locations once and only once and then returning to the depot point. A common application of the TSP is the movement of people, equipment and vehicles around tours of duty to minimize the total travelling cost. For example, in a school bus routing problem, it is required to schedule the bus to pick up waiting students from the pre-specified locations. Post routing is another application of the TSP. The postman problem is modelled as traversing a given set of streets in a city, rather than visiting a set of specified locations.

Besides the above mentioned applications, some other seemingly unrelated problems are solved by formulating them as the TSP. The genome sequencing problem occurs in the field of bio-engineering. The aim of this problem is to find the genome sequence based on the markers that serve as landmarks for the genome maps. The drilling problem is another application of the TSP with the objective of minimizing the total travel time of the drill. In the electronic industry, the printed circuit board normally has a very large number of holes used for mounting components or integrated chips. These holes are typically drilled by automated drilling machines that move between specified locations to drill a hole one after another. Therefore, the locations in the drilling problem correspond to the cities in the TSP. The applications of the TSP are not limited to the examples described above. A detailed review of the applications of the TSP can be found in (Lawler et al., 1990).

It has been proved that TSP is NP-hard in (Garey and Johnson, 1979), which implies that a polynomial bounded exact algorithm for TSP is unlikely to exist. Several heuristic based tour construction algorithms were proposed for solving the TSP. Among these, the nearest neighbour procedure (Rosenkrantz et al., 1977), the Clarke and Wright savings' algorithm (Clarke and Wright, 1964), the partitioning approach (Karp, 1977) and the minimal spanning tree approach (Christofides, 1976). Comprehensive review of the techniques developed for the TSP can be found in (Bodin et al., 1983; Laporte, 1992).

It is important to clearly outline the differences between our problem and the conventional Travelling Salesman Problem. In TSP, the goal is to find a minimum cost tour that visits each node exactly once. However in our problem, a sensor node may need to be visited multiple times before all other nodes are visited depending on the rate of its data generation. In addition, the mobile collector downloads the data once it is in the communication range of the sensor node and need not to be exactly at the sensor location. Also the transfer of the data buffer can be done while the mobile collector is in motion.

2.2 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) (Toth and Vigo, 2001) can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points. There are nodes each has a service request in terms of demand for a certain quantity of goods, and vehicles available for servicing these requests, which are stationed at the depot. Each vehicle has a certain capacity in terms of quantity of goods it can carry. The goal is to find the number of vehicles and the sequence of nodes each vehicle has to visit such that sum of the distances travelled by each vehicle is minimum, subject to the following constraints:

- Each vehicle starts and ends at the depot.
- Each node is serviced exactly by one vehicle.
- Capacity constraints are met, i.e. sum of demands from the nodes on a vehicle's list does not exceed the vehicle's capacity.

The Traveling Salesman Problem (TSP) mentioned earlier is a special case of VRP, where there is a single vehicle that visits the nodes, and there are no capacity constraints.

There are many variants to the basic VRP mentioned above. The VRP with time Windows (VRPTW) (Solomon, 1987) has an added constraint in which there is a time window within which each node has to be visited. The VRPTW can be defined as follows. Let G = (V, E) be a connected digraph consisting of a set of $n + 1$ nodes, each of which can be serviced only within a specified time interval or time window, and a set E of arcs with non-negative weights d_{ij} and with associated travel times, t_{ij}. The travel time t_{ij} includes a service time at node i, and a vehicle is permitted to arrive before the opening of the time window, and wait at no cost until service becomes possible, but it is not permitted to arrive after the latest time window. Node 0 represents the depot. Each node i, apart from the depot, imposes a service requirement q_i that can be a delivery from, or a pickup for the depot. The main objective is to find the minimum number of tours, *K**, for a set of identical vehicles such that each node is reached within its time window and the accumulated service up to any node does not exceed a positive number *Q* (vehicle capacity). A secondary objective is often either to minimize the total distance travelled or the duration of the routes. All problem parameters, such as customer demands and time windows, are assumed to be known with certainty. Moreover, each customer must be served by exactly one vehicle, thus prohibiting split service and multiple visits. The tours correspond to feasible routes starting and ending at the depot.

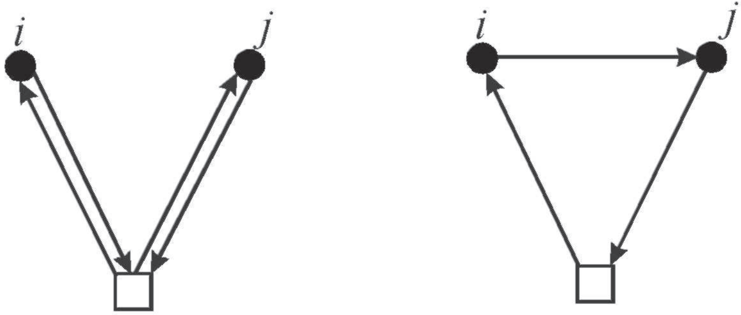


Fig. 1. The savings heuristic. In the left part, customers *i* and *j* are served by separate routes; in the right part, the routes are combined by inserting customer *j* after *i*

There is a dynamic version of VRP, known as Dynamic Vehicle Routing Problem (DVRP) (Ghiani et al., 2003). Here the information (input) is revealed to the decision maker online concurrently with the determination of routes. For instance, the VRP mentioned above is solved for the initial requests (nodes to be visited) and routes assigned to vehicles. Then, as the vehicles start their scheduled routes, new requests come, which need to be accommodated.

There are some differences between our problem and the VRPTW. First of all, some nodes may need to be visited more than once before visiting any other node once depending on its data generation rate. Secondly, the number of mobile collectors servicing the network is fixed, thus the goal is no longer to find the number of mobiles but to find the most feasible mobile collector for servicing the request. Also, the time window (i.e. the time for the data collection) is not constraint; however, the objective is to minimize the data collection time among all sensors and for all generated collection requests. Thus our problem reduces from the VRPTW to the problem of designing the route for the mobile collector to follow. This route must be optimized to provide the minimum data collection time among the sleeping sensors waiting for the arrival of the collector.

The savings method (Clarke and Wright, 1964), originally developed for the classical VRP, is probably the best-known route construction heuristic. It begins with a solution in which every customer is supplied individually by a separate route. Combining the two routes serving respectively customers *i* and *j* results in a cost savings of $S_{ij} = d_{i0} + d_{0j} - d_{ij}$. The arc *(i, j)* linking customers *i* and *j* with maximum S_{ij} is selected subject to the requirement that the combined route is feasible. With this convention, the route combination operation is applied iteratively. In combining routes, one can simultaneously form partial routes for all vehicles or sequentially add customers to a given route until the vehicle is fully loaded. To account for both the spatial and temporal closeness of customers, a limit to the waiting time of the route is set. The savings method is illustrated in Figure 1.

The second heuristic, a time oriented nearest-neighbour, starts every route by finding an unrouted customer closest to the depot. At every subsequent iteration, the heuristic searches for the customer closest to the last customer added into the route and adds it at the end of the route. A new route is started any time the search fails to find a feasible insertion place, unless there are no more unrouted customers left. The metric used to measure the closeness of any pair of customers attempts to account for both geographical and temporal closeness of customers.

The most successful methods among the sequential insertion heuristics is called I1 (Solomon, 1987). A route is first initialized with a "seed" customer and the remaining unrouted customers are added into this route until it is full with respect to the scheduling horizon and/or capacity constraint. If unrouted customers remain, the initializations and insertion procedures are then repeated until all customers are serviced. The seed customers are selected by finding either the geographically farthest unrouted customer in relation to the depot or the unrouted customer with the lowest allowed starting time for service. After initializing the current route with a seed customer, the method uses two subsequently defined criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ to select customer u for insertion between adjacent customers i and j in the current partial route.

Let $(i_0, i_1, i_2,..., i_m)$ be the current route with i_0 and i_m representing the depot. For each unrouted customer u, compute first its best feasible insertion cost on the route as:

$$c_1(i(u),u,j(u)) = \min_{p=1,\ldots,m} c_1(i_{p-1},u,i_p) \tag{1}$$

Next, the best customer u* to be inserted in the route is the one for which:

$$c_2(i(u^*),u^*,j(u^*)) = \max_u \{\mathrm{c}_2(i(u),u,j(u)), u \text{ is unrouted and route is feasible}\} \tag{2}$$

Client u^* is then inserted into the route between $i(u^*)$ and $j(u^*)$. When no more customers with feasible insertions can be found, the method starts a new route, unless it has already routed all customers. More precisely $c_1(i,u,j)$ is calculated as:

$$c_1(i,u,j) = \alpha_1 c_{11}(i,u,j) + \alpha_2 c_{12}(i,u,j), \text{ where } \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, \tag{3}$$

$$c_{11}(i,u,j) = d_{iu} + d_{uj} - \mu d_{ij}, \quad \mu \geq 0, \tag{4}$$

$$c_{12}(i,u,j) = b_{ju} - b_j, \tag{5}$$

and d_{iu}, d_{uj} and d_{ij} are distances between customers i and u, u and j and i and j respectively. Parameter μ controls the savings in distance and b_{ju} denotes the new time for service to begin at customer *j*, given that *u* is inserted on the route and b_j is the beginning of service before insertion. The criterion $c_2(i, u, j)$ is calculated as follows:

$$c_2(i,u,j) = \lambda d_{0u} - c_1(i,u,j), \quad \lambda \geq 0. \tag{6}$$

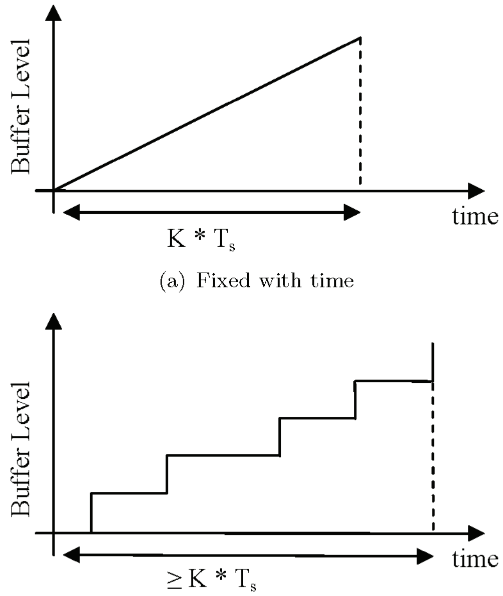
Parameter λ is used to define how much the best insertion place for an unrouted customer depends on its distance from the depot and on the other hand how much the best place depends on the extra distance and extra time required to visit the customer by the current vehicle. Other types of the insertion heuristics such as Type (I2) aims to select customers whose insertion costs minimize a measure of total route distance and time and Type (I3) accounts for the urgency of servicing a customer.

3. Problem Formulation

This section clearly models our problem and outlines all assumptions. A formal definition of the problem is presented to give an insight into the problem objectives. The problem is modeled as follows:

- The sensor network is modeled as a fully connected graph *G(V, E)* of *n* nodes, where every edge (s_i, s_j) has an associated cost *c(i,j)*, and all the costs form a matrix $C = \left[c(i,j)\right]_{i,j=1}^{n}$.
- Sensor nodes remain stationary and are deployed uniformly at random in the sensing field where the phenomenon of interest is to be monitored. The sensing field dimension is *A* = *L* * *L* (*m2*).
- Each sensor node has a deployment location (*X, Y*), buffer size of K bytes, radio communication radius *R* in meters and generates a sample every T_s seconds.
- The time required for the buffer to be full is ≥ $K * T_s$, where the next time the buffer becomes full is unknown and differs from time to time. This is modeled by generating a random number between 0.0 and 1.0 to specify whether the current sample is qualified to be added to the buffer or not. If the random number is greater than a pre-specified threshold φ then the sample is added to the buffer, otherwise, it is discarded. This ensures that the sensor's buffer full time is variable which reflects different network operational modes (i.e. periodic sensing, event driven, query based). This also simulates the operation of data aggregation algorithms operating locally on the sensor for reducing the redundancy in the sensor measurement which alters the buffer overflow time. Figure 2 shows the buffer level accumulation with time for different models.
- Once the sensor's buffer is full, the sensor disseminates a collection request *Q* with fields {*ID, X, Y*} and goes to an idle state and sleeps waiting for the collector arrival. While idle the sensor does not sense any new samples but it relays collection requests sent by others.
- The mobile collector starts at a central position in the sensing field, has a reasonably high amount of energy that can last beyond the network lifetime, and its memory size can accommodate the data generated by the sensors during the network operational

time. The mobile collector moves with a speed $v \leq S_{\max}$ and it can change its speed during the data collection operation.



(a) Fixed with time

(b) Variable with time

Fig. 2. The sensor buffer level accumulation rate

The following assumptions are made:

- At time t = 0 all the buffers of the sensor nodes start filling up.
- The actual data transfer time from the sensor node to the mobile element is negligible.

The **Mobile Collector Route Design (MCRD)** problem is the problem of finding a sequence of visits to nodes requesting the collection of their data buffers such that to minimize the collection time required. Once a node is visited, its buffer is transferred to the mobile collector internal memory and its normal sensing operation is resumed. This is mathematically formulated as follows:
N: is the set of sensor nodes in the network, where $N = \{s_1, s_2,..., s_n\}$,
S: is the set of sensor sites, i.e. deployment locations of the sensor nodes, where
$S = \{(x_1,y_1),...(x_n,y_n)\}$,
d_{ij}: Euclidean distance (meters) between sensor sites, where $i,j \in S$,
R: is the set of requests to be serviced, where $R = \{r_1,r_2,...,r_m\}$,
A_k: is the time taken for servicing request k, where $k \in R$, i.e. time taken for collecting request k generated by sensor i, where $k \in R$, $i \in N$,
v: is the moving speed of the mobile collector in m/sec,

Mobile Collector Route Design (MCRD):

$$\min T = \sum_{\forall k} A_k \tag{7}$$

where T is the overall route period, and each A_k is computed based on the distance cost between requests k and $(k+1)$ and the speed selected by the mobile collector for servicing this request.

The problem of finding the optimum order of arranging the requests to form the least cost route is *NP*-complete. The way is to simply enumerate each possible route, and pick the one with minimum total cost. Having m requests to service, gives $(m - 1)!$ possible route. This leads to an $O(m!)$ algorithm to produce the optimum route, which is not efficient to use.

4. Heuristic Based Algorithms

The problem of designing the mobile collector route is NP-complete. This section presents some heuristic based algorithms for designing the route to be followed by the mobile collector for collecting the data buffers of the sensor nodes requesting the collection service.

A heuristic algorithm provides feasible solution to an optimization problem, which may or may not be optimal. Good heuristics give solutions that are close to the optimal solution, and usually are efficient in terms of theoretical or practical running time. A desirable property of a heuristic is that the worst-case quality of the solution provided can be guaranteed. For example, some heuristics guarantee that the solution they identify is at worst η-optimal; that is, if the optimal cost is C*, the heuristic finds a solution no greater than ηC* for minimization problems. In our problem, the route constructed by ordering the requests based on a timely manner according to their arrival is considered as an upper bound on the solution provided by the heuristic.

4.1 Minimum Spanning Tree Route Construction

In the Traveling Salesman Problem (TSP) (Lawler et al., 1990) context, a vertex can be interpreted as a sensor and the edge weight can be the distance between the sensors or the time of travel between any two sensors. With these notations, the mobile collector waits until a certain number of requests m are received, and then the route construction is interpreted as the problem of finding a minimum-cost tour that visits each of the sleeping sensors exactly once and returns to the starting point (center of the sensing field). The route objective is expressed as follows:

$$\min \sum_{i=1}^{m} c(R(i), R(i+1)) \tag{8}$$

where $R(i)$ is the i^{th} request on the route; and $c(R(i), R(i + 1))$ is the distance cost from request i to request $(i + 1)$.

Defining $x_{ij} = 1$, if the edge from request i to request j is on the route, $i,j \in \{1,..., m\}$ and 0 otherwise, maps the objective to:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} * \chi_{ij} \tag{9}$$

subject to:

$$\sum_{j=1}^{m} \chi_{ij} = 1, \forall i \tag{10}$$

$$\sum_{i=1}^{m} \chi_{ij} = 1, \forall j \tag{11}$$

$$y_1 = 1, \tag{12}$$

$$2 \le y_i \le m, \forall i \ne 1 \tag{13}$$

$$y_i - y_j + 1 \le (m-1)(1 - x_{ij}), \forall i \ne 1, \forall j \ne 1 \tag{14}$$

$$x_{ij} \in \{0,1\}, \forall i, j \tag{15}$$

where
m: number of collection requests; *i*: current request, $i \in \{1,..., m\}$; *j*: next request, $j \in \{1,..., m\}$; and y_i : extra variable to exclude sub tours, $i \in \{1,..., n\}$.

Constraints (10) and (11) are called the degree constraints, which enforce that every sensor reached is left exactly once. Constraints (12), (13) and (14) are subtour elimination constraints, which prohibit the formation of subtours having less than *m* vertices.

It is necessary to show that a route with a Hamiltonian cycle has an overall time period $A_{\max}$ less than that without. Let $r_0, r_1, r_2, \ldots, r_m$ be the requests on a route, where r_0 is the center of the sensing field from which the collector route starts and ends. $c_{01}, c_{12}, c_{23}, \ldots, c_{m0}$ are the cost between consecutive requests. The period of the route is $T = \sum_{\forall c} c / v,$ where *v* is the mobile collector speed. If the route is a Hamiltonian cycle, $A_{\max}$ of any request location *r* is always equal to *T*. On the contrary, considering a route that is not a Hamiltonian cycle, the mobile collector will need to move from one end to another end, then back to the beginning request location to complete a cycle. The $A_{\max}$ of the request locations at two ends will then be *2T*, which is much longer then *T*.

Algorithm 1 presents the steps followed by the mobile collector for constructing its route. The minimum spanning tree is computed based, as presented by Algorithm 2, on partitioning the graph of the collection requests into two independent subsets, *S* and *T*, such that, $S \cup T = N$ and $S \cap T = \varnothing$ and the set (*S, T*) is given by all arcs (*i,j*), where $i \in S, j \in T$, or $i \in T, j \in S$.

```
Algorithm 1 Minimum Spanning Tree Route Construction Algorithm (MST-R)
Input:
$m$ collection requests, each with fields {$ID, X, Y$}
Body:
  1.  Build a fully connected graph $G$ for the m received collection requests.
  2.  Add the center of the sensing field to $G$.
  3.  Compute a minimum spanning tree $T$ for $G$.
  4.  Generate using Depth First Search (DFS) a list $L$ of vertices on $T$.
  5.  Construct the Hamiltonian cycle H for visiting the vertices $L$.
  6.  Follow the Hamiltonian cycle $H$ as the constructed route.
```

```
Algorithm 2 Minimum Spanning Tree Algorithm
Initialization:
$MST = \varnothing$
$S = \{1\}$, $T = N \backslash S$ (the set S contains node 1, while the set T contains all nodes except node 1)
Iterations:
while $| MST | < m$, do
  1- Find the arc $(i, j)$ in $(S, T)$ with minimum cost $c_{ij}$
  2- Add $(i, j)$ to MST
  3- Remove $j$ from $T$ and add $j$ to $S$
```

It is important to note that the list generated by the Depth-First Search (DFS) will produce each node twice as the DFS will visit each edge twice, once going down the tree when exploring it and once going up after exploring the entire sub tree. For example, in the depth-first search of Figure 3, the vertices in order *1-2-1-3-5-8-59-5-3-6-3-1-4-7-10-7-11-7-4-1* are visited, thus using every tree edge exactly twice. Therefore, this tour has weight twice that of the minimum spanning tree, and hence at most twice optimal. To remove the extra vertices, at each step a shortest path to the next unvisited vertex is taken. The shortcut tour for the tree in Figure 3 is *1-2-3-5-8-9-6-4-7-10-11-1*. This ensures that the tour gets shorter and within weight twice that of optimal.

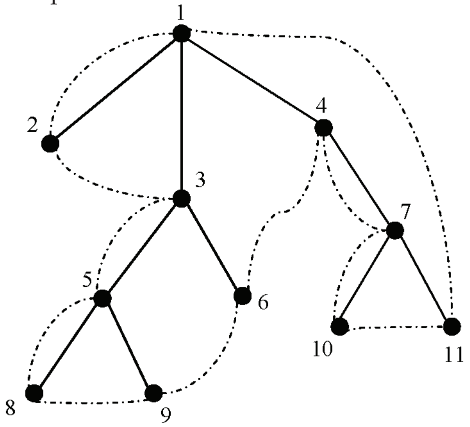


Fig. 3. A depth-first traversal of a spanning tree, with the shortcut tour

4.2 Dynamic Insertion Route Construction

The minimum spanning tree route construction algorithm considers only the distance cost when constructing the collection route. Instead, an algorithm can be designed which gives weights to the distance cost and the time which the sensor remains sleeping until the collection is done. Based on Equation 3, each collection request is inserted dynamically in its minimum insertion position on the collection route. Let r_1 ,r_2,... ,r_m be the requests to be collected by the mobile collector on its route R, and, C_{jk} is the extra cost of placing request *k* after request *j* on *R*, then it is required to:

$$\min \sum_{j=1}^{m} C_{jk} * X_{jk} \tag{16}$$

where X_{jk} = 1, if the placement of request *k* follows request *j*; 0 otherwise, and

$$\sum_{j=1}^{m} X_{jk} = 1 \qquad \text{for any k} \tag{17}$$

Algorithm 3 presents the procedure of finding the best insertion point on the mobile collector route for the current received collection request. The sensor sleeping time ST based on the point of insertion along the collection route is not computed absolute, but relative to the current time. Table 1 shows the effect of different α values. To illustrate the contents of the table, consider the scenario presented in Figure 4. Suppose the mobile collector just serviced node A and the next node on its route is node B. Before initiating the service to node B, a request of collection arrives from node C. The two nodes B and C have distance costs 5 and 15 respectively, and the sleeping time of node B is 10 time units. The insertion position of node C is either before node B or after node B. When $\alpha \le 0.5$, node C is inserted before node B and when $\alpha > 0.5$, node C will be inserted after node B. When $\alpha = 0$, the mobile collector will always favour newly arriving requests, which can result in high delays in servicing older requests, causing those nodes to wait long times until they are serviced.

```
Algorithm 3 Dynamic Insertion Route Construction Algorithm
Input:
Request ($r$) with fields {$ID, X, Y$}
Define:
$R$: current route consisting of $k$ requests; mp: mobile collector position;
$ST$ : sensor sleeping time according to the insertion point;
$r_f$ , $r_l$: first and last requests in R;
$C_b^a = \sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}$
Initialize:
Cost[1..(k+1)] = 0
Body:
IF (R = ∅) THEN $R = r$
ELSE
```

```
$ST_1 = C_{m_p}^{r_f}$
Cost[1] = $\alpha * (C_r^{m_p} + C_{r_f}^{r} - C_{r_f}^{m_p}) + (1-\alpha) * ST_1$
Repeat for $i$ = 2...$k$
    $ST_i = ST_i + C_{k_i}^{k_{i-1}}$
    Cost[i] = $\alpha * (C_r^{k_i} + C_{k_{i-1}}^{r} - C_{k_i}^{k_{i-1}}) + (1-\alpha) * ST_i$
$ST_{k+1} = ST_k + C_{r_l}^{r}$
Cost[k+1] = $\alpha * C_{r_l}^{r} + (1-\alpha) * ST_{k+1}$
Set $j$ = index of minimum in Cost [1..(k+1)]
Insert $r$ in $R$ at position $j$
END
```

4.3 Discussion

The complexity of the MST-Route construction algorithm is upper bounded by the complexity of computing the MST. The MST algorithm requires (m-1) iterations in which a single arc is identified in (S, T) and moved into the MST. Each iteration, scans each arc to determine whether it is in (S, T), and if so, if it has cost less than the current minimum. Such an implementation would scan n arcs each iteration, for a total complexity of O(mn). The DFS constructs the list of nodes in an O(m). Thus the MST-R requires O(m^2) running time. The DI-Route construction requires an O(m) to compute the cost of insertion along the current route and an O(log n) to find the minimum insertion point, resulting in an O(m) running time complexity. It should be noted that the DI-R construction works totally online and adapts the current route according to each received request, while the MST-R construction groups m received requests to construct the collection route.

α∈ [0,1]	Result
1	Insertion point is determined only by the distance cost.
0	Insertion point is determined only by the sensor sleeping time.
> 0.5	Distance cost contributes higher in the insertion point.
≤ 0.5	Sensor sleeping time contributes higher in the insertion point.

Table 1: Effect of a on the insertion point

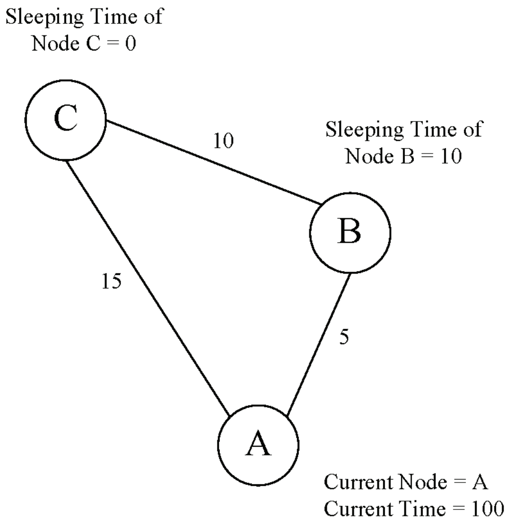


Fig. 4. An example to illustrate the effect of a on the mobile collector route

5. Experimental Methodology and Results

The previous section described some heuristics for constructing the route of the mobile collector. This section presents the evaluation of these algorithms through simulation.

5.1 Methodology

As the parameter space is huge, some parameters are fixed as follows:

- Sensing Field: A square area of dimension 100 x 100 m^2 is considered for the sensors deployment. A number of sensor nodes varying from 50 to 100 sensors are distributed uniformly at random preserving homogenous node density among all areas in the sensing field.

- Sensor Parameters: Each sensor has a radio communication radius of 25m and a buffer of size 1 Kbyte.

- Mobile Collector Parameters: Initially the mobile collector is localized at the center of the sensing field. The mobile collector moves with a fixed speed of 1m/s, and has a communication radius of 50m. The mobile collector achieves communication with the sensor node when it is in the sensor's radio communication range.

- Simulation Time: The simulation run last for 50000 time units, and all results are averaged over 20 different independent networks.

- Sensors Sampling Frequency: The sampling frequency determines the time required for the sensor's buffer to become full and therefore controls the distribution of the collection requests over the simulation time. The first option assumes that events occur regarding some point of interest located at the center of the sensing field. In such a case, a higher number of the requests are coming from a specific part of the network, as the sensors closer to the center sample more frequently. The requests distribution in this case follows Gaussian distribution with small variance. A concentric topology for the sensors sampling rates as shown in Figure 5 is used to model this. A sequence of *n* concentric circles divides the sensing field into several ring shaped regions; Region 1 to Region *n*. The radius of each concentric circle is denoted by $R_1, R_2, R_3,\ldots, R_n$, where $R_1 = 10m$. The value of each radius is calculated as:

$$R_i = i * R_1; \quad i = [1,\ldots,n] \tag{18}$$

where

$$n = \frac{L}{2 * R_1} + 1 \tag{19}$$

The sensors in the innermost region are assigned sensing rates in the range [1, *baserate*] and the sensing rates of sensors in regions radically outwards are calculated as:

$$Sensingrate_i = [\ baserate * (i - 1) + 1\ ,\ baserate * i\ ], \quad i = \{1,\ldots,n\} \tag{20}$$

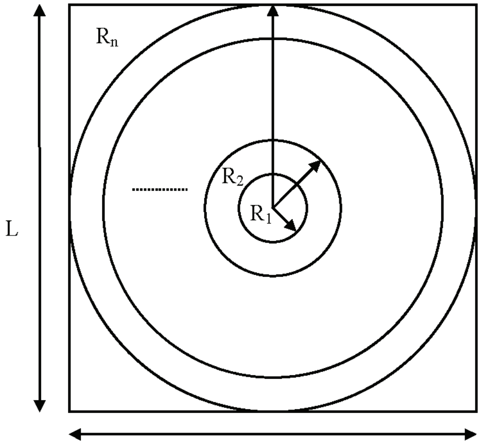


Fig. 5. Concentric Topology: The first type of topology considered in the simulation

Label	*Sensing Rate Topology*	*Number of Sensors*	*Baserate*
A1	Concentric	50	2 sec
A2	Concentric	75	3 sec
A3	Concentric	100	4 sec
B1	Random	50	2 sec
B2	Random	75	3 sec
B3	Random	100	4 sec

Table 2: Set of Experiments

When the baserate is chosen to be 2 secs, the sensors in region R_2 will have a sensing rate in the interval [3,4] which means that some sense a sample every 3 secs and others sense a sample every 4 secs.

The second option assumes random occurrence of events independently from one another in the sensing field. In this case the requests distribution follows Poisson distribution. The sensing rate of each sensor node is chosen randomly in the interval [1, *baserate* * *n*].

Putting everything together, the experiments shown in Table 2 are run. The experiments are labeled for referencing purposes. For each of these, results presented are an average of 20 runs. The parameters that change from run to run are the location of the sensor nodes, and correspondingly the distance cost and the sensing rates.

For the purposes of evaluation, the following performance metrics are considered:

- **Data Collection Time:** This is defined as the period from the time the sensor sends the collection request to the time of arrival of the mobile collector to collect the sensor's buffer. This is averaged across all the nodes in the network.

- **Request Collection Time:** This is computed as the overall data collection time to the number of requests collected from all nodes in the network.

- **Sleeping Sensors Ratio:** This measure is calculated as the number of sleeping sensors waiting the arrival of the mobile collector to the total number of sensors in the network over the simulation time of the experiment.

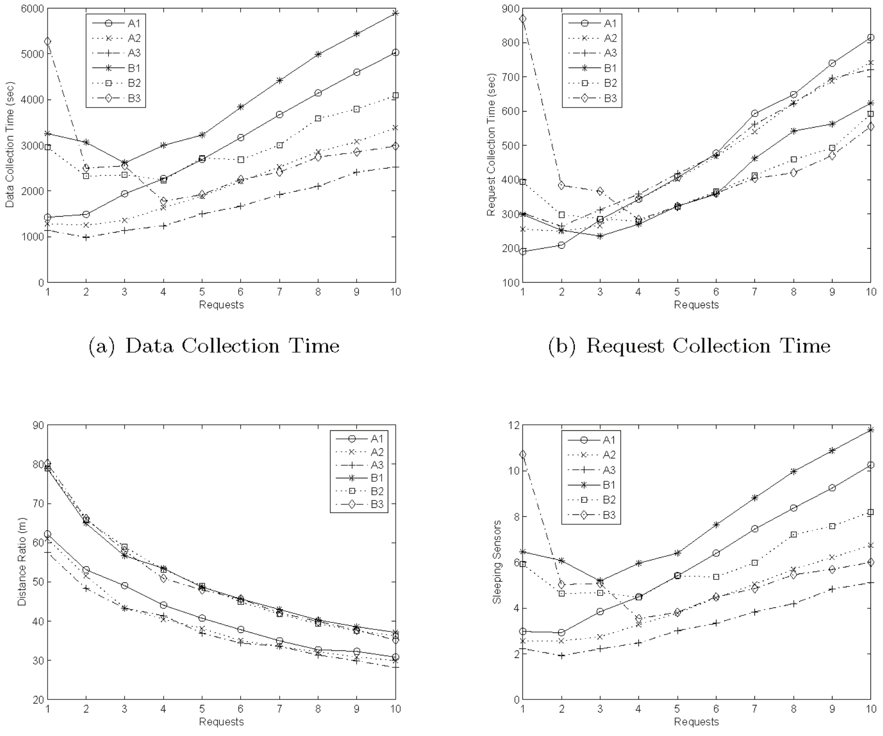
- **Distance Ratio:** This is defined as the ratio of the distance traveled by the mobile collector to the number of requests collected.

5.2 Performance of the MST-R

Figure 6(a) and 6(b) show the performance results of running the MST-R algorithm on the set of experiments described in Table 2. The number of requests m which the mobile collector uses for constructing its collection route ranged form 1 to 10. The case m = 1, is the case where the mobile collector service the arriving requests one by one in a timely oriented manner. As the mobile collector waits for more requests to arrive this increases the data collection time however in such a case the mobile collector service more requests on the same collection route which decrease its distance ratio as presented in Figure 6(c). Also, waiting for more requests to arrive would increase the average percentage of sensor nodes requesting the collection service as shown in Figure 6(d). The performance on A3 was better than A2 which was better than A1. This is obvious because these have sensing base rate of 2, 3, 4 secs respectively, and higher this quantity, lesser the requests arrival rate. Similar trends are observed in experiments B1-B3.

5.3 Effect of a on the DI-R

Figure 7(a) and 7(b) show the result of running the DI-R construction algorithm on the set of experiments described in Table 2. a ranged from 0 to 1 with steps of 0.5. The data collection time and the request collection time decrease as a goes near to 1. This is because the request is inserted on the mobile collector route in the position resulting the minimum extra distance, as explained in Table 1. Therefore, the forth and back traveling between nodes is minimized which results in less waiting times for the sensors, thus less sleeping sensors as in Figure 7(d).



(a) Data Collection Time

(b) Request Collection Time

(c) Mobile Collector Distance Ratio

(d) Percentage of Sleeping Sensors

Fig. 6. Performance of MST-R for Experiments A1-A3 and B1-B3

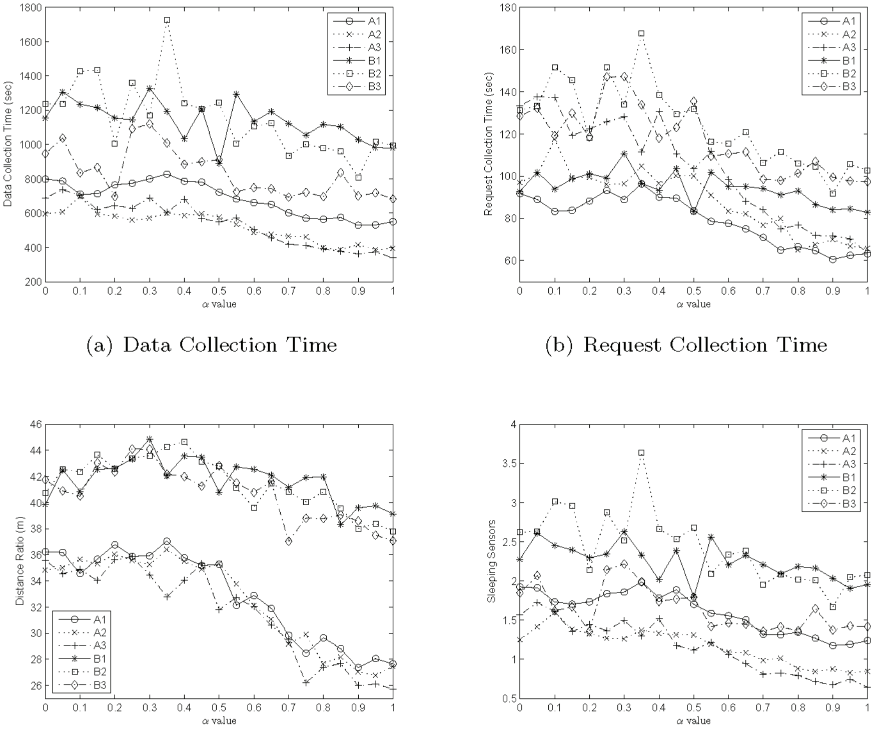
The performance on experiments A1-A3 is better than B1-B3. This is obvious because most of the events come from a specific part of the network which helps the mobile collector to optimize its collection route more than when the events are scattered as in experiments B1-B3. This also appears on the distance ratio of the mobile collector shown in Figure 7(c). As the events are scattered randomly over the sensing field, the mobile collector is required to travel more for the collection which appears on the distance ratio for experiments B1-B3. It is hard to conclude about the dependence of α value on the results, however, α around 0.9 leads to minimum results for the performance metrics described.

5.4 Impact of the Speed of the Mobile Collector

To get an insight into the relative performance of the two algorithms, two constrained topologies are used to show the impact of the mobile collector speed on the data collection time and the request collection time. The topologies considered employ 100 sensor nodes uniformly distributed in a square area of 100 x 100 (m^2). A concentric and random sensing rate topology is used with the sensing base rate equals 2 secs. These are labeled as Topology A and Topology B. Figure 8(a) and 8(b) plot the performance of the two algorithms for different speed values for the mobile collector. The DI-R construction algorithm outperforms the MST-R construction algorithm, however, when the speed of the mobile collector increase, this difference vanishes. Also, previously the algorithms performed better on the concentric sensing rate topology than on the random topology, this does not hold when the speed of the mobile collector is increased.

6. Conclusion and Future Directions

Sensor networks operate under limited energy constraints. Eliminating the relaying overheads can extend the sensor lifetime and prolong the network operational time. In this context, mobile elements (robots) are utilized by acting as mechanical data carriers for collecting the sensory information by approaching physically the sensor node. This chapter presented some heuristics for constructing the mobile collector collection route. The algorithms performance are shown and their impact on the data collection operation is presented. There are many directions in which this work may be pursued further. Statistical measures are required to measure the buffer filling rate and thus the sensor can send its collection request before its buffer is full, which gives an extra advantage for the mobile collector. Applying multiple mobile collectors can enhance the performance. Control schemes for coordinating multiple collectors need to be designed efficiently to maximize the performance.



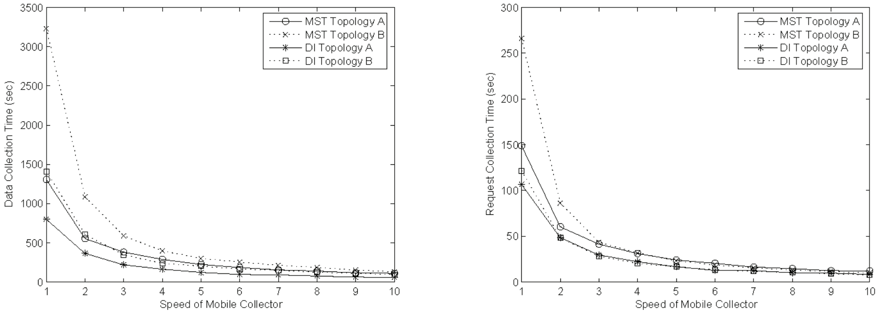
(a) Data Collection Time

(b) Request Collection Time

(c) Mobile Collector Distance Ratio

(d) Percentage of Sleeping Sensors

Fig. 7. Performance of DI-R for Experiments A1-A3 and B1-B3



(a) Data Collection Time

(b) Request Collection Time

Fig. 8. Relative Performance for Topologies A and B

References

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. Computer Networks, 38(4):393-422, 2002.

P. Baruah, R. Urgaonkar, and B. Krishnamachari. Learning enforced time domain routing to mobile sinks in wireless sensor fields. In Proceedings of the 1st IEEE EMNETS, 2004.

L. D. Bodin, B. L. Golden, A. A. Assad, and M. Ball. Routing and scheduling of vehicles and crews, the state of art. Computers and Operations Research, 10:63212, 1983.

J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. IEEE Pervasive Computing, 3(1):38-45, 2004.

A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In 2nd International Workshop on Information Processing in Sensor Networks (IPSN), 2003.

Chee-Yee Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. Proceedings of the IEEE, 91(8):1247-1256, 2003.

N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12:568-581, 1964.

D. Culler, D. Estrin, and M. Srivastava. Introduction: Overview of sensor networks. IEEE Computer, 37(8):41-49, 2004.

G. Engelberger. Services. Shimon Y. Nof, ed., Handbook of Industrial Robotics. John Wiley & Sons, New York, second edition, 1999.

M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: W. H. Freeman, 1979.

G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. European Journal of Operational Research, 151(1):1-11, 2003.

M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. IEEE/ACM Transactions on Networking, 10(4):477-486, 2002.

Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. Lee. Partitioning based mobile element scheduling in wireless sensor networks. In Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON), 2005.

M. Haenggi. Opportunities and challenges in wireless sensor networks. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, pages 1.1-1.14,
2004.

S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. Mobile Networks and Applications, 11(3):327-339, 2006.

A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys), 2004.

R. M. Karp. Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. Mathematics ofOperations Research, 2:209-224, 1977.

G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. European Journal ofOperational Research, 59:231-247, 1992.

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, 1990.

E. C.-H Ngai, J. Lin, and M. R. Lyu. Delay-minimized route design for wireless sensor-actuator networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2007.

D. Rosenkrantz, R. E. Sterns, and P. M. Lewis. An analysis of several heuristics for the traveling salesman problem. SIAM Journal on Computing, 6:563-581, 1977.

R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks, 1(2-3): 215-233, 2003.

M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research, 35(2), 1987.

A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In 25th IEEE International Real-Time Systems Symposium, 2004.

A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava. Mobile element scheduling with dynamic deadlines. IEEE Transactions on Mobile Computing, 6(4):395-410, 2007.

Y. Tirta, B. Lau, N. Malhotra, S. Bagchi, Z. Li, and Y. Lu. Controlled mobility for efficient data gathering in sensor networks with passively mobile nodes. Sensor Network Operations, 3.2:Wiley-IEEE Press, 2006.

P. Toth and D. Vigo, editors. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. ISBN 0-89871-498-2.

O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. IEEE Transactions on Mobile Computing, 3: 366-379, 2004.

W. Zhao and M. H. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS), 2003.

W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc), 2004.

W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2005.

12

An Augmented Reality Human-Robot Collaboration System

Scott A. Green*, ** J. Geoffrey Chase* XiaoQi Chen* and Mark Billinghurst**
**University of Canterbury, Department of Mechanical Engineering*
Christchurch, New Zealand
***Human Interface Technology Laboratory New Zealand (HITLab NZ)*
University of Canterbury
Christchurch, New Zealand

1. Introduction

Interface design for Human-Robot Interaction (HRI) will soon become one of the toughest challenges that the field of robotics faces (Thrun 2004). As HRI interfaces mature it will become more common for humans and robots to work together in a collaborative manner. Although robotics is well established as a research field, there has been relatively little work on human-robot collaboration.

There are many application domains that would benefit from effective human-robot collaborative interaction. For example, in space exploration, recent research has pointed out that to reduce human workload, costs, fatigue driven errors and risks, intelligent robotic systems will need to be a significant part of mission design (Fong and Nourbakhsh 2005). Fong and Nourbakhsh also observe that scant attention has been paid to joint human-robot teams, and that making human-robot collaboration natural and efficient is crucial to future space exploration. Effective human–robot collaboration will also be required for terrestrial applications such as Urban Search and Rescue (US&R) and tasks completed robotically in hazardous environments, such as removal of nuclear waste.

There is a need for research on different types of HRI systems. This chapter reports on the development of the Augmented Reality Human-Robot Collaboration (AR-HRC) system (Green, Billinghurst et al. 2008). Fundamentally, this system enables humans to communicate with robotic systems in a natural manner through spoken dialog and gesture interaction, using Augmented Reality technology for visual feedback. This approach is in contrast to the typical reliance on a narrow communication link.

Truly effective collaboration among any group can take place only when the participants are able to communicate in a natural and effective manner. Communicating in a natural manner for humans typically means using a combination of speech, gesture and non-verbal cues such as gaze. Grounding, the common understanding between conversational participants (Clark and Brennan 1991), shared spatial referencing and spatial awareness are well-known crucial components of communication and therefore collaboration.

In a collaborative team effort it is also important to capitalize on the strengths of each member of the team. For example, humans are good at problem solving and dealing with unexpected events, while robots are good at repeated physical tasks and working in hazardous environments. An effective human-robot collaboration system should therefore exploit these strengths. The AR-HRC system does just that by enabling the human and robot to discuss a plan, review it and make adjustments to it. It then lets the robot execute the plan with interaction from the human team member, if or as, warranted. If an unexpected situation arises the robot can discuss the problem with its human partner and arrive at a solution agreeable to both, similar to the collaborative behaviour in human teams.

Augmented Reality (AR) is a technology for overlaying three-dimensional virtual graphics onto the users view of the real world (Azuma 1997). AR allows real time interaction with these virtual graphics, enabling a user to reach into the augmented world and manipulate it directly. AR is used in this research to provide a common 3D graphic of the robot's workspace that both the human and robot reference.

The internal state of the robot and its intended actions are displayed through the virtual imagery in the AR environment. The human team member is thus able to maintain situational awareness of the robot and its surroundings, thereby giving the human-robot team the ability to ground their communication. By coupling AR with spoken dialog a multi-modal interface has been developed that enables natural and efficient communication between the human and robot team members, thus enabling effective collaboration.

In this chapter, the development of the Augmented Reality Human-Robot Collaboration (AR-HRC) system is discussed. Related work is reviewed and its influence on the design of the AR-HRC system is discussed. AR is introduced, and its benefits in terms of HRC are reviewed. The initial development of an AR multi-modal interface is discussed and then the architectural design of the AR-HRC system is presented. A case study incorporating a mobile robot into the AR-HRC system is then presented and the robustness and effectiveness of this system is evaluated in a performance experiment. The chapter ends with a set of summary conclusions and a presentation of future research directions.

2. Related Work

2.1 Human-Robot Interaction

In this work, collaboration is defined as "working jointly with others or together especially in an intellectual endeavor". Clark and Brennan provide a communication model to interpret collaboration (Clark and Brennan 1991). In this model, conversation participants attempt to reach shared understanding or common ground. Common ground refers to the set of mutual knowledge, shared beliefs and assumptions that collaborators have. Therefore, an effective human-robot collaborative team needs to be able to easily reach common ground.

Milgram *et al* (Milgram, Zhai et al. 1993) highlighted the need for combining the attributes that humans are good at with those that robots are good at to produce an optimized human-robot team. Milgram *et al* pointed out the need for HRI systems that can transfer the interaction mechanisms that are considered natural for human communication to the precision required for machine information.

Adjustable autonomy, enabling the system to vary the level of robot autonomy, increases performance and is an essential component of an effective human-robot collaborative system. The ability to vary the level of robot autonomy has been seen to improve performance (Tsoukalas and Bargiotas 1996; Ishikawa and Suzuki 1997; Bechar and Edan 2003). Varying the level of autonomy of human-robotic systems allows the strengths of both the robot and the human to be maximized. It also enables the system to optimize the problem solving skills of a human and effectively balance that with the speed and physical dexterity of a robotic system.

By giving the robotic system the ability to ask help of its human counterpart, performance is improved through the addition of human skills, perception and cognition. The system also benefits from human advice and expertise (Fong, Thorpe et al. 2002). Adjustable autonomy enables the robotic system to better cope with unexpected events such as being able to ask its human team member for help when necessary. In essence, adjustable autonomy gives the robot the ability to act as a collaborative partner.

Situational awareness, or being aware of what is happening in the robot's workspace, is also essential in a collaborative effort. The lack of situational awareness decreases the performance of human-robot interaction (Murphy 2004; Yanco, Drury et al. 2004). In more extreme cases, the lack of situational awareness can result in a dangerous collision as was found by Ellis (Ellis 2000). In particular, when the unmanned space ship Progress collided with Mir, Ellis found that the lack of situational awareness was one of the key cognitive factors that caused the collision. Thus, improving situational awareness in human-robot interaction also improves performance (Scholtz, Antonishek et al. 2005).

The design of an effective human-robot collaboration system must afford the use of natural speech (Nourbakhsh, Bobenage et al. 1999; Kanda, Ishiguro et al. 2002). However, speech alone is not enough to complete the grounding process. Therefore, a multi-modal approach must be taken (Huttenrauch, Green et al. 2004; Sidner and Lee 2005) which will enable the human-robot team to communicate in a similar manner to that of a human-human team. According to Clark and Brennan (Clark and Brennan 1991) the process of grounding involves communication using a range of modalities, including voice, gesture, facial expression and non-verbal body language.

Fong *et al* (Fong, Kunz et al. 2006) note that for humans and robots to work together as peers, the human-robot system must provide mechanisms for both humans and robots to communicate effectively. For example, to enable the use of spatial dialog, it is an inherent requirement that common frames of references be used (Skubic, Perzanowski et al. 2004). A robot should reach a common understanding in communication by understanding the conversational cues used by humans, such as gaze direction, pointing and gestures.

2.2 Augmented Reality

Augmented Reality (AR) has many benefits that help to create a more ideal environment for human-robot collaboration (Green, Billinghurst et al. 2008). In a typical AR set-up, the user wears a head mounted display (HMD) with a camera mounted on it. The output from the camera is fed into a computer, augmented with 3D graphics and then fed back into the HMD. The user sees an enhanced view of the real world through the video image in the HMD, as shown in Figure 1. Therefore, AR blends virtual 3D graphics with the real world in real time (Azuma 1997). This type of AR set-up is commonly called a video-see-through AR interface.

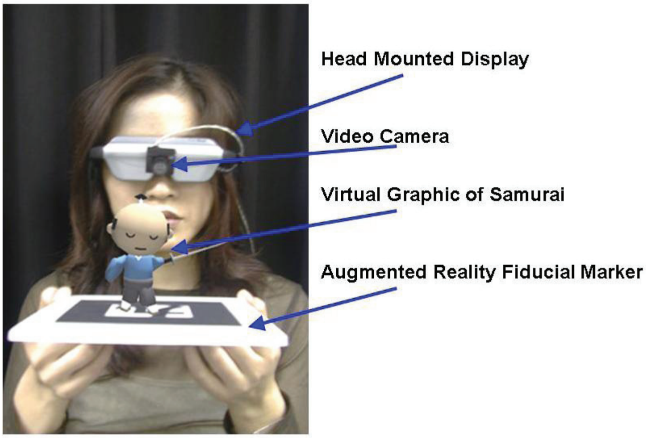


Fig. 1. Video-see-through AR interface (Billinghurst, Poupyrev et al. 2000).

Square fiducial patterns are placed in the real environment with a unique symbol in the middle of each pattern. Computer vision techniques are then used to identify the unique symbols, calculate the camera position and orientation from these symbols, and display 3D virtual images aligned with the position of the fiducial patterns (ARToolKit 2008). This augmented view is then fed into the HMD providing the user with a seamless combination of the real world view and virtual graphics, where the virtual images appear fixed to the fiducial patterns. The ability to manipulate the physical markers with fiducial patterns on them enables direct real-time interaction with the 3D virtual content.

Multiple users can view the same fiducial patterns and therefore have their own perspective of the 3D virtual content. Since the users see each other's facial expressions, gestures and body language, AR therefore supports natural face-to-face communication. This interaction demonstrates that a 3D collaborative environment enhanced with AR content can seamlessly enhance face-to-face communication and allow users to naturally work together (Billinghurst, Poupyrev et al. 2000; Billinghurst, Kato et al. 2001). Shared visual workspaces of this type have been shown to enhance collaboration as they increase situational awareness (Fussell, Setlock et al. 2003).

AR also improves collaboration by allowing the use of physical tangible objects for ubiquitous computer interaction. Thus making the collaborative environment natural and effective by allowing participants to use objects for interaction that they would normally use in a collaborative effort (Billinghurst, Grasset et al. 2005). AR provides rich spatial cues permitting users to interact freely in space, supporting the use of natural spatial dialog (Billinghurst, Poupyrev et al. 2000).

Bowen *et al* (Bowen, Maida et al. 2004) showed through user studies that the use of AR resulted in significant improvements in robotic control performance. Similarly, Drury *et al* (Drury, Richer et al. 2006) found that for operators of Unmanned Aerial Vehicles (UAVs) augmenting real-time video with pre-loaded map terrain data resulted in a significant difference in the comprehension of 3D spatial relationships compared to 2D video alone. The AR interface provided better situational awareness of the activities of the UAV.

Providing the human with an exo-centric view of the robot and its surroundings enables the human to maintain situational awareness of the robot and gives the human-robot team the ability to ground their communication. AR can thus provide a 3D world, within which both the human and robotic system can operate, a shared space (Billinghurst, Poupyrev et al. 2000). This use of a common 3D world provides common reference frames for both the human and robot.

AR supports the use of spatial dialog and deictic gestures, allows for adjustable autonomy, supports multiple human users, and allows the robot to visually communicate to its human collaborators its internal state through the use of graphic overlays on the real world view of the human. AR also enables a user to experience a tangible user interface, where physical objects can be manipulated to affect changes in the shared 3D scene (Billinghurst, Grasset et al. 2005), thus allowing a human to reach into the 3D world of the robotic system and manipulate it in a way the robotic system can understand. Therefore, by taking advantage of the benefits of AR, a robust human-robot collaboration system can be more effectively created.

3. Multi-modal Augmented Reality Interaction

To aid in the development of a multi-modal interaction framework required for the AR-HRC system the first step was to investigate previous multi-modal research and create a multi-modal AR application. One of the first interfaces to support speech and gesture recognition in a multi-modal interface was the Media Room (Bolt 1980). The Media Room allowed the user to interact with a computer through voice, gesture and gaze. Bolt's work showed that gestures combined with natural speech (multi-modal interaction) lead to a powerful and more natural human machine interface.

Speech and gesture compliment each other and when used together create an interface more powerful than either modality alone. The difficulty of using speech alone was demonstrated by Kay (Kay 1993) who constructed a speech driven interface for a drawing program. Even simple cursor movements around the screen required a time consuming combination of continuous and discrete vocal commands.

Tangible User Interfaces (TUIs) use real-world objects as the interaction devices for a software program or computer (Ishii and Ullmer 1997). Therefore, TUIs are extremely intuitive to use because physical object manipulations are mapped one-to-one to virtual object operations (Fitzmaurice and Buxton 1997). Another benefit of a TUI is that it naturally supports sharing and collaboration.

TUIs are a viable approach for interaction with AR applications as they enable users to interact naturally by manipulating real world objects. Thus, the principles of TUIs can be combined with AR's display capabilities in an interface metaphor known as Tangible Augmented Reality (TAR) (Kato, Billinghurst et al. 2001). A TAR interface supports the presentation of 3D virtual objects anywhere in the physical environment, while simultaneously allowing users to interact with this virtual content using real world physical objects (Kato, Billinghurst et al. 2000). An ideal TAR interface facilitates seamless display and interaction, removing the functional and cognitive seams found in traditional AR and TUI interfaces.

To investigate these types of multi-modal interaction in an AR environment, a system was developed that used speech and paddle based gestures as input to an AR system. The system is a modified version of the VOMAR application for tangible manipulation of virtual furniture in an AR setting (Kato, Billinghurst et al. 2000). The VOMAR application used a single modality with a paddle input device that allowed the user to perform different tasks by using real world paddle movements to manipulate virtual objects in an AR setting.

The objective of the Multi-modal AR System (MARS) was to allow people to easily and effectively arrange AR content using a natural mixture of speech and gesture input. A single fiducial marker located on the end of the paddle allows the ARToolKit library to efficiently locate the paddle's position and orientation in the AR environment. A4 sized pages containing an array of fiducial markers serve as menu pages holding virtual furniture models, see Figure 2. A separate sheet also containing AR fiducial markers serves as the workspace. This page displays a 3D graphic of an empty room where the virtual models of furniture are to be placed. The square fiducial markers printed on these pages are used by the ARToolKit library to locate and place the virtual content.

Furniture, for example, can be arranged in a room by selecting various pieces of virtual furniture and placing them in the virtual room, as seen in Figure 3. As the user looks at each of the A4 container pages through a Head Mounted Display (HMD), they see different sets of virtual furniture. The 3D virtual models appear superimposed over the real pages aligned with the fiducial markers. Looking at the workspace page for the first time the user sees an empty virtual room. The user is able to transfer objects from the menu pages to the virtual room using paddle and speech commands.

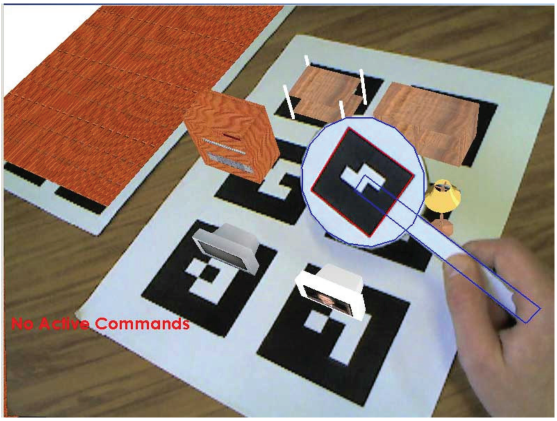


Fig. 2. An A4 sized sheet with an array of square fiducial markers is used as a menu page containing various pieces of furniture. The user can be seen holding the real world paddle that is used to interact with the virtual content. This is the view seen in the user's HMD.

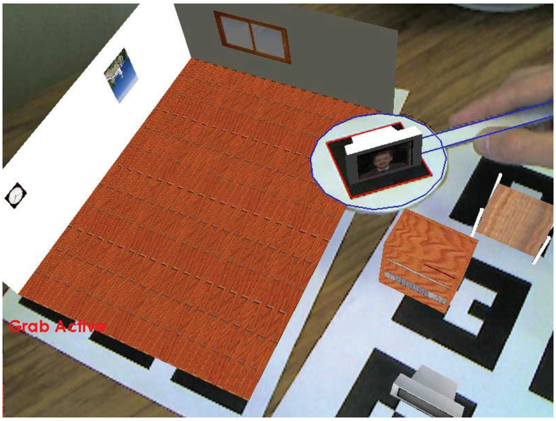


Fig. 3. A separate sheet containing virtual markers locates the virtual room. Shown here is the view the user sees with the room initially empty and a virtual object ready to place in the room.

The user can also modify the position and orientation of furniture already in the virtual room through the use of the real world paddle and speech input. The system provides visual and audio feedback to the user. The speech interpretation result is shown on the screen and audio feedback is provided after the speech and paddle gesture command. Therefore, the user is immediately notified when the system recognizes a speech or gesture.
The MARS architecture is shown in Figure 4. The speech processing module recognizes the spoken dialog of the user and is also responsible for the text to speech (TTS) of the application. The Dialog Management System (DMS) compares the spoken dialog that the speech processor recognized and parsed with predefined goals for the system. The speech processor and DMS make use of the Ariadne spoken dialog system (Denecke 2002).

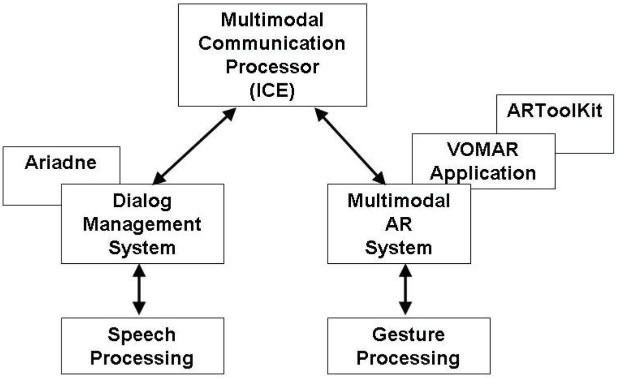


Fig. 4. Architecture for the MARS application.

When a goal has been reached through spoken dialog, the DMS sends the appropriate command to the MARS module via the Multi-modal Communication Processor (MCP). The MCP is built upon the Internet Communications Engine (ICE) (ZeroC 2008). If the command sent is to be combined with a gesture, the MARS module checks for the paddle to be in the users view and calculates its position. At the same time the MARS module calculates the location of all virtual objects, whether they are on the menu pages or in the virtual room, and compares these locations to that of the paddle. When the paddle is within a certain distance of a virtual piece of furniture, this piece of furniture becomes active and the verbal command sent in from the DMS is applied to the selected piece of furniture. The MARS module is written in C++ and uses the VOMAR and ARToolKit (ARToolKit 2008) libraries.

Thus, a user may combine speech commands and paddle gestures to interact with the system. To understand the combined speech and gesture, the system must fuse input from both streams into a single understandable command. This fusion is achieved by recording the time each input occurs, as given by an event time stamp. The paddle and speech input can therefore be considered for fusion only if the input time stamps from both input streams are within a certain time window of each other.

The multi-modal AR application is capable of working in three different modes, a gesture only mode, speech input with static paddle placement, and full paddle gesture combined with speech. The gesture only mode works essentially the same as the initial VOMAR application. In this mode, the user interacts with the system through paddle commands only and as a result no explicit fusion strategy is needed.

The gesture only mode consists of a variety of interaction techniques. The user is able to pick an object from the menu page with an empty paddle by holding the paddle close to the object. If the paddle is empty and placed on a virtual object, the object is picked up after the paddle has remained in this position for a short period of time.

If there is an object attached to the paddle, when the paddle is tilted the object will slide off the paddle and into the empty virtual room. If there is an object attached to the paddle, the object will be deleted from the paddle when the paddle is shaken from left to right. An object already placed in the room can be moved around by “pushing” it with the paddle. An item of furniture that is placed in the room can be deleted by hitting it with the paddle.

Once an object is on the paddle, it can be picked up and viewed from any viewpoint. These interactions are very natural to perform with the real paddle, so in a short period of time a user can assemble a fairly complex arrangement of virtual furniture. However, placement of the virtual furniture is this manner is not very precise.

A second mode of interaction for the system is to use speech combined with static paddle placement. In this interaction mode the user interacts with the virtual content using speech and paddle placement. However, the system only considers the static paddle pose at a particular time and fuses this information with the speech recognition result to interpret the combined speech and gesture commands.

This mode works as follows, when a speech command is recognized it is checked against a set of goals. If a match is found, the appropriate command id number is sent to the AR application. For example, consider the speech input "grab this" while the user has placed the paddle in the proximity of a virtual object on one of the menu pages. The system will check the paddle position, if the paddle is close enough to the object, the object will be grabbed, or selected from the contents page and placed on the paddle for further action. If the paddle position is not close enough, the object will not be grabbed. However, the grab command

remains active for five seconds. So, the user can move the paddle closer to the object. If the paddle is moved to within the proximity limit and the difference between the current time stamp and previous speech input time stamp is five seconds or less, the object will be "grabbed", otherwise the user has to repeat the speech command.
A list of speech commands the system can process are given below:

Delete Command:
This command will delete an object from the paddle or from the workspace. If there is an object on the paddle, it will be deleted. If there is no object on the paddle and the workspace is in view, the object the paddle is touching will be deleted.

Translate Command:
If the workspace is in view, this command attaches a virtual object in the workspace to the paddle so that it follows the paddle translation. The object will be released from the paddle after the user gives the *Stop* or *Place* command.

Rotate Command:
This command has a similar function as the Translate command. It attaches a virtual object in the workspace to the paddle so that it can follow the paddle rotation. The object will be released from the paddle after the user gives the *Stop* or *Place* command.

Move Command:
This command combines the Translate and Rotate commands. It attaches a virtual object from the workspace to the paddle so that it can follow both the paddle translation and rotation. The object will be released from the paddle after the user gives the *Stop* or *Place* command.

Place Command:
If there is an object attached to the paddle, this command places the attached object at the paddle location in the workspace.

Stop Command:
This resets a *Delete*, *Translate*, *Rotate* or *Move* command.

The final mode of interaction is full paddle gesture combined with speech. The user is able to interact with the system using both speech and continuous paddle gestures. This mode is a combination of the two modes explained previously. For example, the user can give a speech command "grab this" and then place the object using the paddle tilting gesture. In this manner, the user can easily combine speech and paddle gesture input and choose which interaction technique is more appropriate for the given task.
A user study was conducted to determine if the multi-modal interface actually improved the efficiency of user interaction in an AR environment. The results of this study did indeed show that a multi-modal interface improved performance for given tasks in an AR environment (Irawati, Green et al. 2006). These positive study outcomes provided the impetus to incorporate this type of interaction into the AR-HRC system.

4. AR-HRC System Architectural Design

A multi-modal approach has been taken for the architectural design of the AR-HRC system. This architecture combines speech and gesture through the use of AR that allows humans to naturally communicate with robotic systems. Through this architecture the robotic system receives the discrete information it needs to operate while allowing human team members to communicate in a natural and effective manner by referencing objects, positions, and intentions through natural gesture and speech. The human and the robotic system each maintain situational awareness by referencing the same shared 3D visuals of the workspace in the AR environment.

The architectural design is shown in Figure 5. The speech-processing module recognizes human speech and parses this speech into the appropriate dialog components. When a defined dialog goal is achieved through speech recognition, the required information is sent to the Multi-modal Communication Processor (MCP). The speech-processing module also takes information from the MCP and the robotic system and synthesizes this speech for effective dialog with human team members. As opposed to the earlier approach using a third party dialog system, it was decided to create a speech processing module catered to the specific needs of the AR-HRC. This speech processing module is based upon the Microsoft Speech SAPI 5.1 (MicrosoftSpeech 2007).

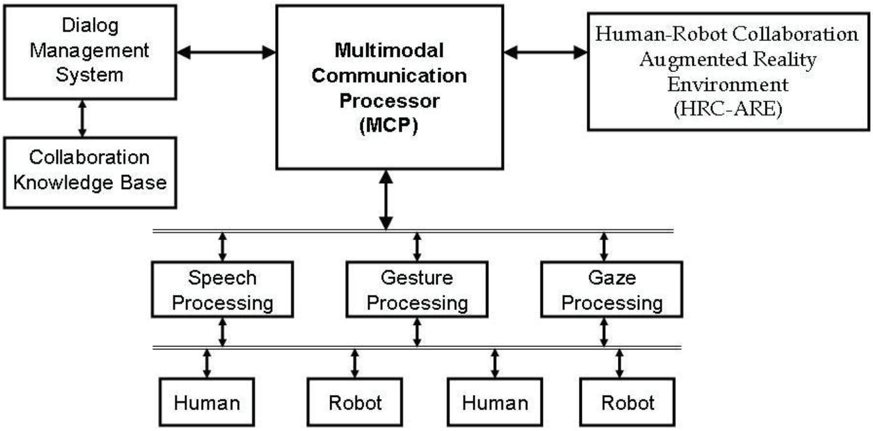

Dialog Management System
Collaboration Knowledge Base
Multimodal Communication Processor (MCP)
Human-Robot Collaboration Augmented Reality Environment (HRC-ARE)
Speech Processing
Gesture Processing
Gaze Processing
Human
Robot
Human
Robot

Fig. 5. The AR-HRC system architecture.

Gesture processing enables the human team members to use deictic referencing and normal gestures to communicate effectively with a robotic system. It is imperative that the system be able to translate the generic references that humans use, such as pointing into 3D space and saying "go here", into the discrete information a robotic system needs to operate. The gesture-processing module recognizes human gestures and passes this information to the MCP.

The MCP combines the speech from the speech processing module, the gesture information from the gesture-processing module and uses the Human-Robot Collaboration Augmented Reality Environment (HRC-ARE) to effectively resolve ambiguous deictic references such as *here, there, this* and *that*. The HRC-ARE is built upon the OSGART libraries (Looser, Grasset et al. 2006). OSGART is an Open Scene Graph (Open Scene Graph 2008) wrapper for the ARToolKit (ARToolKit 2008) and was selected for use in the AR-HRC for its high level rapid prototyping approach to creating virtual content for AR environments.

The disambiguation of the deictic references is accomplished in the AR environment's 3D virtual replication of the robot's world. The human uses a real world paddle to reach into and interact with this 3D virtual world. This tangible interaction, using a real world paddle to interact with the virtual 3D content, is a key feature of AR that makes it an ideal platform for HRC.

The gaze-processing module interprets the users gaze through the use of a camera mounted on a head mounted display (HMD). Through the use of the computer vision algorithms in the ARToolKit, the gaze direction of the human into the 3D world can be computed. The use of individual HMDs enables multiple human team members to view the HRC-ARE from their own perspective.

This personal viewing of the workspace results in increased situational awareness. Each team member views the work environment from his or her own perspective. They can change perspective simply by moving around the 3D virtual environment as they would a real world object, or they could move the 3D virtual world and maintain their position by moving the real world fiducial marker that the 3D world is "attached" to.

Not only do the human team members maintain their perspective of the robotic system's work environment, but they are also able to smoothly switch to the robot's view of the work environment. This ability to smoothly switch between an exo-centric (God's eye) view of the work environment to an ego-centric (robotic system's) view of the work environment is another feature of AR that makes it ideal for HRC and enables the human to quickly and effectively reach common ground and maintain situational awareness with the robot.

The Dialog Management System (DMS) is tasked to be aware of the communication that needs to take place for the human and robot to collaboratively complete a task. The MCP takes information from the speech, gesture and gaze processing modules along with information generated from the HRC-ARE and supplies it to the DMS. The DMS is responsible for combining this information and comparing it to the information stored in the Collaboration Knowledge Base (CKB).

The CKB contains information pertaining to what is needed to complete the desired tasks that the human-robot team wishes to complete. The DMS then responds through the MCP to either human team members or the robotic system, whichever is appropriate, facilitating dialog and tracking when a command or request is complete.

The MCP is responsible for receiving information from the other modules in the system and sending information to the appropriate modules. The MCP is thus responsible for combining multi-modal input, registering this input into something the system can understand and then sending the required information to other system modules for action. The result of this system design is that a human is able to use natural speech and gestures to collaborate with robotic systems.

5. A Case Study

In this section, the implementation of the AR-HRC with a mobile robot is described. This implementation enables a user to communicate with a mobile robot using natural speech and gestures. An example of this type of exchange would be the human gesturing to a point in 3D space and saying “go here” or “go behind that”.

The AR-HRC system takes this ambiguous information and translates it into discrete information for the robot to process and act upon. The robot then responds using speech and portraying its plans in the 3D virtual content of the AR environment. In this manner the human is able to understand the intentions and beliefs of the robot collaborative partner.

When the robot responds using speech, it randomly selects from a list of possible responses appropriate for the given situation. The situations for which the robot responds verbally are predetermined, thus when one of these situations arises, the system randomly selects a response that has been defined for that given situation. By issuing random responses the dialog of the robotic system feels more natural to the human. If the robot were to use the same phrase each time it needed to respond to a given situation, it would feel unnatural to the human and detract from the feeling of being collaborative partner.

As described previously, the use of AR technology enables the human collaborative partner to use natural speech and gestures. The AR environment also provides a common 3D spatial reference for both the robot and human, thus providing a means for grounding of communication and maintaining spatial awareness.

During implementation of the AR-HRC system with the mobile robot the gesture processing was developed to be modal. Verbal commands issued by the human determine which modality the real world paddle will be used in. The paddle can be used as a pointer enabling the human to point into the 3D virtual world of the robot and select a point or object. A second modality lets the human use the paddle for natural gestures.

Natural gestures have been defined from those used by participants in a Wizard of Oz study that was conducted to determine what kind of natural speech and gestures would be used to collaborate with a mobile robot (Green, Richardson et al. 2008). Natural gestures have been defined to communicate to the robot to move straight forward, turn in place, move forward while turning, back up and stop. At any time the user can issue a verbal command resulting in a true multi-modal experience.

The real world paddle has a fiducial marker on the end opposite the handle. The paddle is flat and has a fiducial marker on both sides. By placing markers on both sides of the paddle the vision system will be able to see the marker no matter which way the user holds the paddle. In the pointer mode a virtual pointer appears attached to the paddle. Therefore when the user moves the real world paddle around the virtual pointer follows the motion of the real world paddle.

The virtual pointer is the visual cue used by the human to select locations and objects in the virtual world. When the virtual pointer intersects other virtual content in the scene it is occluded thus providing the user with the cues necessary to determine precisely the point or object selected by the virtual pointer. But when the paddle is used for natural gestures the virtual pointer does not appear. Instead different visual indicators appear to let the user know what command they are giving. If the user holds the paddle straight out in front of them it is interpreted as a go forward gesture and an icon appears alerting the user of this.

The AR-HRC calculates the position and orientation of the fiducial marker on the paddle. The system then determines the orientation of the paddle relative to the user’s point of view.

If the paddle is held straight out in front of the user then the orientation angles of the paddle will fall within certain defined threshold values indicating to the system that a move straight forward command has been issued.

Similarly, when the paddle is moved to either side of straight in front of the user the system calculates the angle from straight ahead and converts this information into a turn. To turn the robot in place the user starts from the straight up position and rotates their arm about their elbow to the right or left. To go in the reverse direction the user places the paddle in a straight up position. Any position of the paddle not specifically defined is interpreted as a stop command and is relayed to the user by displaying a stop sign on the paddle. See Figure 6. for various paddle-gesture commands.

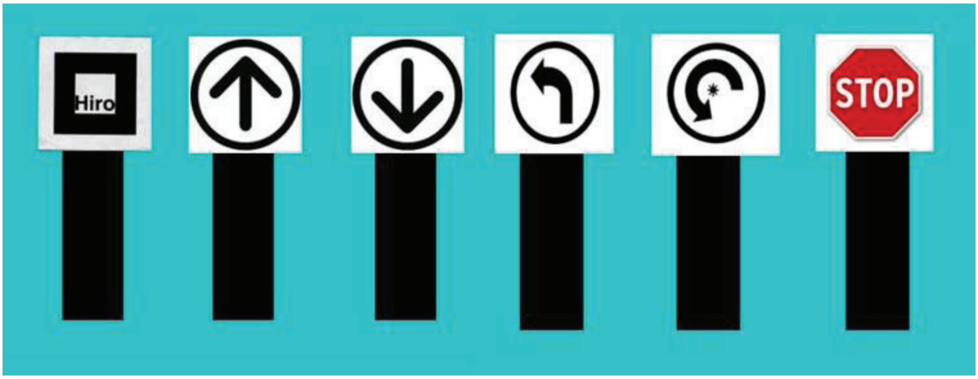


Fig. 6. The image on the far left is of the paddle as seen in the real world. The remainder of images show the AR augmented view showing, from left to right, forward, reverse, forward turn left, turn left in place and stop.

The gaze-processing module defines the gaze direction of the user through the computer vision techniques made available from the use of the ARToolKit libraries. The line of sight of the user into the virtual world is computed by calculating the position of the camera mounted on the HMD, this calculation represents the users gaze direction. By comparing this position to the position and orientation of the marker set that represents the robots virtual world the user gaze direction can be determined.

The gaze direction of the user in the AR environment is used to define spatial references such as "behind" and "to the right of" objects selected using the real world paddle. By knowing where the user is in reference to the objects in the virtual scene these spatial references can be defined in the reference frame of the user, as described in (Irawati, Green et al. 2006). This information is then converted into the reference frame of the robot. The conversion is made possible through the use of AR, which provides a common reference frame for both the robot and human collaborators. The desired location is then sent to the robot where it uses its autonomous capabilities to move to the position in the real world.

As a case study a Lego MINDSTORMS™ NXT (The Lego Group 2007) mobile robot in the Tribot configuration was used as a mobile robot to collaborate with. The NXT robot can be seen in Figure 7. To incorporate the mobile robot into the system the NXT++ libraries (NXT++ 2007) were used. These libraries represent an interface to the MINDSTORMS™ robot written in C++ that enables a PC to communicate with the robot through a Bluetooth connection. A Lego MINDSTORMS™ robot was chosen because it is a simple low cost platform to prove out the functionality of the AR-HRC system.

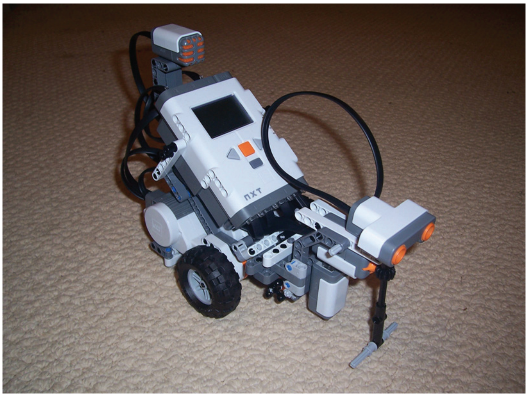


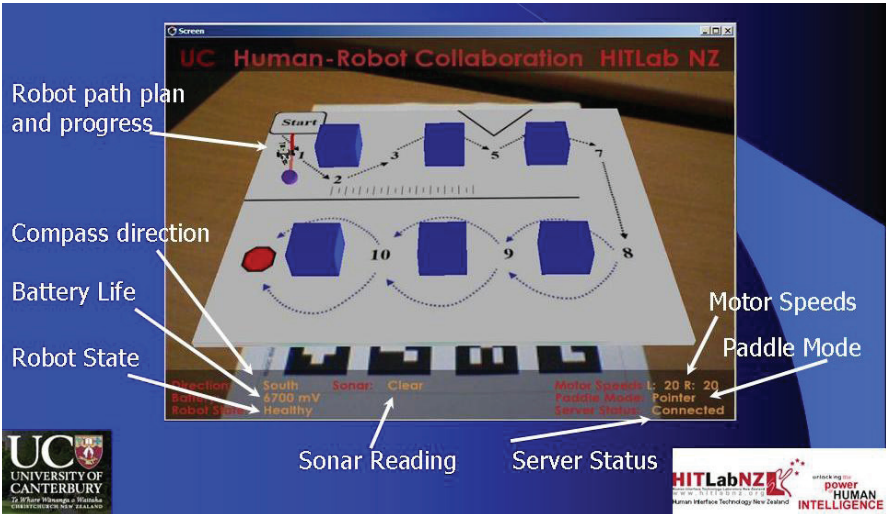
Fig 7. Lego MINDSTORMS™ NXT robot (The Lego Group 2007).

An obstacle course was created for the NXT robot to manoeuvre through. A virtual representation of this world and the NXT robot was created to be used in the AR environment. The motions of the real robot were calibrated to match those of the virtual representation of the NXT robot in AR.

The NXT robot used had one ultrasonic sensor on the front to sense objects and measure the distance to them. The robot also had a touch sensor on the front that would stop the robot if triggered to avoid colliding into objects in its world. The limited sensing ability of the robot allowed us to take advantage of dialog to ensure the robot took a safe path.

An example of using dialog to ensure safe robot motion would be when the robot had to back up. With no rear sensors the robot was unable to determine if a collision was imminent. In this case, the robot asked the human if it was ok to move in reverse without hitting objects in its environment prior to commencing movement. Once the robot received confirmation that the path in the reverse direction was clear, it began to move in the reverse direction. Since the robot had to ask for guidance to complete the reverse manoeuvre, the user was aware that the robot might need assistance. It was then assured that the user has maintained spatial awareness, which in turn enabled a collaborative human-robot exchange and the resulting safe execution of robot motion.

A heads up display was used to keep the human informed of the internal state of the robot. The human could easily see the direction the robot was moving, the battery level, motor speeds, paddle mode and server status. Figure 8 is an example of the view provided to the human through the HMD. The internal state of the robot is easily identifiable as is the robots intended path and progress.


Screen
UC Human-Robot Collaboration HITLab NZ
Robot path plan and progress
Start
1
2
3
5
7
8
9
10
Compass direction
Battery Life
Robot State
Motor Speeds
Paddle Mode
Sonar Reading
Server Status
UC UNIVERSITY OF CANTERBURY Te Whare Wānanga o Waitaha CHRISTCHURCH NEW ZEALAND
HITLabNZ www.hitlabnz.org Human Interface Technology New Zealand
unlocking the power HUMAN INTELLIGENCE

Direction: South
Battery: 6700 mV
Robot State: Healthy
Sonar: Clear
Motor Speeds L: 20 R: 20
Paddle Mode: Pointer
Server Status: Connected

Fig. 8. View provided to user in HMD highlighting robot state information and path.

Since the robot's environment was modelled in 3D and used as the virtual scene in AR, the human is provided the means to gain a feeling of presence in the robot's world. The system allows the human to naturally communicate with the robot in the modality most comfortable to the user. Given the limitations of the MINDSTORMS™ robot sensors the human had to do more monitoring than would be necessary with a more autonomous robot. Another benefit of using AR as a means to mediate the communication between the robot and human is the ability to smoothly transition from an exo-centric (god's eye view) to an ego-centric view. This means the user can smoothly transition from a bird's eye view of the robot in its environment to the view provided by the robot's camera, and vice-versa. The user is able to issue a verbal command to switch from one viewpoint to the other. This ability to view the robot's world from two vantage points increases the situational awareness of the human by allowing the scene to be viewed from the various vantage points.

The human and robot are able to create a path plan and review this plan before the robot is set in motion. The ability to review the plan prior to execution provides the means for detection of unexpected situations and the identification of probable collisions. The path plan can be interactively created and modified through dialog and gesture interaction within the AR environment.

A path can be planned and then viewed as overlays on the 3D virtual model of the robot's environment. The user can choose to show the planned path or hide the trajectory if the overlay interferes with viewing other important parts of the environment. Waypoints can be added to or deleted from the path plan to ensure smooth motion and obstacle avoidance. The result is that the motion of the robot, once it executes its collaboratively designed plan, is smoother and collision free.

During review of a path plan the robot gives the user verbal feedback if a collision is imminent, as well as when a collision happens. This feedback allows the user to modify the plan to ensure collision free movement. This setup also enables the user to provide a margin of safety between the robot and any objects it might possibly collide with. This type of interaction between the robot and human in creating the path plan, identifying possible problems with the plan and eliminating these issues highlights the collaborative nature of the human-robot interface.

6. Performance Experiment

This section discusses an experiment that compared three user interface techniques for interaction with a simulated mobile robot that was located remotely from the user. A typical means of operating a robot in such a situation is to teleoperate the robot using visual cues from a camera that displays the robot's view of its work environment. The operator has a difficult time maintaining awareness of the robot in its surroundings due to this single ego-centric view. Therefore, this interface was compared with two versions of the AR-HRC system.

6.1 Experimental Design

The task for the user study was to work with a simulated robot to get it through a predefined maze. Three conditions were used:

- A typical teleoperation mode with a single ego-centric view from the robot's onboard camera. This condition was called the *Immersive Test*.
- A limited version of the AR-HRC system that allowed the user to see the robot in its work environment in AR and interact with the robot using speech and gesture, but without pre-planning and review of the robot's intended actions. This condition was called *Speech and Gesture no Planning (SGnoP).*
- The full AR-HRC system that allowed the human to view the robot in the AR environment, use spoken dialog and gestures to work with the robot to create a plan and review this plan prior to execution. This was the *Speech and Gesture with Planning, Review and Modification (SGwPRM)* condition.

The three conditions are, therefore, distinguished by increasing levels of collaboration or communication channels.

6.2 Participants

Ten participants were run through the experiment, seven male and three female. Ages ranged from 28 to 80 and all participants were working professionals. Six participants had Bachelor's degrees and four advanced degrees. Seven of the participants were engineers while the other three had non-scientific backgrounds. Overall the users rated themselves as not familiar with robotic systems, speech systems or AR.

6.3 Procedure

The first step of the experiment was to have each participant fill out a demographic questionnaire to evaluate their familiarity with AR, game playing experience, age, gender and educational experience. Since speech recognition was an integral part of the experiment it was necessary to have each participant run through a speech training exercise. This training created a profile for each user so that the system was better able to adapt to the speech patterns of the individual participant.

The objective of each trial was then explained to the participants. They were told that they would be interacting with a mobile robot to get it through the predefined maze shown in Figure 9. The maze contained a defined path for the robot to follow and various obstacles the robot would need to maneuver around. The black lines indicate a path that needed to be followed while the blue lines indicate that the user had the choice of how to proceed. The participants were told that the robot must arrive at each of the numbers on the map as this was going to be a measure of accuracy for the test. Other parameters measured were impending collisions, actual collisions and time to completion.

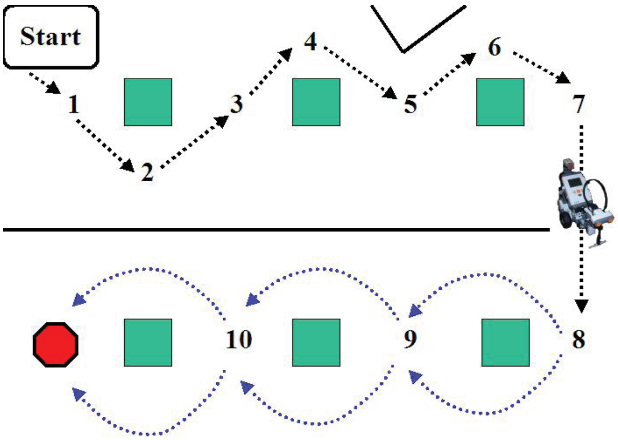


Fig. 9. User study maze, black lines indicate defined path, blue lines indicate users choice.

It was explained to the participants that the robot was located remotely. The effect on the trials this had was that when the robot was directly driven a time delay would be experienced. So a delay in reaction of the simulated robot was not the system failing, but was the result of the time taken for the commands to reach the robot and the update from the robot to arrive back to the user.

The experimental setup used was a typical video see through AR configuration. A webcam attached to an eMagin Z800 Head Mounted Display (HMD) (eMagin 2008) were both connected to a laptop PC running ARToolKit based software. Vision techniques were use to identify unique markers in the user’s view and align the 3D virtual images of the robot in its world to these markers. This augmented view was presented to the user in the HMD. Figure 10 shows a participant using the AR-HRC system during the experiment.

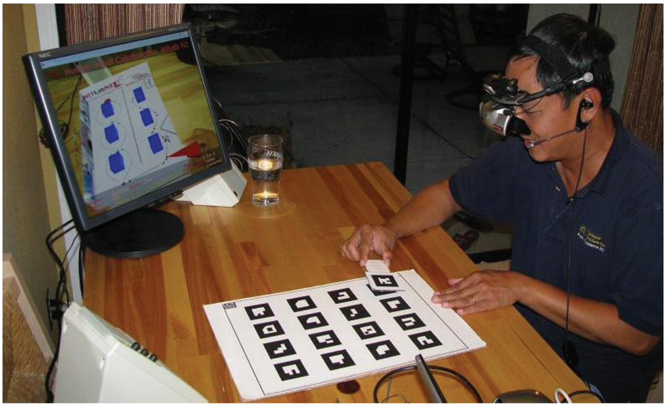


Fig. 10. A participant using the AR-HRC system. The image on the monitor is what is being displayed to the user in the HMD.

The same sequence of events took place for each trial. Before the trial was run, the participant practiced using the system to become familiar with the interface for that particular condition. The user also practiced the speech specific to that trial. Once the user felt comfortable with the interface the trial was run.

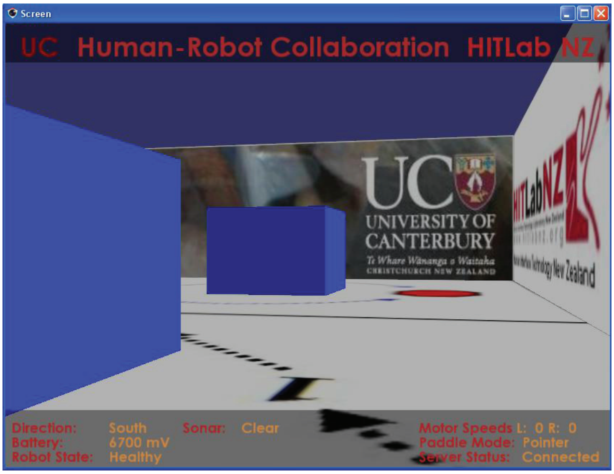
When a trial was complete the user was given a subjective questionnaire to determine if they felt that they had a high level of spatial awareness during the trial. The user was also questioned about whether they felt present in the robot's world and their view of the robot as a partner. The participants were also asked to list what they liked and disliked about the interface. This questionnaire was exactly the same for all three trials.

At the end of the experiment, after the participant had completed all three trials, a subjective questionnaire was given so the user could compare the three conditions. The post trial questionnaires discussed previously referred only to the trial that had just been completed. The subjective questioning was conducted in this manner to let the user express their feeling of each condition individually and then compare the three conditions upon completion of the full experiment.

The order of the trials was randomly selected for each user to eliminate the effects of sequencing in the results. The Immersive condition was intended to mimic the traditional teleoperational control of a mobile robot. The SGnoP condition was to introduce a part of the AR-HRC system, namely the ability to reach into the world of the robot through the AR graphics and use spatial dialog to issue commands. The SGwPRM condition also included the AR interaction but added deeper dialog with the robot and the ability to create and review a plan with the robot prior to its execution. The intent was to see if the Speech and Gesture with Planning, Review and Modification condition provided the user a feeling of presence in the robot's world and a feeling of the robot being a collaborative partner rather than a tool.

6.4 Immersive Condition

The Immersive Test simulated the direct teleoperation of the robot with visual feedback to the user displaying the view that the robot saw through its camera. This view provided the user an ego-centric view of the robot's environment. User interaction included keyed input for robot translation and rotation. The view the user experienced can be seen in Figure 11.


Screen
UC Human-Robot Collaboration HITLab NZ
UC UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND
HITLabNZ
New Zealand

Direction: South Sonar: Clear
Battery: 6700 mV
Robot State: Healthy
Motor Speeds L: 0 R: 0
Paddle Mode: Pointer
Server Status: Connected

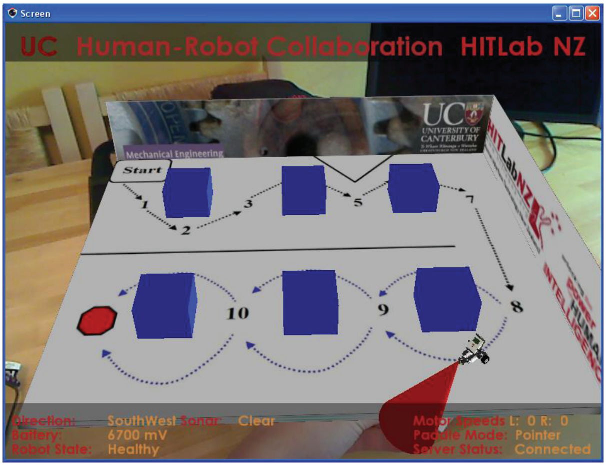
Fig. 11. The user's view for the Immersive condition. The view shown is that from the robot.

6.5 Speech and Gesture no Planning

The SGnoP condition provided the user with a 3D graphic of the robot and maze. The participant was able to use spatial dialog coupled with gestures using a paddle to interact with the graphical world of the robot in the AR environment. Thus, the participant was able to point to a 3D location on the maze and instruct the robot to "go there" or select an object and instruct the robot to "go to the right of that". The robot responded immediately to the verbal commands given, minus the built in time delay for the simulation of a remotely located robot. The speech was one-way in that the system in this condition understood the user's spatial dialog but did not respond verbally. The view provided to the participant can be seen in Fig 12.

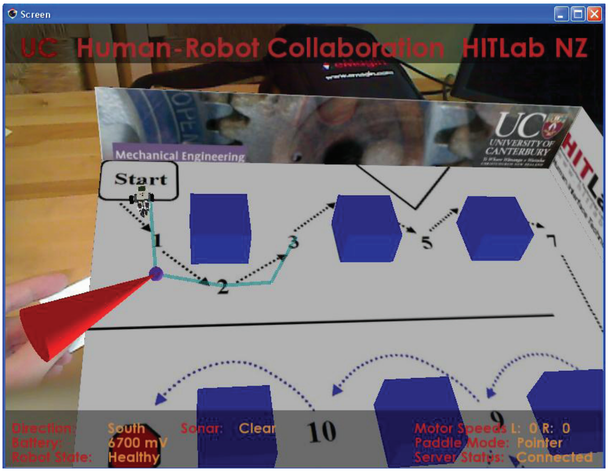
6.6 Speech and Gesture with Planning Review and Modification

The user's view for the SGwPRM condition is shown in Figure 13. This condition included all the features of the SGnoP condition but also allowed the participant to use spatial dialog to create a plan with the robot. The user was able to select a goal location then assign waypoints for the robot to follow to arrive at the goal destination. The user was able to interactively modify the plan by adding or deleting way points. The plan was displayed to the user in the AR environment thus allowing the participant to determine if the intentions of the robot matched those of the user before any commands were executed by the robot. The robot participated in the dialog by responding to the user verbally for each interaction and alerting the user verbally when the robot came close enough to an object that the robot "thought" it would collide.


Screen
UC Human-Robot Collaboration HITLab NZ
UC UNIVERSITY OF CANTERBURY
Mechanical Engineering
HITLabNZ
Start
1
2
3
5
7
8
9
10

Direction: SouthWest Sonar: Clear
Battery: 6700 mV
Robot State: Healthy
Motor Speeds L: 0 R: 0
Paddle Mode: Pointer
Server Status: Connected

Fig. 12. The user's view for the Speech and Gesture no Planning condition. This condition uses a Tangible User Interface metaphor with the user selecting a point that the robot moves to.


Screen
UC Human-Robot Collaboration HITLab NZ
UC UNIVERSITY OF CANTERBURY
Mechanical Engineering
HITL
Start
1
2
3
5
7
9
10

Direction: South Sonar: Clear
Battery: 6700 mV
Robot State: Healthy
Motor Speeds L: 0 R: 0
Paddle Mode: Pointer
Server Status: Connected

Fig. 13. The user's view for the Speech and Dialog with Planning, Review and Modification condition. The user creating a plan (blue line) that includes various waypoints through the use of spatial dialog and gesture.

6.7 Results

The ten participants each performed three tasks, one for each condition. Each trial yielded a measure of time to completion, impending collisions, collisions and accuracy in reaching each of the ten defined locations on the map. An impending collision was defined as any time the robot came within a predefined threshold of an object. A warning was given to the user that an object was close enough to the robot that a human perspective was needed to determine if the current course of action was clear. The following section reports the results of these measures as well as the results of the subjective questioning.

6.7.1 Objective Measures

There was a significant main effect of condition on task completion times with an ANOVA test finding ($F_{2,27} = 9.83$, $p< 0.05$). Pairwise comparison with Bonferroni correction ($p < 0.05$) revealed significant differences between the SGwPRM and the other two conditions while there was no significant difference between SGnoP and the Immersive conditions. Users in the Immersive condition performed faster than the other two conditions with a mean completion time of 331.60 seconds (se = 36.72). The results for mean time to completion are shown in Figure 14.

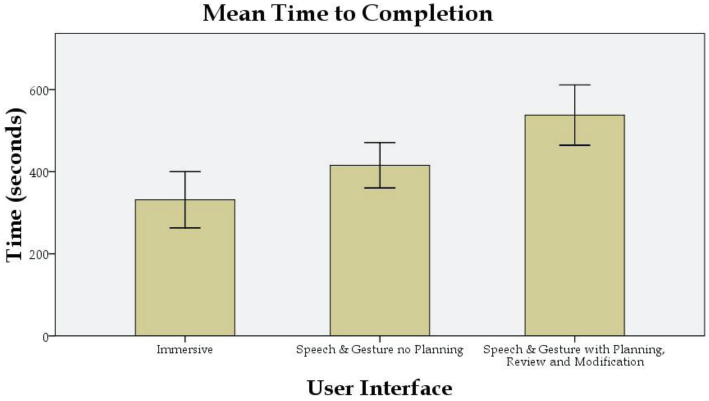


Fig. 14. Mean time to completion.

The condition also had a significant main effect on accuracy with an ANOVA test finding ($F_{2,27} = 8.44$, $p< 0.05$). Accuracy was a count of the number of predefined locations reached during the traversal of the maze. Pairwise comparison with Bonferroni correction ($p < 0.05$) revealed significant differences between the SGwPRM and Immersive conditions but no significant differences between the SGnoP and the other two conditions. Users in the SGwPRM condition performed the best by arriving at an average of 9.50 out of 10 defined locations (se = 0.22). The results of accuracy measures are shown in Figure 15.

There was a significant main effect of conditions on the number of close calls with an ANOVA result of ($F_{2,27} = 13.10$, $p< 0.05$), but no significant effect on the number of collisions. Pairwise comparison using Bonferroni correction ($p < 0.05$) showed significant differences for close calls between the Immersive condition and the other two conditions. There was no significant difference between SGnoP and SGwPRM. The SGwPRM condition performed best with a mean number of close calls of 3.60 (se = 1.01). The results of close calls are shown in Figure 16.

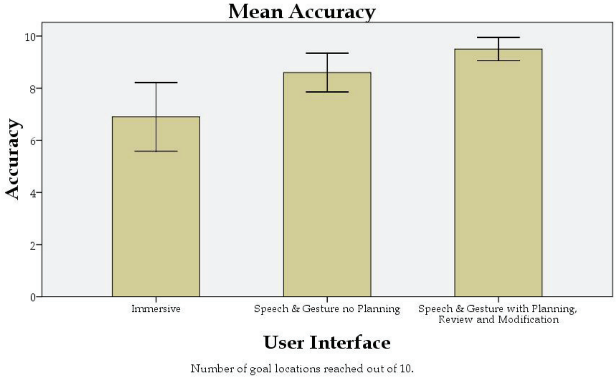


Fig. 15. Mean accuracy.

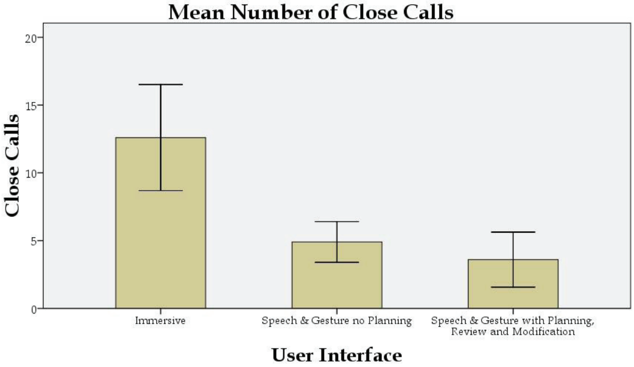


Fig. 16. Mean number of close calls.

6.7.2 Subjective Measures

The answer for each post trial question was given on a Likert scale of 1- 7 (1 = disagree completely, 7 = agree completely) and analyzed using an ANOVA test. Where necessary, post-hoc analysis was performed using Bonferroni correction ($p < 0.05$). The results of the questionnaires for the individual trials (PT) are presented first and can be seen in Fig 17.

- PTQ1: *I knew exactly where the robot was at all times*. There was a significant difference between conditions ($F_{2,27} = 7.43$, $p < 0.05$). Pairwise comparison showed a significant effect between the Immersive condition and the other two conditions, but no significant effect between the SGnoP and SGwPRM conditions. Users felt that they maintained situational awareness best in the SGwPRM condition.
- PTQ2: *The interface was intuitive to use.* There was no significant difference between the conditions.
- PTQ3: *The robot was a member of my team as we completed the task.* There a significant difference between conditions ($F_{2,27} = 6.07$, $p < 0.05$). Pairwise comparison revealed

a significant effect between the Immersive condition and the two others. There was no significant difference between the SGnoP and SGwPRM conditions. The users felt that the robot was a member of their team in the SGwPRM condition.

- PTQ4: *I felt a sense of being present in the robot's world.* There was no significant difference between the conditions.
- PTQ5: *I was always aware of how close the robot was to objects in its environment.* There was no significant different between the three conditions.
- PTQ6: *I felt like the robot was just a tool and not a collaborative partner.* There was a significant difference between conditions ($F_{2,27} = 5.68$, $p < 0.05$). Pairwise comparison revealed a significant effect between the SGwPRM and Immersive conditions. There was no significant effect between the SGnoP and the other two conditions. Users felt that the robot was more of a collaborative partner in the SGwPRM condition.

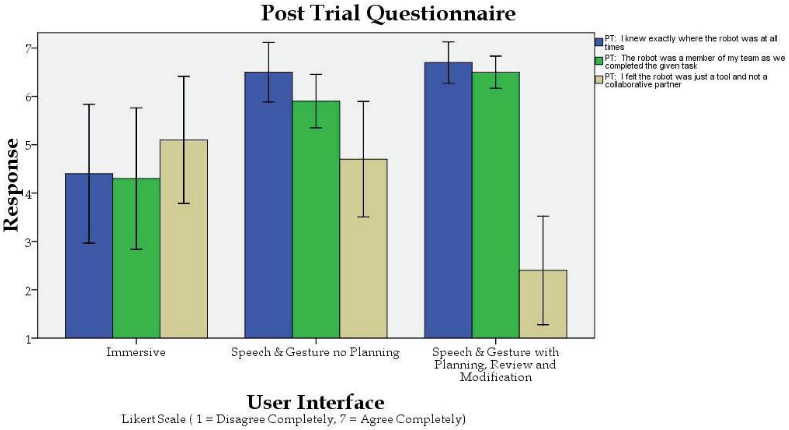


Fig. 17. Post trial questionnaire responses.

The results of the post experiment (PE) questionnaire are now presented. As opposed to the questions above which were completed for each condition individually, the users ranked the three conditions in order of preference for the following questions. The results of the post experiment questionnaire can be seen in Figure 18.

- PEQ1: *I was aware of collisions as they happened.* There was a significant difference between conditions ($F_{2,27} = 12.47$, $p < 0.05$). Pairwise comparison revealed a significant effect between the SGwPRM and the other two conditions, but no significant effect between the SGnoP and the Immersive conditions. Users felt that they were most aware of collisions while using the SGwPRM condition.
- PEQ2: *I had a feeling of working in a collaborative environment.* There was a significant difference between conditions ($F_{2,27} = 17.90$, $p < 0.05$). Pairwise comparison revealed a significant main effect between SGwPRM and the other two conditions, but no significant effect between the Immersive and SGnoP conditions. The SGwPRM condition was selected as providing the users with the greatest feeling of working in a collaborative environment.

- PEQ3: *I felt the robot was a partner.* There was a significant difference between conditions ($F_{2,27}$ = 17.90, p < 0.05). Pairwise comparison revealed a significant main effect between SGwPRM and the other two conditions, but no significant effect between the Immersive and SGnoP conditions. The SGwPRM condition provided the users with a feeling that the robot was a partner.
- PEQ4: *The interface was intuitive to use.* There was no significant difference due to condition.
- PEQ5: *I was aware of the robot's surroundings.* There was a significant difference between conditions ($F_{2,27}$ = 8.39, p < 0.05). Pairwise comparison showed a significant effect between the SGwPRM and Immersive conditions, but no significant effect between the SGnoP and the other two conditions. Users felt that the SGwPRM condition enabled them to be the most aware of the robot's surroundings.
- PEQ6: *I had to always pay attention to the robot's actions.* There was a significant difference between conditions ($F_{2,27}$ = 8.77, p < 0.05). Pairwise comparison showed a significant effect between the Immersive condition and the two others, but no significant effect between the SGnoP and SGwPRM conditions. User felt that they needed to pay attention to the robot's action more in the Immersive condition.
- PEQ7: *I felt the robot was a tool.* There was no significant difference between the three conditions.
- PEQ8: *I felt I was present in the robot's environment.* No significant difference was found between the three conditions.
- PEQ9: *I knew when the robot was about to collide with an object.* There was a significant difference between conditions ($F_{2,27}$ = 9.62, p < 0.05). Pairwise comparison revealed a significant effect between the SGwPRM and the other two conditions, but no significant effect between the Immersive and SGnoP conditions. Participants felt that the SGwPRM condition was best for maintaining awareness of potential collisions.

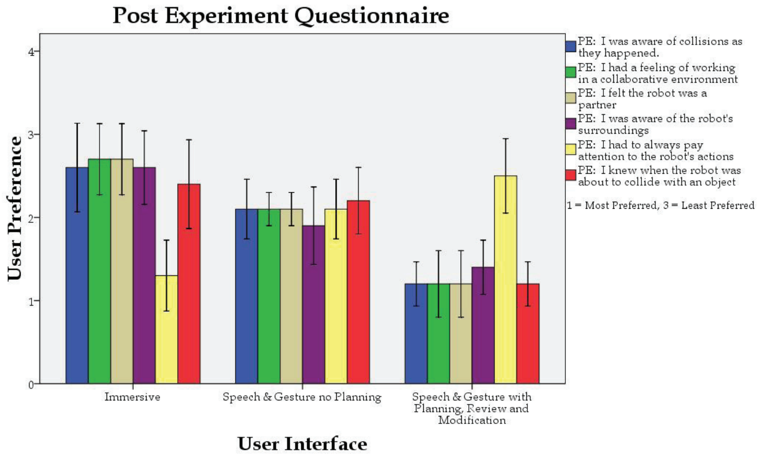


Fig. 18. Post experiment questionnaire responses.

6.8 Discussion

The Immersive condition was significantly faster than both the SGnoP and SGwPRM conditions. This result could be in part due to the lower learning curve of the Immersive condition. This hypothesis is supported by comments users provided in the post experiment questionnaire. Five users commented that the Immersive condition was simple and straight forward to use or that there was no learning curve. The SGnoP and SGwPRM conditions, on the other hand, were a bit more difficult for the participants to become acquainted with. This higher learning curve is due to two things. One, the user had to become familiar with the dialog that the system understood in a relatively short period of time. And two, at the same time the users also had to become familiar with selecting locations and objects in the AR environment.

In the Immersive condition the participants did complete the task faster. However, the measure of accuracy showed that the users performed worst in the Immersive condition. The participants performed best in terms of accuracy in the SGwPRM condition. So although this condition took on average the longest time to complete the task, it resulted in the most accurate performance. It's not surprising to see that the SGwPRM has a longer completion time. This result is inherent in the design of the interface as it takes time for the robot to display its plan in AR, for the user to agree with or modify the plan, and then have the robot execute the plan.

Although there was no significant effect of condition on the number of collisions, there was a significant effect on the number of close calls. The condition that performed the worst in this measure was the Immersive condition, while the SGwPRM condition performed the best. This result combined with the results from questions PTQ1, PEQ1, PEQ5 and PEQ9 indicate that the SGwPRM condition provided the users with the highest level of situational awareness.

An analysis of the dialog used revealed that deictic phrases, such as "go here", were used 87% of the time for the SGnoP condition and 93% of the time for SGwPRM. The remaining times deeper spatial dialog was used, such as "to the left of this" whilst selecting an object in the AR environment. This result of mainly using the deictic gestures could be due to the learning curve mentioned previously. To use the deeper spatial dialog the participants had to remember longer phrases and coordinate issuing these phrases with the selection of objects in AR. Although this coordination is not difficult to master with practice, the participants tended to use a method that they could immediately master. The use of the deeper spatial dialog tended to happen later in the experiment, once the participants had become familiar with interacting with the system.

Another subjective measure was the feeling of working in a collaborative environment. The responses from questions PTQ6, PEQ2 and PEQ6 show that the users felt that they were working in a collaborative environment when completing the task using the SGwPRM condition. Question PEQ3 responses show that participants felt the robot was a partner when working with in the SGwPRM condition. These results show that participants felt they were working in a collaborative team environment in the SGwPRM condition.

The last subjective question posed to the users was to select the most effective condition. Nine of the ten participants selected the SGwPRM as the most effective. The remaining user selected the SGnoP condition. Reasons provided for the selection of SGwPRM included effective path creation, verbal feedback from the robot and the ability to change the plan mid-stream. Conversely, reasons given for not choosing the other conditions included the lack of planning caused crashes, the Immersive condition lacked situational awareness and limited feedback from the robot. These results show that being able to exchange dialog with the robot and seeing the robots intentions does indeed create a collaborative environment.

7. Future Work

The AR-HRC system presented in this chapter can be viewed as a first step into an emerging research area in HRI. With that in mind, there exists opportunities to expand on this research. These opportunities are presented first by modules of the AR-HRC system, then the system as a whole and finally some potential areas for integration and evaluation studies.

Speech recognition and text-to-speech obviously play a major role in the AR-HRC system and are themselves an active field of research. As this field matures further, false detection rates will be reduced and, consequently, recognition rates will increase. As false detection rates are reduced it will be possible to create dialog that more closely replicates how humans speak. Currently the design of the dialog must be mindful of false detection, so it is necessary to define more complex phrases for a situation than may be necessary. For example, instead of having a command of just “stop”, the AR-HRC system uses “robot stop”. The word “robot” was added to the goal phrase for recognition to prevent the system from falsely recognizing the single syllable word “stop” from either utterances of the user or background noise.

The AR-HRC system uses the freely available Microsoft Speech for text-to-speech feedback. The options for voice selection are limited and sound very robotic. The implementation of a commercial speech recognition system might offer more options for less robotic sounding voices. The intent of the research presented in this chapter was not to explore speech recognition or text-to-speech, but to incorporate this technology into the AR-HRC. Therefore an avenue for future research would be an improved speech recognition and text-to-speech package.

Augmented Reality is another active field of research. There are numerous avenues being pursued to enhance AR technology, a small number are listed here:

- Outdoor tracking
- Mobile AR applications
- Natural feature tracking / marker-less tracking
- Reduction of noise in tracker output
- World model creation

Future work up to this point has addressed technology that has been incorporated into the AR-HRC system. Obviously as these technologies mature any system that implemented them will improve as a result. However, the AR-HRC system could be enhanced through further research. A proof of concept application with a mobile robot was described in this chapter, numerous other robotic applications could benefit from the HRI techniques afforded by the AR-HRC system. Lunar or Martian rovers are possible applications for the AR-HRC. Unmanned Aerial Vehicles (UAVs), Unmanned Underwater Vehicles (UUVs) and terrestrial rovers, to name just few, could also benefit from the HRI techniques presented in this chapter. And with each new application the dialog will need to be catered to that specific domain and a variety of evaluation studies will need to be conducted to determine how best to implement the system to the given application.

Gesture interaction is yet another area of active research. A variety of gesture interaction methods could be explored for use in the AR-HRC system. Data gloves, visual hand tracking, and even the use of the Nintendo Wii™ remotes (Nintendo 2008) could be explored as gesture input devices. Computer vision based natural hand input is a particularly promising area of current research that could be extended for HRI.

Improvements or variations to the display device could be explored as well. The implementation presented in this chapter used a head mounted display (HMD). Other possibilities include large LCD screens, white boards, or even the use of a Cave Automatic Virtual Environment (CAVE) and fully immersive graphics environments.

The AR-HRC system could also be expanded to accommodate multiple humans and multiple robots. Possible scenarios could include co-located humans or humans located remotely from each other. These groups could be interacting with a single robot or several robots that do not necessarily have to be located in the same work space.

8. Conclusions

This chapter leads the reader through the development of the AR-HRC system from concept and background through the design of the necessary set of interfaces required to enhance human-robot interaction. It thus began by introducing the need for human-robot collaborative teams in terms of current and emerging application spaces requiring collaboration to achieve or significantly improve outcomes. In particular, the area of space exploration will require human-robot interaction at levels well beyond current state of the art or understanding. Similar terrestrial applications are outlined that will be significantly enhanced, as well. However, it was also shown that little attention has been paid to research in this field. All of these issues provided the impetus for the creation of the Augmented Reality Human-Robot Collaboration (AR-HRC) system described here.

A discussion of the related work in HRI has shown that an effective system should transfer the interaction mechanisms natural for humans to the precision required for machine information. Previous work in HRI has also shown that the autonomy level of an HRI system should be variable so that it can match the needs of a given situation. In this manner the system is able to capitalize on the problem solving skills of a human while also effectively balancing that with the speed and dexterity of a robot.

Prior work in HRI also highlighted the importance of situational awareness. The lack of situational awareness has been shown to decrease performance and, in certain cases, can lead to catastrophic failures. Use of natural speech has also been shown to be effective in HRI. However, speech alone is not enough to complete the grounding process in the exchange between human and robot, leading to a reduced ability to communicate as a result. Therefore, a multi-modal interface is shown to provide a more effective approach. By combining speech with gesture, a more natural interface and the requisite grounding is achieved. The multi-modal medium used for the AR-HRC system presented here is Augmented Reality, which affords both speech and gestural communication channels.

The literature review therefore includes an introduction to AR and the state of the art of AR in the context of using it in a multi-modal human-robot interaction system. AR has been shown to provide a shared work space that is conducive to collaboration and at the same time increases situational awareness, enhancing its potential in this situation. AR also supports a tangible user interface, essentially allowing a person to use a real world object to affect change on the 3D graphics of the AR environment, providing an enhanced graphical or visual communication channel. AR was also shown to increase performance in robotic control directly. In particular, the use of AR improved situational awareness by providing the human with an exo-centric view of the robots workspace. Therefore, AR provides rich spatial cues in the shared environment and enables the use of natural spatial dialog. By taking explicit advantage of the benefits that AR offers, a robust human-robot collaboration system can be created.

As a first step towards the development of the AR-HRC system, a multi-modal interface for AR was created. This interface fused spatial dialog and gesture interaction to affect change in an AR environment. The results of a user study for this system showed that the multi-modal interface improved performance in the AR environment. These positive results drove the design of the AR-HRC system to include multi-modal AR interaction through the use of spatial dialog and gestures.

The architectural design of the AR-HRC system was then presented. The various components of the system were described in detail. The intercommunication of these modules was also discussed. The system design is seen to fuse speech and gesture inputs with the AR overlays of the robots plans and internal state. As a result, the system is able to provide a communication environment this is equally and highly effective for both parties in the human-robot collaboration.

The chapter then discussed the integration of a mobile robot into the AR-HRC system. The environment the robot was to work in was described, as well as a task for the robot to complete. The ability to create, review and modify robot plans was described highlighting the collaborative nature of the AR-HRC system.

A performance experiment comparing three user interfaces was then discussed. The three interfaces used were:

- A typical teloperation interface
- A version of the AR-HRC that did not include planning or review
- The full version of the AR-HRC that did include path planning, review and modification

Each of these interfaces was described in detail. The task to be completed, the variables measured and the subjective questionnaires participants filled out were also discussed. Results showed that participants felt the robot was more of a tool in the teleoperation interface. Participants thought of the robot as more of a collaborative partner when using the full version of the AR-HRC interface.

While these results might be as expected, they clearly highlight the change in perception of the human partner in the robots capability that arises with increasingly effective two-way communication through an environment explicitly designed to maximize that collaborative discussion. Hence, it is clear that human-robot interaction, while a nascent field, can offer significantly improved task performance for both robot and operator, even in the simple proof of concept studies presented here. Thus, the main conclusion of this chapter is that human-robot collaboration represents an immediate and significant frontier to be crossed on the way to developing next generation robotic applications and that AR technology can be of significant benefit in this work.

In summary, this chapter has shown that the AR-HRC system does enable natural and effective communication to take place. The use of AR affords the integration of a multi-modal interface combining speech and gesture interaction, as well as providing the means for enhanced situational awareness. The AR-HRC system gives the user the feeling of working in a collaborative human-robot team rather than the feeling of the robot being a tool, as a typical teleoperation interface provides. Therefore, the development of the AR-HRC system brings closer the day when humans and robots can truly interact in a collaborative manner.

9. References

ARToolKit (2008). http://www.hitl.washington.edu/artoolkit/: accessed January 2008.

Azuma, R. T. (1997). "A Survey of Augmented Reality." Presence: Teleoperators and Virtual Environments **6**(4): 355-385.

Bechar, A. and Y. Edan (2003). "Human-robot collaboration for improved target recognition of agricultural robots." Industrial Robot **30**(5): 432-436.

Billinghurst, M., R. Grasset, et al. (2005). "Designing Augmented Reality Interfaces." Computer Graphics SIGGRAPH Quarterly **39(1)**: 17-22 Feb.

Billinghurst, M., H. Kato, et al. (2001). "The MagicBook: A transitional AR interface." Computers and Graphics (Pergamon) **25**(5): 745-753.

Billinghurst, M., I. Poupyrev, et al. (2000). Mixing realities in Shared Space: An augmented reality interface for collaborative computing. 2000 IEEE International Conference on Multimedia and Expo (ICME 2000), Jul 30-Aug 2, New York, NY.

Bolt, R. A. (1980). "Put-That-There: Voice and Gesture at the Graphics Interface." In Proceedings of the International Conference on Computer Graphics and Interactive Techniques **14**: 262-270.

Bowen, C., J. Maida, et al. (2004). "Utilization of the Space Vision System as an Augmented Reality System for Mission Operations." Proceedings of AIAA Habitation Conference: Houston TX.

Clark, H. H. and S. E. Brennan (1991). Grounding in Communication. Perspectives on Socially Shared Cognition. L. Resnick, Levine J., Teasley, S. Washington D.C., American Psychological Association**:** 127 - 149.

Denecke, M. (2002). "Rapid Prototyping for Spoken Dialog Systems." In Proceedings of 19th International Conference on Computational Linguistics.

Drury, J., J. Richer, et al. (2006). "Comparing Situation Awareness for Two Unmanned Aerial Vehicle Human Interface Approaches." Proceedings IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR). Gainsburg, MD, USA August.

Ellis, S. (2000). "Collision in Space." Ergonomics in Design.

eMagin (2008). www.3dvisor.com , last accessed June 2008.

Fitzmaurice, G. W. and W. Buxton (1997). An Empircal Evaluation of Graspable User Interfaces: Towards Specialized, Space-Multiplexed Input. Conference on Human Factors in Computing Systems (CHI 97), Atlanta, GA, USA.

Fong, T., C. Kunz, et al. (2006). "The Human-Robot Interaction Operating System." Proceedings of 2006 ACM Conference on Human-Robot Interaction, March 2-4: 41-48.

Fong, T. and I. R. Nourbakhsh (2005). "Interaction challenges in human-robot space exploration." Interactions **12**(2): 42-45.

Fong, T., C. Thorpe, et al. (2002). Robot, asker of questions. IROS 2002, Sep 30, Lausanne, Switzerland, Elsevier Science B.V.

Fussell, S. R., L. D. Setlock, et al. (2003). Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. The CHI 2003 New Horizons Conference Proceedings: Conference on Human Factors in Computing Systems, Apr 5-10, Ft. Lauderdale, FL, United States, Association for Computing Machinery.

Green, S. A., M. Billinghurst, et al. (2008). "Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design." International Journal of Advanced Robotic Systems **5**(1): 1- 18, March 2008.

Green, S. A., S. M. Richardson, et al. (2008). "Multimodal Metric Study for Human-Robot Collaboration." 1st International Conference on Advances in Computer-Human Interaction (ACHI-08), February 10 - 15: Sainte Luce, Martinique.

Huttenrauch, H., A. Green, et al. (2004). "Involving users in the design of a mobile office robot." IEEE Transactions on Systems, Man and Cybernetics, Part C **34**(2): 113-124.

Irawati, S., S. Green, et al. (2006). An Evaluation of an Augmented Reality Multimodal Interface Using Speech and Paddle Gestures. In Proceedings of the 16th International Conference on Artificial Reality and Telexistence (ICAT 2006), Hangzhou, China.

Irawati, S., S. Green, et al. (2006). Move the Couch Where? Developing an Augmented Reality Multimodal Interface. In Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006), Santa Barbara, California.

Ishii, H. and B. Ullmer (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atom. Conference on Human Factors in Computing Systems (CHI 97), Atlanta, GA, USA.

Ishikawa, N. and K. Suzuki (1997). Development of a human and robot collaborative system for inspecting patrol of nuclear power plants. Proceedings of the 1997 6th IEEE International Workshop on Robot and Human Communication, RO-MAN'97, Sep 29-Oct 1, Sendai, Japan, IEEE, Piscataway, NJ, USA.

Kanda, T., H. Ishiguro, et al. (2002). Development and evaluation of an interactive humanoid robot "Robovie". 2002 IEEE International Conference on Robotics and Automation, May 11-15, Washington, DC, United States, Institute of Electrical and Electronics Engineers Inc.

Kato, H., M. Billinghurst, et al. (2000). Virtual Object Manipulation on a Table-top AR Environment. IEEE and ACM International Symposium on Augmented Reality (ISAR 2000).

Kato, H., M. Billinghurst, et al. (2001). Tangible Augmented Reality for Human Computer Interaction. NICOGRAPH 01, Nagoya, Japan.

Kay, P. (1993). "Speech-driven Graphics: A User Interface." Journal of Microcomputer Applications **16**(3): 223-231.

Looser, J., R. Grasset, et al. (2006). OSGART - A Pragmatic Approach to MR. Industrial Workshop at ISMAR 2006, Santa Barbara, CA, USA.

MicrosoftSpeech (2007). http://www.microsoft.com/speech/default.mspx: accessed August 2007.

Milgram, P., S. Zhai, et al. (1993). Applications of Augmented Reality for Human-Robot Communication. In Proceedings of IROS 93: International Conference on Intelligent Robots and Systems, Yokohama, Japan.

Murphy, R. R. (2004). "Human-robot interaction in rescue robotics." Systems, Man and Cybernetics, Part C, IEEE Transactions on **34**(2): 138-153.

Nintendo (2008). http://www.nintendo.com/wii: last accessed July 2008.

Nourbakhsh, I. R., J. Bobenage, et al. (1999). "Affective mobile robot educator with a full-time job." Artificial Intelligence **114**(1-2): 95-124.

NXT++ (2007). www.nxtpp.sourceforge.net/index.php: accessed August 2007.

Open Scene Graph (2008). www.openscenegraph.org: accessed June 2008.

Scholtz, J., B. Antonishek, et al. (2005). A Comparison of Situation Awareness Techniques for Human-Robot Interaction in Urban Search and Rescue. CHI 2005, April 2- 7, Portland, Oregon, USA.

Sidner, C. L. and C. Lee (2005). "Robots as laboratory hosts." Interactions **12**(2): 24-26.

Skubic, M., D. Perzanowski, et al. (2004). "Spatial language for human-robot dialogs." Systems, Man and Cybernetics, Part C, IEEE Transactions on **34**(2): 154-167.

The Lego Group (2007). http://mindstorms.lego.com/: accessed August 2007.

Thrun, S. (2004). "Toward a Framework for Human-Robot Interaction." Human-Computer Interaction **19**: 9-24.

Tsoukalas, L. H. and D. T. Bargiotas (1996). Modeling instructible robots for waste disposal applications. Proceedings of the 1996 IEEE International Joint Symposia on Intelligence and Systems, Nov 4-5, Rockville, MD, USA, IEEE, Los Alamitos, CA, USA.

Yanco, H. A., J. L. Drury, et al. (2004). "Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition." Human-Computer Interaction Human-Robot Interaction **19**(1-2): 117-149.

ZeroC (2008). http://www.zeroc.com/: accessed June 2008.

13

Indoor Localization Techniques based on Wireless Sensor Networks

Hyo-Sung Ahn[1] and Wonpil Yu[2]
[1]*Department of Mechatronics, Gwangju Institute of Science and Technology (GIST)*
[2]*Electronics and Telecommunications Research Institute (ETRI)*
Korea

1. Introduction

Indoor localization is one of the most important problems in intelligent service robots, and home and office automation. For mobile robot navigation usually vision-based image processing techniques and dead-reckoning techniques based on inertial navigation systems have been used. These traditional technologies however have revealed many problems in actual applications. Vision-based image processing requires landmarks that should be sequentially processed via image detection, feature extraction and scene matching techniques. In actual applications it is hard to extract feature from environment and process data in a real-time on the mobile robot platform. Performance of inertial navigation systems highly depends on the specifications of gyroscope and accelerometer mounted on the robot platform. Measurements from these types of equipment contain various types of random and bias noises. Furthermore since measurement error is usually accumulated, the performance of dead-reckoning system is substantially degraded as time passes. Thus, it is seen that the traditional navigation techniques such as vision and inertial navigation systems are not trustworthy in actual applications.

Recently, with the progress of wireless communication techniques, sensor network-based localization schemes have been actively researched. It is shown that localization techniques on the base of wireless sensor networks are able to overcome many weak points of traditional navigation systems. Wireless sensor network-based localization techniques particularly appear to be beneficial for indoor applications. The main motivation of this chapter is thus to provide a comprehensive overview on state-of-the-art wireless localization networks and a recent progress in this field.

Indoor localization problems are considered much more difficult than outdoor localization problems because GPS signals are not available within buildings or nearby huge structure. Since there are lots of signal interference and signal reflection inside the building, it is even hard to range the signal propagation length using wireless RF signals. Typically in indoor localization, problems of tens-of-meters-long distance or less than ten meters distance are primarily concerns. Thus, if wireless communication signals are contaminated by interference and/or path loss, then the estimated range may include lots of errors. Thus indoor localization is much tougher than outdoor localization. Note that the measurement noises and estimated errors on the base of wireless sensor networks (WSN) have different characteristics from vision-based navigation and dead-reckoning systems; that is, measurement errors contained in the measurements of wireless sensor network are not accumulated, which is a case in dead-reckoning systems. Also since wireless sensor networks use wireless communication signals, it is relatively more robust against obstacles than vision-based systems. Moreover, it does not use any landmarks; thus computational load is substantially reduced compared with vision-based systems. However the performance of indoor localization based on wireless sensor network is environmental-dependent. Thus, even though it has many advantages over traditional approaches, it has also weak points.

The main problem considered in this chapter is thus to introduce some novel approaches to handle this problem. The major results presented in this chapter are based on authors' previous works (Ahn & Yu, 2006; Ahn et al, 2007[1]; Ahn et al, 2007[2]; Ahn et al, 2008[1]; Ahn & Yu, 2008[2]; Ahn & Yu, 2008[3]; Ahn & Yu, 2008[4]). Before presenting the alternative approaches developed by the present authors, in the following section we first take a brief review on some relevant works.

2. Related Works

There are various aspects in WSN-based indoor localization technologies in terms of communication methodologies, measurement attributes, and communication protocols. In this section we first discuss some recent research trends in this field, and then we categorize indoor localization techniques.

2.1 Indoor Localization and Navigation

In the field of WSN-based indoor localization, in fact there is a huge amount of publications. An outstanding survey was carried out in 2001 by (Hightower & Borriello, 2001) where Active Badge, Active Bat, Cricket, RADAR, MotionStar DC magnetic tracker, etc. were introduced. In (Ssu et al, 2005), they proposed a localization algorithm based on an inspiration from the perpendicular bisector of a chord conjecture. However they developed an algorithm under some restrictive assumptions. In (Peyrard et al, 2000), a localization protocol for mobile stations in a wireless local area network (WLAN; *IEEE 802.11*) was developed. In their approach the localization was carried out in the scale of extended service set and basic service set (a similar approach can be found in (Elnahrawy et al, 2004)). There also has been lots of research for ultra wideband (UWB; *IEEE 802.15.4a*)-based localization systems. In particular there have been various research efforts such as outdoor localization application of UWB (Oppermann et al, 2004), UWB localization algorithms (Yu & Montillet et al, 2006), introduction of a commercialized product (Fontana et al, 2003), channel modelling of UWB signal (Irahhauten et al, 2004), and an overall description about the UWB-based localization (Ingram et al, 2004).

The navigation system is divided into the outdoor navigation and the indoor navigation. For the outdoor case, global positioning system (GPS) can provide reliable positioning information with a favourable accuracy, whereas there is no dominant solution for the indoor object localization. From the literature (Patwari et al, 2005; Sun et al, 2005), it is seen that the indoor navigation can be further categorized into the vision-based navigation system, where the object recognition is a main concern, and the WSN-based navigation system, where the signal detection and time synchronization between sensors are important considerations. Recently Federal Communication Commission (FCC) of the USA required the wireless-service providers to accurately locate the position of the 911 caller, and this requirement has accelerated the development of indoor localization systems based on WSN technologies. In this section, as background material, we classify the WSN-based indoor localization systems into several categories.

2.2 Classification by Measurement Attribute

Indoor localization technologies have been developed on various concepts and aspects. First, we can consider two different types based on the placement of the main computational unit; *client-based systems* and *server-based systems*. In client-based systems the client receives signals from the distributed signal sources. The client is equipped with tags, adapters, or receivers. The client interprets the received-signals as ambient information for localizing its position in a local coordinate system. The server-based methods measure the signals radiated from a client. Then using the signals measured at distributed sensors, the server estimates the position of the client (the fingerprinting can be considered as a server-based approach (Bhargava et al, 2005)). Localization systems based on ad-hoc sensor networks, wherein sensors are distributed in a ubiquitous space as communication nodes, however cannot be categorized as either the client-based or the server-based, because beacons can be placed within a cluster of sensors as reference points. Instead it is a combination of the client-based and the server-based localization systems.

According to the mobility of reference beacons we can also categorize a localization method as one out of the following three: *localization systems with fixed-beacons*, *localization systems with mobile beacons*, and *beacon-free localization systems*. According to the availability of distance information in the localization, we can further differentiate *range-based* methods from *range-free* methods, even though most of localization methods can be classified as range-based methods. For a range-free method, see (He et al, 2003). The accuracy of range-free methods is not good while it provides quite reliable positioning information.

2.3 Classification by Localization Algorithms

According to the algorithms used for the localization, we can categorize algorithms into several different types. RSSI (*received signal strength index*) method uses signal strength arrived, and uses a relationship between the signal-loss and the propagation distance. The following formula is usually used for this relationship:

$$I(r) = c/r^a \qquad (1)$$

where I(r) represents signal strength at distance r and c can be a constant or environment-dependent variation.

However, the signal attenuation parameter, a, is not fixed, but it depends on the situation. TOA (*time of arrival*) method uses time-of-flight of the radio signal transmitted. However, in a short distance, it is extremely difficult to calculate the time-of-flight. Furthermore in this method, it is very tough to establish a highly accurate positioning system, since the time synchronization is preliminarily required. To handle the difficulty of this time synchronization, an idea of using the round-trip signal has been proposed (Ping, 2003). However, in this case, usually the size of tag becomes larger because a powerful transceiver is necessary. TDOA (*time difference of arrival*) method also uses time-of-flight; but it does not require the time synchronization between senders and receivers. Instead, it requires time synchronization between receivers. AOA (*angle of arrival*) method measures the emitted-direction and/or received-direction of signal at fixed reference receivers. Using these directions, the position of the signal source is determined by the triangulation. In TOA and AOA, the array of antennas is an important issue (Sayed et al, 2005), and usually various filtering techniques are used for the signal detection (Gustafsson & Gunnarsson, 2005). Digital map information-based method (it is called *fingerprinting*) (Li et al, 2005) makes use of strengths of radio signals radiated from access points (APs). A reference radio map is previously generated based on the signal strengths. When the positions of APs are fixed, the radio map will then not change much, which enables the use of the reference map when estimating the position of a tag. However, note that the radio signal strength is time-dependent; hence it is varying even at the same position (Ahn et al, 2007[2]). Thus, the fingerprinting method does not guarantee the accuracy of several meters; instead it can be used for room-level localization (Ahn et al, 2007[2]). Various hybrid methods, which fuse two or more algorithms given above, also have been proposed for a performance improvement (Sayed et al, 2005; Gustafsson & Gunnarsson, 2005).

However, there are many error sources in the above algorithms. For example, when using TOA and TDOA algorithms, the line-of-sight (LOS) between senders and receivers as well as the timing synchronization between sensors should be ensured, which is usually not satisfied in actual applications. Besides the timing synchronization and LOS, various error sources such as channel fading, shadowing, low signal-to-noise ratios (SNRs), multi-user interference, and multi-path effects deteriorate the performance.

2.4 Classification by Communication Protocols

Although various signal/communication techniques such as RFID, laser, sonar, radar, infrared, etc. have been used for the indoor navigation, recently Wi-Fi (IEEE 802.11), UWB (IEEE 802.15.4a), and ZigBee (IEEE 802.15.4) have attracted much attention as introduced in the previous section. Particularly, there have been commercial Wi-Fi and UWB products for the indoor localization. Wi-Fi technology is based on WLAN, which is standardized by IEEE 802.11 a/b/g (Rehim, 2004).

In Wi-Fi-based methods, the fingerprinting approach is most popular. However this method needs a reference map that must be generated based on radio signal measurements in advance. Thus, this method is quite sensitive to the environmental variations (Ladd et al, 2004; Xiang et al, 2004). UWB communication is used for WPAN, which is standardized by IEEE 802.15.3a and IEEE 802.15.4a (Gezici et al, 2005; Crepaldi, 2005). UWB system is not much affected by environmental variations, because the signal band width is ultra wide. Thus, recently lots of research and development efforts have been devoted to the UWB-based localization.

In (Guoping & Rao, 2005), a set of UWB-prototype composed of tag, reader and antenna was developed for the localization purpose. The leading edge of the received-pulse is detected by a maximum likelihood method, which reduces noises and interference effects. However, in this system, the size of UWB transmitter is too big, so it cannot be used for the localization purpose in ubiquitous space. In (Ruiz et al, 2005), a typical structure of UWB chipset was introduced. The typical UWB transmitter is composed of modulator, control digital processor, digital to analog interfaces, pulse generator module, amplifier, filtering, and antenna. The UWB receiver consists of antenna, LNA, a variable attenuator, peak hold detector, analog to digital converter, and band pass filter. As explained in (Guoping & Rao, 2005), UWB-based localization has some advantages over other localization techniques: it uses very narrow pulse (~250 ps) signal; so the signal-arrival instant can be calculated by the receiver accurately if the signal pulse is detected. Also since the narrow pulse signal is immune to multi-paths and free-space fading, UWB is known as an appropriate localization technique in a complicated indoor environment.

However, it is still tough to get rid of multi-paths completely (Bocquet et al, 2005). Moreover, the UWB signal, which ranges from 3.1 to 10.6 GHz, coexists with aeronautical and marine radars (Larson et al, 2003); so an appropriate interference suppression method is still necessary when designing transmitter and receiver.

A most challenging problem in the UWB-based localization is the time synchronization between receivers, or between receivers and the transmitter. As an existing technique for this time synchronization, in (Guoping & Rao, 2005), digital phase lock loop (DPLL) and a lock indicator were used. Actually, it is a toughest thing to ensure the time synchronization of the GHz-UWB systems using the MHz clock (or GHz clock) generator. As a solution for this, only the clock alignment between transmitted-signals and receiver clocks was required (Ruiz et al, 2005). In fact, there are numerous publications for the theoretical developments of UWB algorithms as explained in (Lin et al, 2004). However, a successful and practically-useful system for localizing any asset has not been reported yet in the literature.

As an alternative to Wi-Fi and UWB, recently IEEE 802.15.4 (ZigBee)-based localization system has been introduced by Chipcon (www.chipcon.com). However, the method used in cc2420 and cc2431 chipsets needs an accurate signal propagation model; hence it is very sensitive to the environmental variation. Probably one of the most advanced technologies in this field is chirp spread spectrum (CSS)-based localization system. A chirp pulse is a frequency modulated signal; it has up-chirp, down-chirp and combination of up-chirp and down-chirp. Since the up-chirp and down-chirp have time-variant frequency, it has a wide bandwidth and uniform power spectral density. Due to the wide bandwidth of CSS, it is highly robust against obstacle, interference, and multi-path. Like UWB, the physical layer of CSS is also standardized by IEEE 802.15.4a.

3. State-of-the-Art Technologies

This section presents some technical background and experimental test results of the state-of-the-art WSN-based indoor localizations.

3.1 Ultra Wide-band (UWB)

In UWB, there are two different types of signals: Impulse UWB (I-UWB) and multicarrier UWB (MC-UWB). MC-UWB is for the data communication due to its high data rate characteristic. The modulation of MC-UWB is based on orthogonal frequency division multiplexing (OFDM). I-UWB does not use a traditional sinusoidal wave for its modulation. Instead, it uses a series of pulse signals whose pulse duration is extremely short, which means it is a ultra wideband in its frequency bandwidth and is suitable for the indoor localization. Generally, however it is still tough to eliminate narrow and wide band noises.

Thus, special attentions are required when designing transmitter and receiver (Reed, 2005; Oppermann et al, 2004). In our experimental test, we used Ubisense products (www.ubisense.net), because it was seen that Ubisense has overcome these technical challenges in some respects. It is quite straightforward to install Ubisense systems; so a detailed explanation is not given in this chapter. From experimental test (Ahn & Yu, 2006; Ahn et al, 2007[1]), we found that the Ubisense system can provide about one or two meter accuracy of stationary assets. However, the Ubisense system requires heavy hardware equipment and many cables for data transfer, power, and time synchronization. Also, the hardware size of reference nodes is too big to be used in the indoor service robot.

3.2 Wi-Fi

The WLAN technology has an advantage of using license-free radio area as well as of operating in low-power. The WLAN is composed of wireless LAN card, access points (APs), and LAN bridge. The LAN card operates as an interface between network operating system (NOS) and a client like a laptop, and the APs can be seen as LAN hub for interfacing wired network and WLAN area. In WLAN-based localization systems, there are two main issues. The first issue is associated with the size of tag. When monitoring the target, the size of tag (i.e., client) is required to be small such that it can be attached to an object easily. The second issue is related to the generation of a reference radio map. It usually takes a time to generate the radio map and the radio signal strength is a function of time.

For the localization system used in service robot, we have used Positioning Engine 3.1 of Ekahau3 (www.ekahau.com), because it provides a portable tag and user-friendly development software (API), and a nice performance as reported in (Gifford, 2005). Another reason for selecting Ekahau as the sensor network is that it does not require any hardware equipment for its operation. That is, Ekahau is purely software-based localization system. It is also straightforward to use Ekahau system; so a detailed description is not provided in this chapter. From experimental tests (Ahn et al, 2007[2]), we found that WLAN-based localization system can be used for zone-to-zone navigation and room-level localization of stationary objects (Ahn et al, 2007[2]).

3.3 ZigBee

The ZigBee communication protocols are defined in IEEE 802.15.4a wireless medium access control (MAC) and physical layer (PHY), and ZigBee Specification generated by ZigBee alliance, Inc. The ZigBee alliance added the network layer (NWT) and the network framework in the application layer based on the IEEE 802.15.4 MAC and PHY. The ZigBee communication is effective for the short range information exchange. The key feature is low-power and no infrastructure required. Unlike UWB, however ZigBee is not designed for the ranging estimation and positioning. Thus, there are relatively few ZigBee products for indoor localization. Even though the ZigBee signal strength can be used for a range detection purpose, it is difficult to keep the constant signal strength. Also, since the signal strength is dependent on the environmental variation, a fixed signal propagation model cannot be used for a ranging purpose.

3.4 CSS

The chirp spread spectrum (CSS)-based communication protocol and chipset technology is shown to dominate the indoor localization market (Ahn et al, 2008[1]). The major advantage of the CSS technology is to provide robust performance for LR-WPAN (low rate wireless personal area network) even in the presence of path loss, multi-path, and reflection. The center frequency of chirp pulse is 2.44GHz, chirp duration is 1 μs, and usaully the chirp bandwidth is 80MHz. Since Doppler effect of chirp signal is ignorable, it is particularly useful for mobile applicaations. Up-chirp or down-chirp signal is detected by autocorrelation of the same signal. When the same signal is auto-correlated with a signal that has been time-synchronized, a pulse will appear. The magnitude of the pulse is dependent on the accuracy of time synchronization between two convoluted signals.

The ranging technique outlined in IEEE 802.15.4a is called symmetric double sided two-way ranging (SDTWR). The SDTWR is based on the time duration that takes for a round-trip between a transmitter and receiver. NanoTron has commercialized a localization chipset that is based on CSS communications. In nanoLOC, the range between a transmitter and receiver is estimated by compensating for reply time within nodes. Recently the real-time one way ranging technique (RT-OWRT) was established, which localizes the tag using one-way communication signals (Ahn et al, 2008[1]).

3.5 Comments

In (Ahn et al, 2007[1;] Ahn et al, 2007[2]; Hur & Ahn, 2008), experimental test results of UWB, Wi-Fi, ZigBee, CSS are presented. In the case of Ubisense (UWB), overall accuracy is about 1~2 meters when the target is stationary for a long time and Ekahau (Wi-Fi) is only valid for the room-level position estimation. In the case of Chipcon cc2431 (ZigBee), the position is detected within about 5 meters. NanoLOC (CSS) provides localization information with 2~5 meters accuracy. It was shown that the accuracy of localization is highly dependent on the mobility of target. If the target is stationary, the accuracy is improved while if it is mobile, then the accuracy is not reliable.

From experimental tests it is also observed that the performance of localization systems is affected by the environment. Table 1 summarizes the advantage and disadvantage of individual WSN localization system. As shown in this table, no one is a dominant solution for indoor object localization. Also, since the WSN-based localization systems of Table 1 are server-based methods, many tags cannot be tracked at a time.

In the following section, we address a new localization technique that is very cheap to implement and can provide stable position information of stationary objects. Also, since the new method is a client-based localization, many tags can be localized at a time. The new method is developed based on received signal strength index (RSSI) of the radio signal radiating from fixed reference nodes. For the signal strength detection and for obtaining a reliable signal propagation model, the reference tags are used.

	Wi-Fi	ZigBee	UWB	CSS
Disadvantages	-Low accuracy -Slow response -Signal interference -Radio map is required	-Signal propagation model is required -Time-dependent signal characteristics	-Heavy hardware -Difficult to locate mobile objects -Expensive equipment	-Expensive equipment
Advantages	-No additional infrastructure -Stable position information	-Simple hardware -Small tag size	-Relatively accurate localization of stationary objects	-Relatively reliable and precise localization

Table 1: Comparison of performance of WSN-based localization systems

4. Environmental Adaptive Indoor Localization Techniques

As mentioned in the previous section, the existing localization technologies have their own disadvantages and advantages. One of the challenging problems in indoor localization based on WSN is the sensitivity of RF signals in indoor environment. To resolve this problem in this section we outline environmental adaptive indoor localization techniques, which were developed in our earlier research works (Ahn & Yu, 2008[2; Ahn & Yu, 2008[3]; Ahn & Yu, 2008[4]).

This section presents a new indoor localization method based on RSSI in an office environment. The RSSI-based indoor localization has attracted a lot of research recently due to its simplicity and low cost as explained in (Zhang et al, 2006), and due to a less sensitivity to the bandwidth and occurrence of undetected direct path (Hatami & Pahlavan, 2006). The newly-proposed RSSI-based method consists of reference nodes, which receive and send radio signals, and a tag, which computes its position using radio propagation model updated in real time based on received signal strength. Thus, the newly-proposed method can be considered as a client-based method.

The tag measures received-signal strength and calculates the distance between reference nodes and the tag. The tag is attached to an object that is a target to be tracked, and reference nodes are placed at reference points. A reference node is composed of transmitter and receiver. The receiver computes model parameters based on received-strengths of signals, which are sent from other reference nodes.

The left figure of Figure 1 depicts reference nodes and a tag. This figure shows that the radio signal emitted from a transmitter in a reference node is received by its own receiver, by the receivers in other reference nodes, and by the tag. Even though this figure illustrates three reference nodes, the number of reference nodes and tags generally can be extended to N.

In Equation (1), we need to calculate model parameters a and c based on received signal strengths. If the model parameters a and c are known, then using the measured signal strength I(r), we can reversely calculate the distance between reference nodes and the tag. In what follows, we will explain the algorithm briefly (for a more comprehensive description, refer to Ahn & Yu, 2008[4]). The signal attenuation parameter a is sensitive to the environmental variation and the initial signal strength c is a function of time. It is assumed that the positions of reference nodes are known. Also it is supposed that when the tag and receivers receive signals, they can know where the signal comes from. For convenience, let us say that the receiver of the reference node #i receives its own signal with strength c_i and receives the signal of reference node #j with strength c_i^j. Also, the tag receives the signal of the reference node #i with strength I_i. The distance between the reference node #i and the tag is denoted as r_i, and signal attenuation parameter is a_i. Then, unknown parameters are r_i and a_i, and measured values are c_i^j and I_i. Now, the problem is to find r_i and a_i as accurately as possible using measurements c_i^j and I_i.

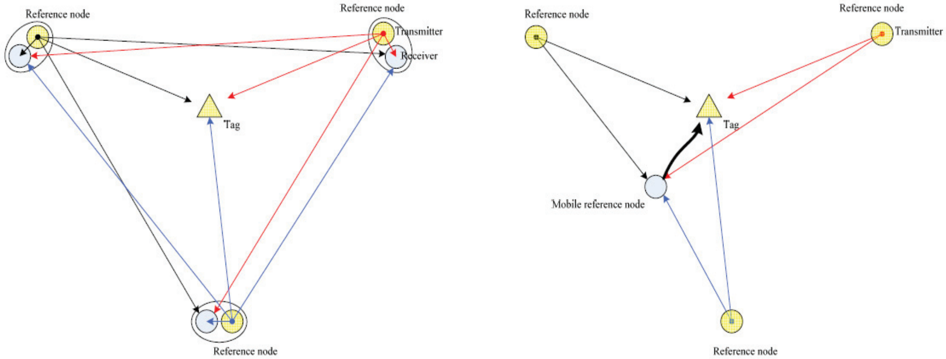


Fig. 1. Left: Localization network composed of three reference nodes (transmitter and receiver; N=3) and a tag. Right: Localization network using a mobile reference node.

A simple solution for the parameter estimation is to use the fixed length between the reference node #i and the reference node #j, i ≠ j. That is, since reference nodes are fixed at known points, and c_i and c_j are related by the following equation:

$$c_i^j = c_j/(r_i^j)^{a_j} \tag{2}$$

where r_i^j is the distance between the reference node #i and the reference node #j, and c_i^j is a signal strength of the node #j received by the node #i. Therefore, a_j is simply calculated as

$$a_j = \ln^{c_j/c_i^j} / \ln^{r_i^j} \tag{3}$$

Thus, attenuation parameters a_i, i=1,…,N calculated by Equation (3) are used with together I_i for the distance, r_i, i=1,…,N, calculation in Eq. (1). However, Equation (3) calculates the attenuation parameters in a simple manner and it does not filter out measurement noises. So, as an alternative, we can conceive a method that uses average of received signals. That is, for the calculation of a_j, we use all c_i, i=1,…,N (i≠j) in the following way:

$$a_j = 1/(N-1)\sum_{i=1,\, i\neq j}^{N} [\ln^{c_j/c_i^j} / \ln^{r_i^j}] \tag{4}$$

which will remove some high frequency noise components and uncompensated bias errors. The overall procedure can be summarized in the sequel.

The receivers, which are placed in reference nodes, and the tag receives signals radiating from the transmitter of individual reference node. Then, reference nodes send information of measured signal strengths to the tag using wireless local area network so that the tag can estimate a_j using Equation (3) or Equation (4). Then, using Equation (1), the tag calculates distances between reference nodes and it such as

$$r_i = [c_i/I_i]^{1/a_i} \tag{5}$$

Now, using Equations (5) it is straightforward to estimate the position of the tag. A nice review on the localization algorithms is presented in (Sayed et al, 2005). In (Sayed et al, 2005), if the positions of reference nodes are known as (x_i, y_i), then the position of the tag is calculated as:

$$[x, y]^T=[H^TH]^{-1}H^Tb$$

where:

$$H=\begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix};\ b=\begin{bmatrix} K_2^2 - r_2^2 + r_1^2 \\ K_3^2 - r_3^2 + r_1^2 \\ K_4^2 - r_4^2 + r_1^2 \\ \vdots \\ K_N^2 - r_N^2 + r_1^2 \end{bmatrix}$$

and $K_i^2=x_i^2+y_i^2$.

So far, we have presented a localization method using reference nodes which include both transmitter and receiver. The receiver receives radio signal sent from transmitter and it measures the signal strength for calculating signal propagation parameters. However, the communication load may be heavy in this method.

Another simple localization method also can be developed using a reference tag without receivers in reference nodes. The right figure of Figure 1. depicts the wireless localization network that is composed of a mobile reference node, fixed reference nodes, and a tag. The mobile reference node is movable and its position is assumed known. This assumption is valid in a mobile robot application. The mobile robot position is measured from localization sensor network such as StarLITE (Yu, W. et al, 2006; Chae et al, 2005). In the right figure of Figure 1., the tag can be attached to pedestrian.

In this method, the mobile reference node is used for modeling the signal propagation characteristics. The signal propagation is governed by the following formula:

$$I_i(r) =A/r_i^a\ I_i(0) \qquad (7)$$

where $I_i(r)$ is the received strength of the i-th reference node signal; r_i is the distance between the mobile reference node and the i-th reference node; and $I_i(0)$ is the initial signal strength of the i-th reference node.

Now, the problem is to find the signal attenuation parameter a and proportional constant A based on $I_i(r)$, r_i, and $I_i(0)$. Since in this approach the initial signal strengths $I_i(0)$ are not calculated, $I_i(0)$ should be measured first. The relationship given by Equations (7) can be changed as:

$$a\ \ln^{r}{}_{i} = \ln[I_i(0)/\ I_i(r)] + \ln A \qquad (8)$$

From Equations (8), we can easily obtain the following relationships:

$$a(\ln^{r}{}_{1} - \ln^{r}{}_{2}) = \ln[I_1(0)/\ I_1(r)] - \ln[I_2(0)/\ I_2(r)] \qquad (9)$$

$$\vdots$$

$$a(\ln^{r}{}_{N-1} - \ln^{r}{}_{N}) = \ln[I_{N-1}(0)/\ I_{N-1}(r)] - \ln[I_N(0)/\ I_N(r)] \qquad (10)$$

Equations (9-10) can be now used for finding the attenuation parameter a. A more accurate estimation can be performed by combination of all measurements. The combination is achieved by the following formula:

$$a(\ln^{r}{}_{i} - \ln^{r}{}_{j}) = \ln[I_i(0)/\ I_i(r)] - \ln[I_j(0)/\ I_j(r)],\ \forall\, i,j,\ i \neq j \qquad (11)$$

Therefore, the parameter a is estimated as:

$$a = 1/N(N-1)\ \sum_{j=1}^{N} \sum_{i=1,\, i \neq j}^{N} (\ln[I_i(0)/\ I_i(r)] - \ln[I_j(0)/\ I_j(r)])/(\ln^{r}{}_{i} - \ln^{r}{}_{j}) \qquad (12)$$

The proportional constant A is obtained by inserting a into Equations (8).

5. Hardware Development

For the verification of the localization algorithms described in the previous section, we developed a ubiquitous ZigBee (2.4 GHz RF communication system) sensor network. The reason we selected the ZigBee communication system as our hardware platform is two-fold: First, it is very convenient and cheap to use ZigBee transceiver as reference node. Second, we can easily add a motion sensor to ZigBee system so that the tag senses the motion of object, which can be nicely used for saving the power and for reliable object localization. The ZigBee system developed thus operates in sleep mode and operating mode according to events; hence, the battery life of the tag is relatively longer than other wireless communication systems. We developed fixed reference transmitters and fixed reference tags, which can be used as receivers in the reference nodes. That is, we place the fixed reference tags nearby the reference transmitters. In what follows, we explain the hardware equipment in detail.

5.1 Fixed Reference Transmitter

The basic requirements for the fixed reference node are as follows:

- Nodes communicate each other wirelessly and only can be wire-connected to the power outlet
- The user can assign coordination values to individual reference node
- When a battery is used as a power source, it should operates in either sleep mode or active mode based on event
- When there is a request from the moving tag, it must send its coordination to the tag
- The fixed reference node can be easily attached to a wall and a ceiling
- The hardware size is limited to 15cm x 10cm x 2cm
- Each node has its own identity and can encode its identity in the data packet

To satisfy the above requirements, we chose the following components:

- ATmega128L(MCU): 128K Flash, 4K SRAM, 4K EEPROM, 2.56V~3.3V operation volts
- cc2420(RF Transceiver): 2.4 GHz RF communication, 16 Channels, RSSI detection
- SMD Poll type antenna

The firmware and protocol stack are embedded into ATmega128L to manage overall activities of the system. The cc2420 sends and receives data packet based on ATmega128L's control signals.

5.2 Mobile Reference Tag

This mobile reference tag can used as the receiver of individual reference node and can be used for mobile reference node. Thus, it is used for the tag, which can be attached to object. The requirements for the mobile reference tag are:

- To enable the mobile capability, it should be easily mounted on a mobile vehicle like mobile robot, object, pedestrian, etc.
- When it is attached to the mobile robot, it must receive the position of mobile robot through electrical interface (it is assumed that the position of mobile robot is known)
- When there is a request from the other reference tag, it should deliver its position and received-strengths of the signals emitted from the reference nodes
- Individual mobile reference tag has its own identity
- When it is mounted on the mobile robot, the power cable can be connected to the robot

The particular requirements for mobile tag, when it is attached to object, are given as:

- The tag is composed of a motion sensing module and a ZigBee RF module
- The tag operates based on power of the battery
- Physical size is limited as 8cm x 5cm x 1cm and weight is less than 150g
- The tag operates in either sleep mode or active mode according to the motion signal detected by the motion sensor
- The tag should receive signals from the fixed reference nodes and from other mobile tag
- The tag is able to calculate the signal propagation parameters in real time
- The calculated signal propagation information should be transmitted to the server, if necessary

The detailed hardware specification developed at ETRI is summarized as follows:

- cc2431(MCU & RF Transceiver): 128K Flash, 8K SRAM, 4K EEPROM, 250Kbps transmit rate, Multi-purpose general I/O port
- Fractus Chip antenna
- I/O: 2 LEDs, 1 Push button, 1 Reset button, Download Port

Firmware and IEEE 802.15.4 protocol are embedded on cc2431. Figure 2 is the mobile reference tag developed. In this picture, accelerometer is the motion sensor.

6. Experimental Results

For the experimental verification of the algorithm proposed in the previous section (the algorithm described by Equations (2)-(6); let us call this method as the first method), we constructed a wireless sensor network as shown in Figure 3, Figure 4 and Figure 5. From Figure 5, we know that fixed reference nodes, which are composed of fixed reference transmitter and mobile reference tag, are placed at points (10.5, 14.5), (10.5, 8.0), (10.5, 0.0), (0.0, 14.5), (0.0, 8.0) where the unit of the coordinate value is meter.

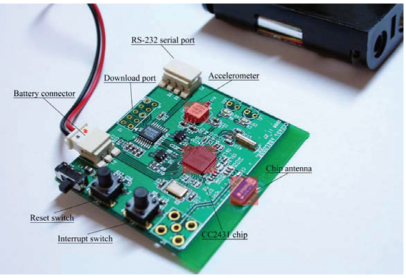


Fig. 2. Mobile reference tag produced.

The fixed-reference transmitter developed does not operate as a transceiver. So, we place mobile reference tag, which can receive signals from the fixed reference transmitter, beside the fixed-reference transmitter so that they (the fixed-reference transmitter and mobile reference tag) work as a reference node in a pair. Note that these nodes are placed at three-meter height above the surface and under the ceiling; but we do not take account of the height.

Figure 3. shows that the reference nodes are composed of mobile reference tag and fixed-reference transmitter. The left figure is a reference node attached on the wall and the right figure is a reference node attached under the ceiling. Figure 4 shows the reference nodes installed in office. We place tags at (6.6, 9.8) and (10.5, 8.0) respectively, as shown in Figure 5. (the height is about 1.6 meters). The place of (6.6, 9.8) is within the office and the place of (10.5, 8.0) is near the wall.

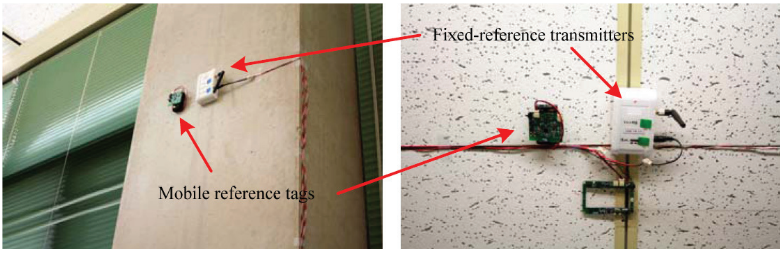


Fig. 3. Fixed reference node is composed of a reference transmitter and a mobile reference tag: the left-top is the node installed in the wall; the right-top is the node installed under the ceiling.

The experiment was performed for 10 minutes. At each sampling time, we obtained a data packet from the tag as shown in Figure 6. Individual mobile reference tag measures RSSI of the signal sent from the fixed-reference transmitters. Then, each mobile reference tag sends the data packet as shown in Figure 6. to the tag. In the packet, the last two data x and y are position of the tag calculated by Chipcon cc2431 chip mounted in the tag. So, we can compare the estimated-positions of the tag calculated by the proposed-algorithm and by the Chipcon cc2431.

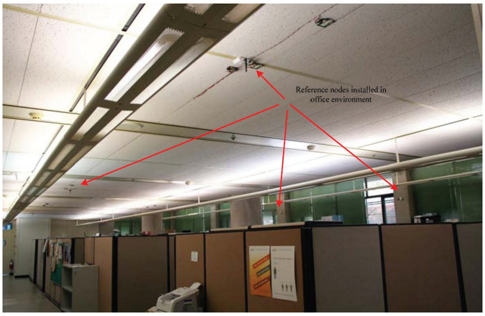


Fig. 4. Fixed reference nodes are installed in office.

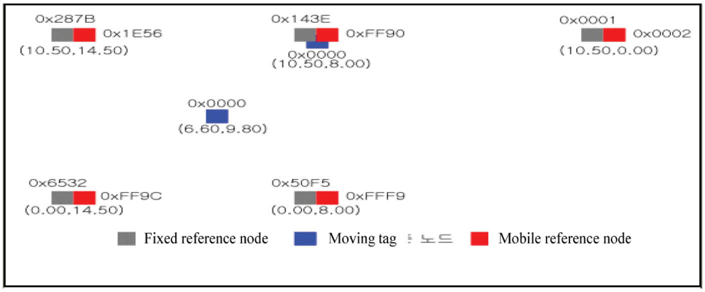


Fig. 5. Fixed reference nodes, moving reference nodes, and tag.

Mobile reference node ID	RSSI of 0x0001	RSSI of 0x143E	RSSI of 0x287B	RSSI of 0x6532	RSSI of 0x50F5	X	Y

Fig. 6. Data packet.

Figure 7-(a) to Figure 7-(f) show experimental test results when the tag is placed at (6.6, 9.8). Figure 7-(a) shows the estimated a_j from Equation (4). From the plots, it is shown that the attenuation parameters are estimated as about 3 except a_4, which is estimated as 4 approximately. Figure 7-(b) shows the estimated distances calculated by Equation (5). The dots in Figure 7-(c) are the estimated positions of the tag from the proposed-method and the **x**-mark represents the actual position of the tag. The dots in Figure 7-(d) show the estimated positions of the tag from cc2431. From Figure 7-(d), we can see that cc2431 estimates the tag position in several fixed points, whereas the proposed-method estimates the position in a random manner.

Since this chapter estimates the stationary object, we recalculate the position based on Figure 7-(c) and Figure 7-(d) in the following way:

$$x_n = 1/n \sum_{i=1}^{n} x_i, \; y_n = 1/n \sum_{i=1}^{n} y_i \qquad (13)$$

where x_i are y_i values from Figure 7-(c) and Figure 7-(d). Figure 7-(e) shows the re-calculated x_n and y_n. In Figure 7-(e), the solid lines are the re-calculated x and y of the tag from cc2431; the dot-dashed are the re-calculated x and y of the tag from the proposed method; and the dashed-lines are actual values. From these re-calculated plots, we can see that the proposed method is better in the estimation of y; but the cc2431 is better in the calculation of x.

To compare the performance quantitatively, we calculate the total errors of Figure 7-(e), which is shown in Figure 7-(f). The total errors are calculated as

$$e_n = \sqrt{(x_n - x)^2 + (y_n - y)^2} \tag{14}$$

where x, y are true position values of the tag. From Figure 7-(f), we can see that the proposed-method has a better performance than cc2431. Figure 8-(a) to Figure 8-(f) show the test result when the tag is placed nearby the wall.

As shown in Figure 8-(f), the proposed-method has a better performance than cc2431. For the experimental test of the algorithm described by Equations (7-12) (let us call this method as the second method), we carried out experimental tests by repetition. For this test, we simply turned-off all mobile reference tags shown in Figures 3-4, and placed a reference mobile reference tag inside the office. This reference tag placed inside the office acts as a mobile reference.

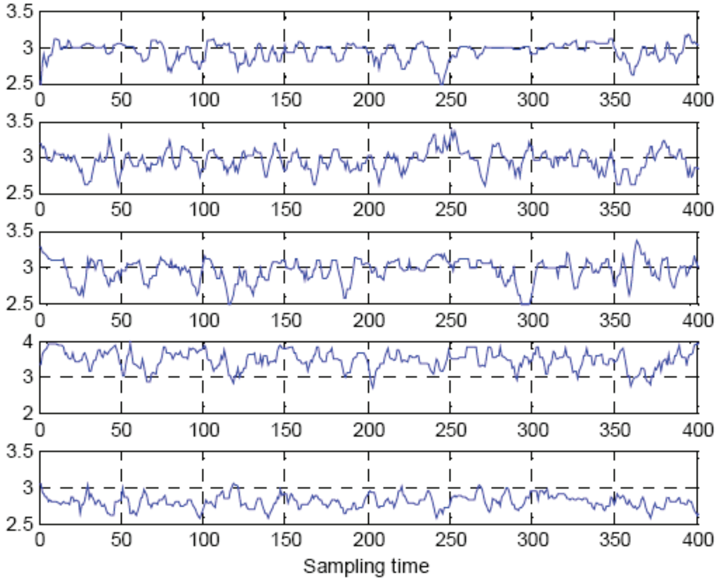


Fig. 7-(a). Experimental test results inside the sensor network.

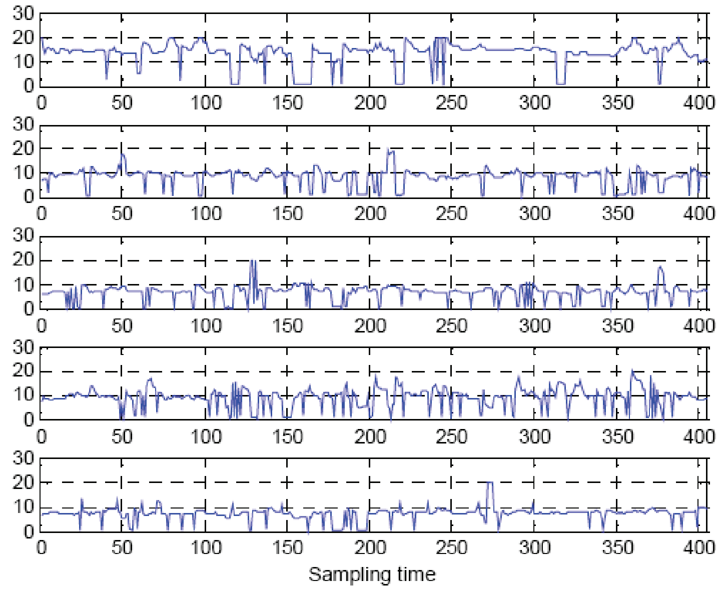


Fig. 7-(b). Experimental test results inside the sensor network.

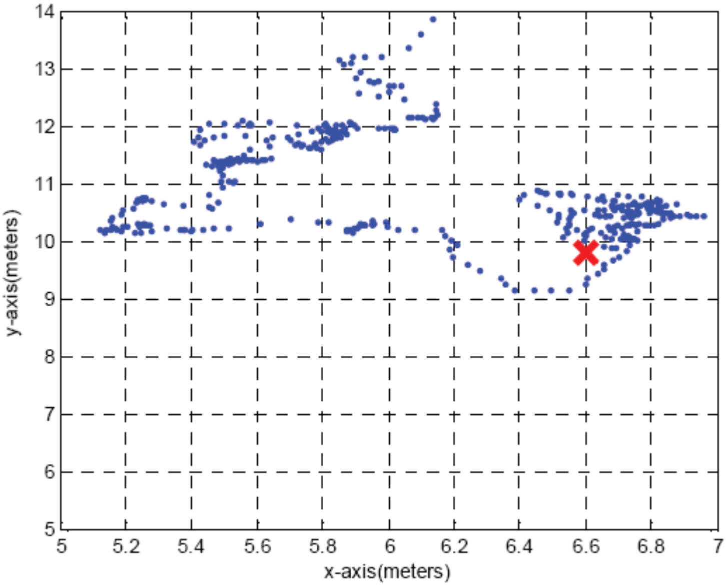


Fig. 7-(c). Experimental test results inside the sensor network.

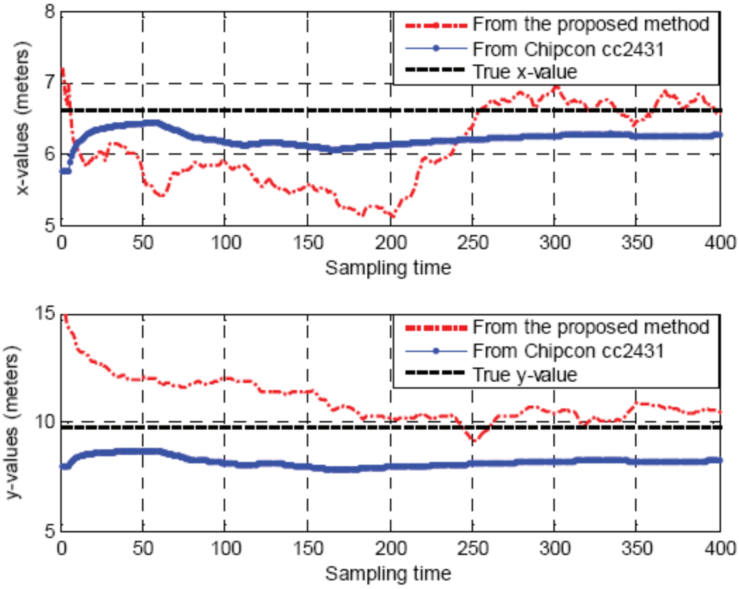


Fig. 7-(d). Experimental test results inside the sensor network.

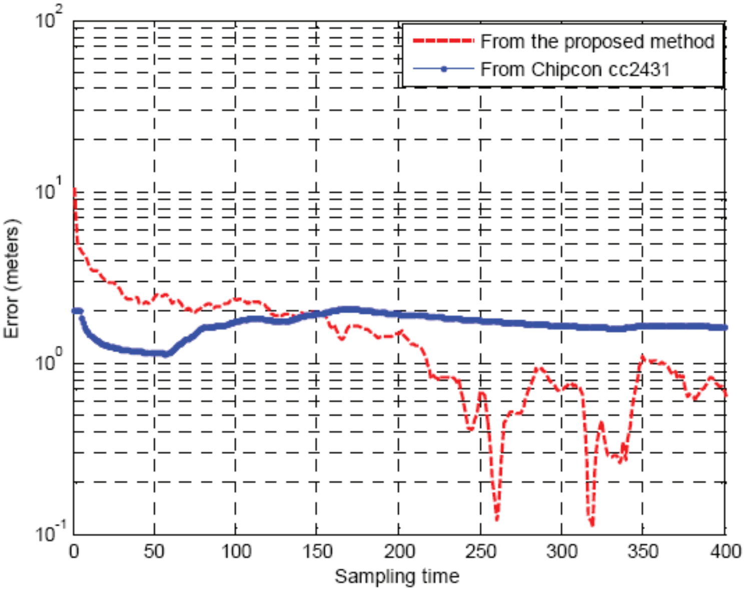


Fig. 7-(e). Experimental test results inside the sensor network.

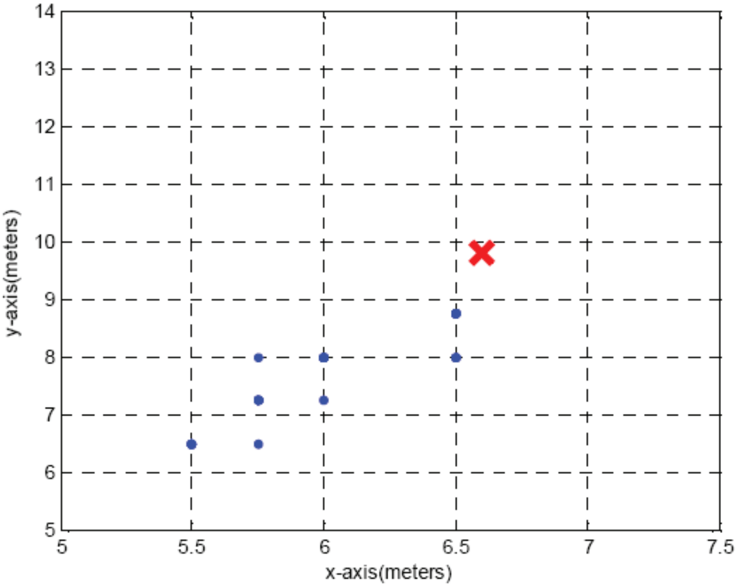


Fig. 7-(f). Experimental test results inside the sensor network.

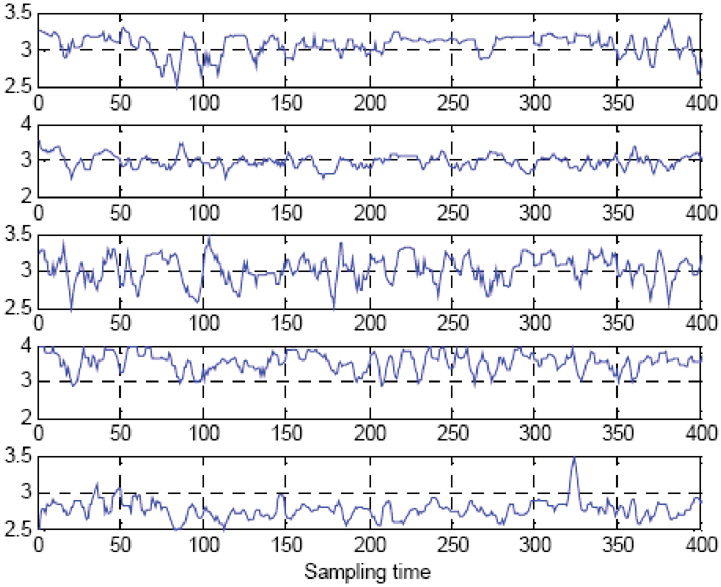


Fig. 8-(a). Experimental test results near the wall.

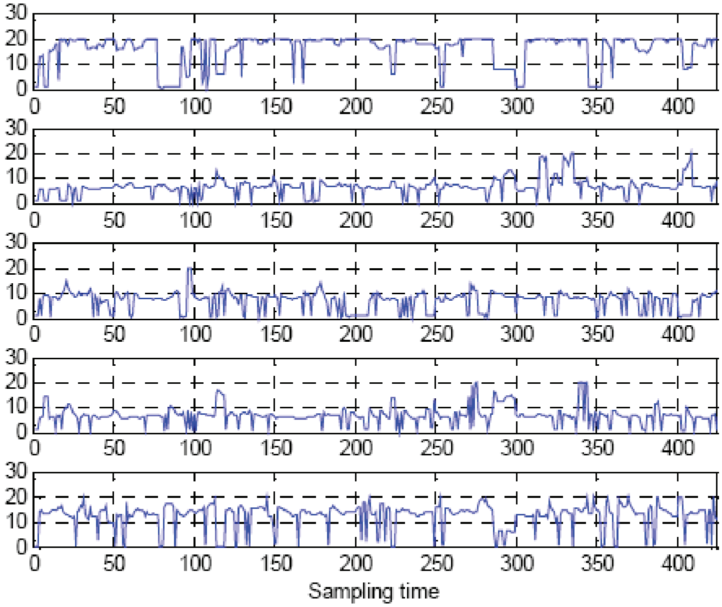


Fig. 8-(b). Experimental test results near the wall.

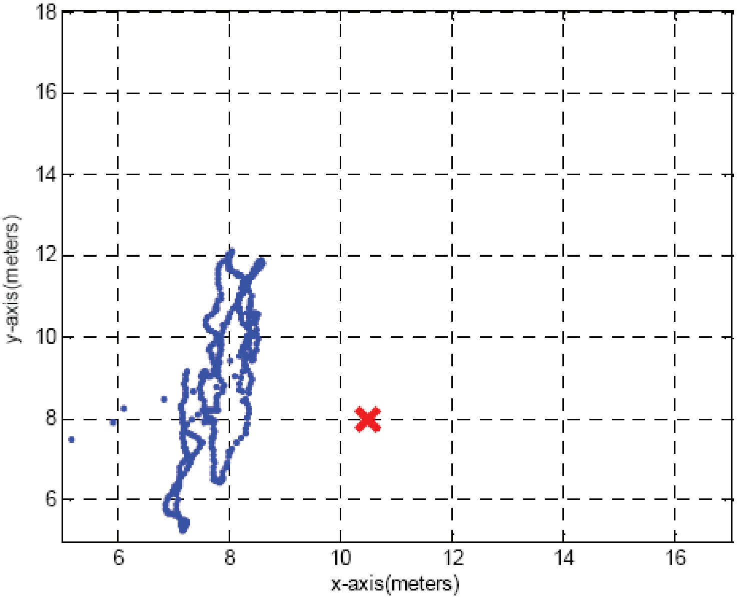


Fig. 8-(c). Experimental test results near the wall.

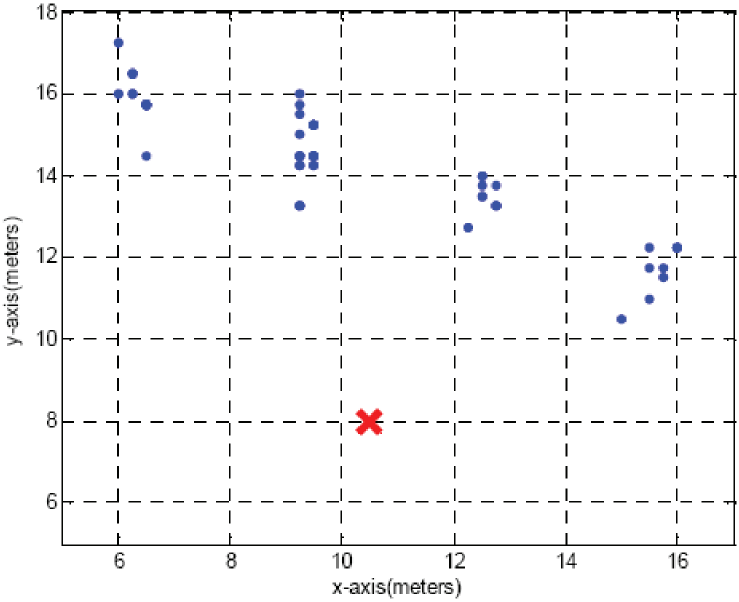


Fig. 8-(d). Experimental test results near the wall.

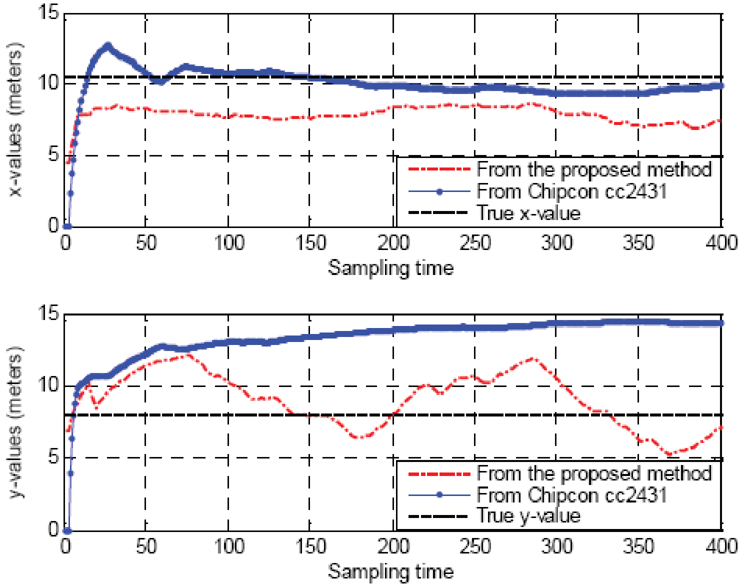


Fig. 8-(e). Experimental test results near the wall.

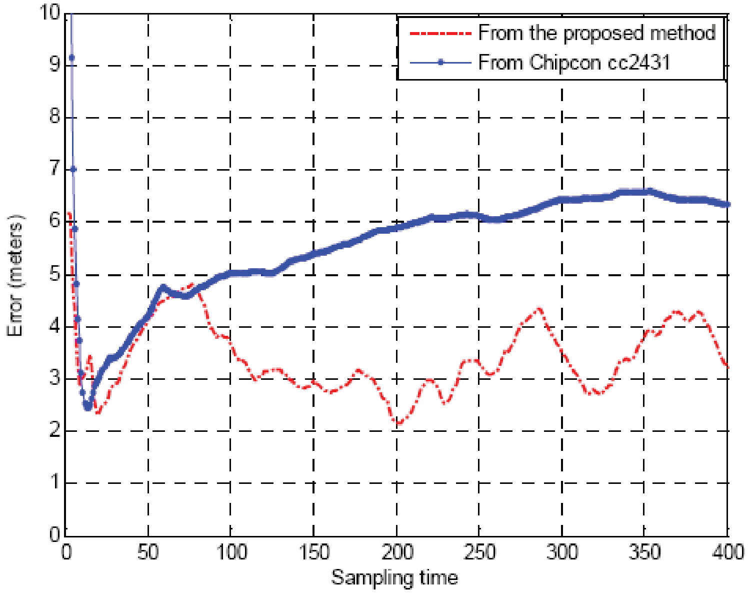


Fig. 8-(f). Experimental test results near the wall.

Table 2 and Table 3 show the test results. The true position is the actual place where the tag was placed. In these tables, MSE stands for mean square error given in Equation (14). From these tables, we observe that the new method has much better performance than cc2431 in Tests 1, 2, 3, and 4; however cc2431 is slightly better than the new method in Test 5. Thus, we can see that the new method is dominantly better (four times better out of five tests; 80 percents better) than the commercial cc2431. However, we also note that the second method estimates the position of the tag much faster than the first method. The measured and estimated position values given in Table 2 and Table 3 were sampled at every second, which can be considered as real time estimation.

7. Conclusions

In this chapter, we presented a set of classifications of indoor localization techniques. We generated categories according to measurement attribute, location algorithms, and communication protocols. The classifications presented in this chapter provide a compact form of overview on WSN-based indoor localizations. Then, based on the classifications, we introduced server-based and range-based localization systems that can be used for the indoor service robot. Specifically, we presented UWB, Wi-Fi, ZigBee, and CSS-based localization systems.

From actual experimental tests, however we found that the existing WSN-based methods have their own disadvantage. That is, Ubisense system is expensive and needs heavy hardware equipment. The Wi-Fi system (Ekahau) has a low accuracy and is only useful for the room-level localization. The CSS-based system is too expensive. Thus, this chapter introduced a localization method based on received signal strength index (RSSI).

	True position		cc2431			New method		
	x	y	x	y	MSE	x	y	MSE
Test1	7.80	5.40	5.50	9.50	4.70	5..93	3.82	2.44
	7.80	5.40	5.50	9.75	4.92	5.87	3.80	2.51
	7.80	5.40	5.50	9.75	4.92	5.87	3.80	2.51
	7.80	5.40	5.50	9.75	4.92	5.87	3.80	2.51
	7.80	5.40	5.50	9.50	4.70	5.83	3.80	2.54
	7.80	5.40	5.50	9.50	4.70	5.83	3.87	2.49
	7.80	5.40	5.50	9.50	4.70	5.83	3.80	2.54
	7.80	5.40	5.50	9.50	4.70	5.90	3.79	2.49
	7.80	5.40	5.50	9.50	4.70	5.83	3.87	2.49
	7.80	5.40	5.50	9.50	4.70	5.90	3.84	2.46
Test2	5.40	5.40	9.50	8.00	4.85	6.28	3.25	2.32
	5.40	5.40	9.50	8.50	5.14	6.41	3.11	2.50
	5.40	5.40	9.50	8.50	5.14	6.41	3.11	2.50
	5.40	5.40	9.50	8.50	5.14	6.41	3.03	2.58
	5.40	5.40	9.50	8.50	5.14	6.41	3.03	2.58
	5.40	5.40	9.50	8.50	5.14	6.41	3.03	2.58
	5.40	5.40	9.50	8.50	5.14	6.41	3.03	2.58
	5.40	5.40	9.50	8.50	5.14	6.41	3.03	2.58
	5.40	5.40	9.50	8.50	5.14	6.27	3.15	2.42
	5.40	5.40	9.50	8.50	5.14	6.27	3.15	2.42
Test3	5.40	5.40	13.50	14.75	11.47	8.25	3.29	1.17
	9.00	4.20	13.50	14.75	11.47	8.25	3.29	1.17
	9.00	4.20	13.50	14.50	11.24	8.56	3.25	1.05
	9.00	4.20	13.50	14.50	11.24	8.56	3.25	1.05
	9.00	4.20	13.50	14.50	11.24	8.25	3.29	1.17
	9.00	4.20	13.25	14.75	11.37	8.25	3.29	1.17
	9.00	4.20	13.50	14.50	11.24	8.25	3.29	1.17
	9.00	4.20	13.50	14.75	11.47	8.56	2.94	1.34
	9.00	4.20	13.25	14.50	11.14	8.25	3.01	1.41
	9.00	4.20	13.25	14.25	10.91	8.25	3.29	1.17

Table 2. Comparison of performance between cc2431 and new method

The algorithms introduced in this chapter update the signal attenuation parameter in real time and calculate the distances between reference nodes and mobile tag. The algorithms have been implemented in ubiquitous ZigBee (2.4 GHz RF communication system) sensor network. The hardware equipment required for the test was developed and tested in office environment. From the comparisons with existing localization chipset Chipcon cc2431, we found that the proposed algorithm (the first method) located the position of an object more accurately than cc2431 as time passed. The second method estimates the position of the tag very fast and accurately. The second method estimates the position much faster than the first method and estimates the position accurately; four cases out of five were better than cc2431 and one case is slightly worse than cc2431. Thus, we conclude from experimental tests that the first method is particularly useful for the position estimation of the stationary object, and the second method is practically useful for the fast and reliable position estimation of slowly moving object.

	True position		cc2431			New method		
	x	y	x	y	MSE	x	y	MSE
Test4	5.20	7.80	55.50	62.00	73.94	-0.09	5.33	5.84
	5.20	7.80	55.50	62.00	73.94	-0.09	5.20	5.90
	5.20	7.80	55.50	62.00	73.94	1.63	4.94	4.58
	5.20	7.80	55.50	62.00	73.94	1.63	4.94	4.58
	5.20	7.80	55.50	62.00	73.94	1.17	5.13	4.83
	5.20	7.80	55.50	62.00	73.94	0.08	5.21	5.65
	5.20	7.80	55.50	62.00	73.94	1.17	5.00	4.90
	5.20	7.80	55.75	62.00	74.11	1.68	5.00	4.50
	5.20	7.80	55.75	62.00	74.11	1.68	5.00	4.50
	5.20	7.80	55.50	62.00	73.94	2.47	4.89	3.99
Test5	5.40	5.40	11.75	7.50	3.67	7.76	1.35	4.69
	8.40	6.00	12.50	8.00	4.56	7.70	1.43	4.62
	8.40	6.00	11.75	8.00	3.90	7.70	1.43	4.62
	8.40	6.00	11.75	8.00	3.90	7.76	1.42	4.62
	8.40	6.00	12.50	8.00	4.56	7.70	1.36	4.69
	8.40	6.00	12.50	7.50	4.37	7.70	1.36	4.69
	8.40	6.00	12.50	8.00	4.56	7.51	1.58	4.51
	8.40	6.00	11.75	7.50	3.67	7.29	1.84	4.31
	8.40	6.00	11.00	7.25	2.88	7.51	1.58	4.51
	8.40	6.00	11.75	7.50	3.67	7.51	1.58	4.51

Table 3. Comparison of performance between cc2431 and new method (cont.)

Note that since the methods introduced in this chapter are RSSI-based method, the system is very simple and the implementation cost is much cheaper than TOA and TDOA-based methods, such as Ubisense systems and CSS systems. For a more comprehensive overview and experimental test results of WSN-based localization systems, it is recommended to refer to (Ahn & Yu, 2006; Ahn et al, 2007[1]; Ahn et al, 2007[2]; Ahn et al, 2008[1]; Ahn & Yu, 2008[2]; Ahn & Yu, 2008[3]; Ahn & Yu, 2008[4]; Hur & Ahn, 2008).

8. Acknowledgement

The work of this chapter was supported in part by the IT R&D program of Korea MIC (Ministry of Information and Communication) and IITA (Institute for Information Technology Advancement) [2005-S-092-02, USN-based Ubiquitous Robotic Space Technology Development], in part by the financial support from Korea Science and Engineering Foundation [KOSEF, Project No. R01-2008-000-10031-0], and in part by a grant from the institute of Medical System Engineering (iMSE) in the GIST of Korea.

9. References

Ahn, H.-S. & Yu, W. (2006). Wireless localization network for a ubiquitous robotic space: Background and concept, *Proceedings of the 3rd Int. Conf. on Ubiquitous Robots and Ambient Intelligence*, pp. 187-192, Seoul, Korea, Nov. 2006

Ahn, H.-S. ; Yu, W. & Lee, J.-Y. (2007)[1]. Wireless localization network for ubiquitous robotic space: Approaches and experimental test, *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pp. 1-6, Jeju, Korea, Aug. 2007

Ahn, H.-S.; Lee, J.-Y.; Yu, W. & Han, K.-S. (2007)[2]. Indoor localization technique for intelligent robotic space (written in Korean). *ETRI's New Technologies,* Vol. 22, No., (2007), pp. 48-57

Ahn, H.-S. ; Hur, H. & Choi, W.-S. (2008)[1]. One-way ranging without time synchronization for CSS-based indoor localization, *Proceedings of the IEEE Int. Conf. on Industrial Informatics*, pp. 1-6, Daejeon, Korea, July 2008

Ahn, H.-S. & Yu, W. (2008)[2]. Wireless localization network for indoor service robots, Submitted to *the IEEE/ASME Int. Conf. Mechatronics and Embedded Systems and Applications,* pp. 1 -6, Beijing, China, Oct. 2008

Ahn, H.-S. & Yu, W. (2008)[3]. Reference tag-based indoor localization technique, *Proceedings of the IFAC World Congress,* pp. 1-6, Seoul, Korea, July 2008

Ahn, H.-S. & Yu, W. (2008)[4]. Environmental-adaptive RSSI-based indoor localization Submitted to *IEEE Trans. on Automation Science and Engineering*, 2008

Bhargava, V. & Sichitiu, M. L. (2005). Physical authentication through localization in wireless local area networks, *Proceedings of the IEEE Global Telecommunications Conference*, pp. 2658-2662, 2005

Bocquet, M.; Loyez, C. & Benlarbi-Delai, A. (2005). Using enhanced-TDOA measurement for indoor positioning. *IEEE Microwave and Wireless Components Letters,* Vol. 15, No. 10, (2005), pp. 612-614

Chae, H. ; Lee, Y. ; Yu, W. & Doh, N. L. (2005). StarLITE: a new artificial landmark for the navigation of mobile robots, *Proceedings of the 1st Japan-Korea Joint Symposium on Network Robot Systems*, pp. 1-5, Kyoto, Japan

Crepaldi, M. (2005). *Analysis, design and simulation of an UWB receiver for indoor localization* (Ph.D. thesis), Politecnico Di Torino

Elnahrawy, E.; Li, X. & Martin, R.-P. (2004). Using area-based presentations and metrics for localization systems in wireless LANs, *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pp. 650-657, 2004

Fontana, R.-J., Richley, E., & Barney, J. (2003). Commercialization of an ultra wideband precision asset location system, *Proceedings of the 2003 IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 369-373, 2003

Gezici, S.; Tian, Z.; Giannakis, G. B.; Kobayashi, H.; Molisch, A. F., Vincent, H. & Sahinoglu, Z. (2005). Localization via ultra-wideband radios. *IEEE Signal Processing Magazine,* Vol. 22, No. 4, pp. 70-84, 2005

Gifford, S. (2005). Experiences with location sensing systems at the University of Michigan, *Proceedings of the 2005 NSF CISE/CNS Infrastructure Experience Workshops*, pp. 1-5, Urbana, Illinois, 2005

Guoping, Z. & Rao, S. V. (2005). Position localization with impulse ultra wide band, *Proceedings of the IEEE/ACES International Conference on Wireless Communications and Applied Computational Electromagnetics*, pp. 17-22, Honolulu, Hawaii, 2005

Gustafsson, F. & Gunnarsson, F. (2005). Mobile positioning using wireless networks. *IEEE Signal Processing Magazine,* Vol. 22, No. 4, (2005), pp. 41-53

Hatami, A. & Pahlavan, K. (2006). Performance comparison of RSS and TOA indoor geolocation based on UWB measurement of channel characteristics, *Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1-6, Helsinki, Finland

He, T.; Huang, C.; Blum, B. M.; Stankovic, J. A. & Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks, *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 81-95, San Diego, CA, 2003

Hur, H. & Ahn, H.-S. Hybrid-style wireless localization network for indoor mobile robot applications, Accepted by *the US-Korea Conf. on Science, Technology, and Entrepreneurship*, pp. 1-5, San Diego, CA, Aug. 2008

Hightower, J. & Borriello, G. (2001). Location systems for ubiquitous computing. *Computer,* Vol. 34, No. 8 (2001), pp. 57-66.

Ingram, S. J. ; Harmer, D. & Quinlan, M. (2004). Ultrawideband indoor positioning systems and their use in emergencies, *Proceedings of the 2004 Position Location and Navigation Symposium*, pp. 706-715, 2004.

Irahhauten, Z.; Nikookar, H. & Janssen, G. J. M. (2004). An overview of ultra wide band indoor channel measurements and modeling. *IEEE Microwave and Wireless Components Letters,* Vol. 14, No. 6, (2004), pp. 386-388

Ladd, A. M.; Bekris, K. E.; Rudys, A. P.; Wallach, D. S. & Kavraki, L. E. (2004). On the feasibility of using wireless Ethernet for indoor localization. *Name IEEE Trans. Robotics and Automation,* Vol. 20, No. 3, (2004), pp. 555-559

Larson, L. ; Laney, D. & Jamp, J. (2003). An overview of hardware requirements for {UWB} systems: interference issues and transceiver design implications, *Proceedings of the IEEE Military Communications Conference*, pp. 863-867, Boston, MA, 2003

Li, B.; Wang, Y.; Lee, H. K., Dempster, A. & Rizos, C. (2005). Method for yielding a database of location fingerprintings in WLAN. *IEE Proceedings-Communications,* Vol. 152, No., (2005), pp. 580-586

Lin, Y.-H.; Jan, I-C.; Ko, P.~C.; Chen, Y.-Y.; Wong, J.-M. & Jan, G.-J. (2004). A wireless PDA-based physiological monitoring system for patient transport. *IEEE Trans. Inf. Tech. In Biomedicine,* Vol. 8, No. 4, (2004), pp. 439-447

Oppermann, I.; Hamalainen, M. & Iinatti, J. (2004). *UWB Theory and Applications,* John Wiley & Sons, West Sussex, England

Oppermann, I.; Stoica, L.; Rabbachin, A.; Shelby, Z. & Haapola, J. (2004). UWB wireless sensor networks: UWEN-A practical examples. *IEEE Communications Magazine,* Vol. 42, No., (2004), pp. S27-S32.

Patwari, N.; Ash, J. N.; Kyperountas, S.; Hero, A. O.; Moses, R. L.; & Correal, N. S. (2005). Locating the nodes. *IEEE Signal Processing Magazine,* Vol. 22, No. 4, (2005), pp. 54-69

Peyrard, F.; Soutou, C. & Mercier, J.-J. (2000). Mobile stations localization in a WLAN, *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, pp. 136-142, Tampa, FL, 2000.

Ping, S. (2003). Delay measurement time synchronization for wireless sensor networks, Technical Report, IRB-TR-03-013, Intel Research Berkeley Lab, June 2003

Reed, J. H. (2005). *An Introduction to Ultra Wideband Communication Systems,* Prentice Hall, Upper Saddle River, NJ

Rehim, M. A. A. A. Y. A. (2004). *HORUS: A WLAN-based indoor location determination system* (Ph.D. thesis), University of Maryland

Ruiz, B. Q.; Vazquez, A. A.; Rubio, M. L. & Gareia, J. L. (2005). Impulse radio {UWB} system architecture for smart wireless sensor networks, *Proceedings of the 2nd International Workshop on Networking with Ultra Wide Band and Ultra Wide Band for Sensor Networks*, pp. 35-39, Rome, Italy, 2005

Sayed, A. H.; Tarighat, A. & Khajehnouri, N. (2005). Networked-based wireless location. *IEEE Signal Processing Magazine,* Vol. 22, No. 4, (2005), pp. 24-40

Ssu, K.-F.; Ou, C.-H., & Jiau, H.-C. (2005). Localization with mobile anchor points in wireless sensor networks. *IEEE Trans. Vehicular Tech.,* Vol. 54, No. 3 (2005), pp. 1187-1197

Sun, G.; Chen, J.; & Guo, W. (2005). Signal processing techniques in network-aided positioning. *IEEE Signal Processing Magazine,* Vol. 22, No. 4, (2005), pp. 12-23

Xiang, Z.; Song, S.; Chen, J.; Wang, H.; Huang, J. & Gao, X. (2004). A wireless LAN-based indoor positioning technology. *IBM J. Res. & Dev,* Vol. 48, No. 5, (2004)

Yu, K., Montillet, J.-P., & Rabbachin, A. (2006). UWB location and tracking for wireless embedded networks. *Signal Processing,* Vol. 86, No., (2006), pp. 2153-2171

Yu, W. ; Chae, H. ; Lee, J.-Y. ; Doh, N. L. & Cho, Y.-J. (2006). Robot localization network for development of ubiquitous robotic space, *Proceedings of the 2006 International Symposium on Flexible Automation*, Osaka Japan, 2006

Yu, W.; Lee, J.-Y.; Ahn, H.-S.; Ha, Y.-G.; Jang, M.; Sohn, J.-C. & Kwon, Y.-M (2008). Design and implementation of a ubiquitous robotic space using wireless sensor network and IT infrastructure. Submitted to *IEEE Trans. on Automation Science and Engineering*

Zhang, M.; Zhang, S.; Cao, J. & Mei, H. (2006). A novel indoor localization method based on received signal strength using discrete Fourier transform, *Proceedings of the First International Conference on Communications and Networking in China*, pp. 1-5, Beijing, China

14

Multi-criteria Optimal Design of Cable Driven Ankle Rehabilitation Robot

P. K. Jamwal, S. Q. Xie, K. C. Aw and Y. H. Tsoi
University of Auckland
New Zealand

1. Introduction

An ankle rehabilitation robot has been conceptualized and designed to realize the range of motion, muscle strengthening and proprioception training exercises for ankle joint. The robotic device is intended to help patients and therapists in their cooperative efforts for the treatment of impaired ankle joint as a result of injury or stroke. After analyzing the ankle joint anatomy and its motions, a parallel mechanism is proposed for the robot. To mimic the human ankle joint and its muscle actuation, the proposed robot uses artificial air muscle configured in a fashion close to the actual muscle arrangement. The apparent advantages of the proposed robot over the existing ankle rehabilitation parallel mechanisms have been emphasized. As a matter of fact, the performance of parallel robots greatly depends on their dimensions and the configuration of their actuators. Thus to explore the potential of these robots, it is essential to obtain a set of kinematic parameters, leading to optimal robot performance. To achieve this, robot designs need to be optimized on the basis of performance parameters such as, workspace, condition number and Euclidean 2-norm of actuator forces, under various operational constraints. The performance criteria and the constraints are discussed in detail to justify their influences on the robot design. The existing Multi-objective Optimization Approaches (MOA) e.g. weighted formula approach, population based approach and Pareto optimal approach have been discussed. The algorithm used in this chapter is based on genetic algorithms and attempts to draw advantages of the weighted formula and the Pareto optimal approaches simultaneously for the optimization of robot design. The results obtained from the optimization are discussed and important inferences for further work are drawn.

2. Rehabilitation and Robotics

Rehabilitation in a broader sense means a practice by which any form and grade of human physical disorder can be reinstated. The disorder could be the result of an injury or a stroke. Conventionally, to restore range of motions and strength of limbs, rigorous and repetitive exercises are performed under the supervision of a therapist. These exercises over the time improve motor functions by enhancing neuro-plasticity and neuro-recovery at the affected limbs. Apparently during a rehabilitation treatment, cooperative efforts of therapist and patient are required over prolonged sessions of treatment in a clinic. Moreover the patient is required to continue the prescribed exercises at home for a speedy recovery. It has been documented (Krebs et al., 2003) that using conventional way of treatment the recuperation is slow and sometimes continues for more than a year. The patient, the therapist and the rehabilitation process suffer from the drawbacks of conventional treatment. Patients have to travel in their disabled state to attend the clinical sessions which is undesirable especially when they have lower limb injuries. Treatments in the rehabilitation clinic are costly and time consuming, considering the travel time and the waiting time of patients. Furthermore, exercises advised by the therapist are monotonous and lacks motivation, hence resulting in inadequate improvement. Similarly the therapist has to perform strenuous and repetitive efforts with the patients and thus he can only attend a limited number of patients in a day. Due to lack of documented history of the patient's improvement, therapists normally advise further treatment based on their own perception which adds to the undesired subjectivity. Robotics can play an important role in the process of rehabilitation by assisting the therapist and the patient. While using the robot, the patient doesn't get tired of moving his ankle, as is now being moved by the robot for the range of motion exercises. Further to make exercises more interesting and motivating certain visual and haptic effects can be appended with the robot. Using a personal computer, the therapist can establish a remote connection with the patient's robot and get the required information about his progress. Similarly the patient can also receive instructions from therapist staying at home. Rehabilitation process can also be improved by acquiring progressive data of patient's improvement, which in turn can help the therapist to make accurate decisions on the choice of further exercises. Moreover the expert knowledge of the therapist can be incorporated in the robot controller to make it adaptive to different modes of exercises.

Rehabilitation robots are different (Tejima, 2000) from industrial robots in application and operation and hence special care must be taken in their design. Human augmented robots should be especially safe to use and must be user friendly in operation. This calls for ergonomic design and intelligent and adaptive robot controllers. Thus the design and control of these robots are challenging tasks requiring multi-disciplinary skills and in-depth knowledge of human joint anatomy and movements.

There are robotic devices currently in use such as MIT-MANUS for the upper limb rehabilitation (Krebs et al., 2003), LOKOMAT for gait training (Hesse et al., 2003) and Rutgers Stewart platform and other parallel robots (Dai et al., 2004) for ankle rehabilitation. However, the potential of robotics in rehabilitation has not been completely explored and key issues such as optimal design and intelligent and adaptive control, requires further research.

This chapter provides a discussion on the complexities of the ankle joint, its rehabilitation and challenges on the optimal design and development of a new parallel rehabilitation robot. Section 3 elucidates the anatomy, problems and physiotherapy of the human ankle joint along with a brief review on existing robotic devices and their shortcomings. A new wearable parallel robot which has been conceptualized to compensate the drawbacks of previous designs is proposed in Section 4 with brief discussion on its kinematic and geometrical modeling and the workspace analysis. The important design criteria and their significance are discussed in Section 5, followed by the design optimization problem formulation in Section 6. Section 7 investigates possible approaches to solve multi-criteria and multi-variable optimization problems. Genetic algorithm (GA) has been used to implement the proposed optimization scheme and hence the GA methodology and the key steps of the proposed algorithm are also explained in Section 7. Results obtained from our proposed algorithm are discussed along with some inferences in Section 8. Conclusions drawn and the future work are presented in Section 9.

3. Human Ankle, its Problems and Physiotherapy

3.1 Ankle Complex

The ankle is a complex bony structure in the human skeleton (Dul and Johnson, 1985) and is a combination of two joints (Figure1). The first joint is called the *ankle joint* which is made up of three bones: the lower end of the tibia (shinbone), the fibula (the small bone of the lower leg) and the talus (the bone that fits into the socket formed by the tibia and the fibula). The talus sits on top of the calcaneus (the heel bone) and moves mainly in one direction. The ankle joint works like a hinge to allow foot to move up (dorsiflexion) and down (plantar flexion). The second joint is the *subtalar joint*, also known as the *talocalcaneal joint* and this is a joint of the foot. It occurs at the meeting point of the talus and the calcaneus. This joint is responsible for the inversion and eversion of the foot, but plays no role in dorsiflexion or plantarflexion of the foot. However it is very much a part of the ankle joint and thus can not be ignored.

There is one more joint called MTP (metatarsophalangeal) joint connecting fore and the rear with Calcaneus, Cuboid and Navicular bones as shown in Figure1. The raising and lowering motions of the Toe and the heel are achieved about this joint. In our study the *ankle* and the *subtalar joints* have been collectively considered as a spherical joint and are called *ankle joint* henceforth for simplicity. Since our study is limited to the ankle joint motions and not the fore foot motions, the MTP joint is not considered.

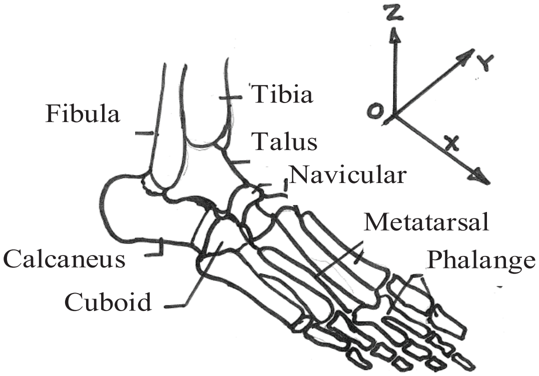


Fig. 1. Schematic of the Ankle and the sub talar joint

There are ligaments on both sides of the ankle joint that hold the bones together and many tendons cross the ankle to help move the ankle and the toes. Ligaments connect bones to bones while tendons connect muscles to bones. The ankle joint is capable of rotations in all three planes (sagittal, frontal and transverse planes), sagittal plane is defined by *x* and *z* and movements in this plane occur around the *y* axis as shown in Figure 1; transverse plane is defined by *x* and *y* and movements in this plane occur around the *z* axis; and frontal plane is defined by *y* and *z* and movements in this plane occur around the *x* axis. Various ankle movements (Siegler et al., 1988) and approximate passive moment requirements (Parenteau et al., 1998) are summarized in Table 1.

Axes	Name of the motion	Range of Motion	Torque Requirement (Nm)
X	Inversion	14.5°-22°	48
	Eversion	10°-17°	34
Y	Dorsiflexion	20°-30°	50
	Plantarflexion	37°-45°	50
Z	Adduction	22°-35°	40
	Abduction	15°-25°	40

Table 1. Approximate limiting values of range of motions and the passive moments at ankle joint .

3.2 Ankle Injuries and Physiotherapy

Ankle injuries (Dul and Johnson, 1985) are one of the most common injuries in sports and daily life. Youngsters are subjected to ankle injuries from sports and whilst carrying excessive load whereas children and the elderly gets them from walking on uneven surfaces and bone weakness. Non-functionality of ankle joint is also quite common in stroke surviving patient.

Common ankle injuries are sprain, strain and fracture. An overstretched muscle or tendon causes strain which is a mild injury. However if a ligament is overstretched it causes more serious injury called sprain which results in pain and joint non-functionality. Sometimes when a ligament is overstretched or broken it may pull off a piece of bone causing a fracture.

Primary treatment for ankle injuries (Dai et al., 2004) includes, rest, ice, compression and elevation (RICE) of the affected foot. Ice is used to reduce swelling, compression stockings are used to firmly support the ankle and foot and elevation helps to minimize further swelling. Surgical repair of the ankle ligaments is not required until the sprain in the ankle is recurrent. The primary treatment should be followed by some stretching and exercise therapy along with partial weight bearing to maintain mobility in the ankle. Achilles tendon is the *strongest tendon* of the body and is responsible for the calcaneus motion. It should be put in stretching exercise as soon as possible (within 48 to 72 hrs) after the injury to recover the range of motions (ROM). Once the ROM is achieved, strengthening of weakened muscles is essential for rapid recovery and is a preventive measure against further reinjury. As the patient achieves full weight bearing capability without pain, proprioceptive exercise is initiated for the recovery of balance and postural control using wobble boards. Finally, advanced exercises using uneven surface wobble board should be performed to regain functions specific to normal activities.

The ankle joint is an important joint in human skeleton since it is responsible to carry the body weight and maintain balance during gait. It is subjected to high impact forces which may be as high as several times of the body weight. It is a very strong joint with stiffer muscles and hence offers large resistive moments as mentioned in Table 1. In the light of above facts, it can be concluded that a wearable robotic device for ankle rehabilitation should have high stiffness and higher payload capacity to realize required passive and resistive moments at the ankle joint. Moreover the robot should be light in weight so that the patient can comfortably wear it on his leg.

3.3 Ankle Rehabilitation Robots

For ankle rehabilitation, typical devices such as elastic bands, wobbles boards and foam rollers are in use, but they allow only simple and functional rehabilitation exercises. Commercially available rehabilitation units such as ARTU, Biodex and Pro-fitter can be used but they are expensive, function specific and are not versatile. Conversely, a robotic device can be constructed to implement complete rehabilitation program which includes ROM, muscle strengthening and proprioception training. Physiotherapy requires painstaking repetitive movements of limbs about their respective joints and robotic machines are useful in such applications once appropriately programmed. Robotic devices can be programmed by a physiotherapist and exercise modes could be selected depending on the type of injury and patient's state of disability. Thus, task oriented training by supervised robotic systems is helpful to provide the patient with more useful exercises. Looking to the requirements for a robot specific to ankle treatments (as discussed in the previous section), it is realized that the parallel mechanism is a good choice owing to its high stiffness and payload capacity. Parallel robots normally have two platforms, a fixed platform (F.P.) and a moving platform (M.P.) connected together with rigid or flexible links or joints. In a recent development researchers have proposed some ankle rehabilitation robots based on parallel mechanisms. These designs have been studied critically to provide inputs for our proposed robot.

In one of the earliest works, (Girone et al., 1999) proposed the "Rutgers Ankle" that uses a Stewart platform which can provide six degrees of freedoms (DOF). Double acting pneumatic cylinders are used as the actuators to move the platform to perform various ankle rehabilitation modes. The patient's foot is fixed firmly to the platform and assistive or resistive forces are applied for passive and active mode of exercises respectively. This platform has further been interfaced with the game-like virtual environments (Girone et al., 2000) to make the exercises more interesting and enjoying for the patient. The Rutgers Ankle was also used in the clinical trials for post-stroke rehabilitation (Deutsch et al., 2007) apart from sprained ankle treatment. In a recent development (Deutsch et al., 2007) have proposed a remote console which is a telerehabilitation system providing real time interaction between the patient, the robot and the therapist sitting remotely. Though the Rutgers Ankle is well developed and is being used in the clinical trials, the mechanism has not been thoroughly analyzed from the point of view of its workspace optimization and stiffness analysis. Moreover the patient's leg also contributes to the overall dynamics of the mechanism since it is not constrained and this fact has not been highlighted. The position of the ankle joint in the robot does not remain constant and there is a possibility of small shift in the ankle joint location causing inconvenience to the patient and inaccurate control. Since the ankle movements in most exercises require less than four DOF motions, (Dai et al., 2004) proposed a parallel robot for sprained ankle treatments using a three and four DOF parallel mechanism with a central strut. Kinematic and stiffness analysis has been carried out for the proposed mechanism. In particular they have used a central strut and analyzed three different types of parallel robots in the domains of stiffness. A single platform-based reconfigurable robot mechanism has been proposed by (Yoon et al., 2006). This robot design considers the MTP joint apart from the ankle joint motions and has less than six DOF motions. Since it is a reconfigurable robot, the same platform can be used for ROM, muscle strengthening and proprioception training. An impedance controller is proposed for this robot wherein the impedance parameters can be varied to accommodate different exercise modes. A 3-RSS/S parallel mechanism is proposed by (Liu et al., 2006) and the kinematic design of prototype has been validated using simulations.

So far most platform type devices require patient’s foot to be placed on top of the platform, actuated from the bottom. There are two major issues with such a configuration, firstly, when the platform and the foot fixed to the platform are moved the position of the ankle joint keeps changing with respect to the ground. This inaccuracy in the position of the ankle joint leads to control errors which are difficult to comprehend. Secondly, the existing platforms have translational motion along with the rotation which causes shift in the patient’s leg. Hence the dynamic model of the robot should include dynamic inertia of the patient’s leg which can not be estimated. In the absence of an accurate dynamic model, large trajectory errors are expected. In (Girone et al., 2000) authors have proposed an Inside Track 3D tracker to measure the position and the orientation of patient’s shinbone to avoid trajectory errors and prevent ankle movements beyond specified ROM. However movements of the shinbone relative to the ankle joint definitely causes discomfort to the patient as he is required to change his position intermittently.

To compensate these issues we have considered a new configuration of actuators for our proposed design which is similar to the actual muscular system in the human leg. The actuation of the end platform is performed using air muscles connected parallel to the patient’s leg in the robot. Thus in our proposed robot the position of ankle joint and the leg remains stationary when the foot is moved in different exercise modes.

4. Proposed Soft Parallel Robot for Ankle Rehabilitation

4.1 Soft Parallel Manipulators

The parallel mechanisms can be classified as Rigid Parallel Mechanisms (RPM), if the links connecting the two platforms are rigid bodies and Soft Parallel Mechanisms (SPM) when the links are tendons or cables. There are certain problems using conventional RPM for the proposed wearable design of the robot. First of all, for a wearable robot the weight of the actuators should be kept low so that the patient can comfortably move his leg around with the robot. The RPM’s use linear actuators which are heavy and rigid and hence can not be used in the proposed robot. Secondly the RPM uses spherical joints which results in the reduced ROM of the robot.

Soft parallel devices are very light in weight and have higher payload to weight ratio. Air muscle actuators along with the cable in the proposed robot weigh only 85gm for each link which is very low compared to the conventional linear actuators which weigh approximately 2500gm. SPM has simpler dynamic model than their rigid-link counterparts because the inertias of their links (i.e., cables) can be ignored. Soft parallel robots are being used recently (Alp and Agrawal, 2002) in a variety of applications ranging from sophisticated medical and manufacturing applications to simple construction and shipment activities. However, wire flexibility of SPM poses some constraints on the workspace and the robot controllability and an extra variable called 'tensionability' is required to be considered during its kinematic design. Thus the design of SPM's is more critical and it is essential to select design parameters carefully to achieve controllability.

4.2 Proposed Wearable Ankle Rehabilitation Robot

This chapter proposes a wearable air muscle actuated SPM for ankle joint rehabilitation treatments. The robot is designed to provide three rotational degrees of freedom to the ankle joint. The device uses two parallel platforms, a fixed platform (FP) built-in with a leg support structure and a moving platform (MP). Air muscles are used as the actuators and are mounted on the leg support with their actuating end connected to MP through cables. These cables pass through sleeves provided in the FP. The leg support structure is light in the weight and can be fixed to the patient's leg using straps moving over the knee and fixed at thigh. MP remains below the foot and has a heel locator and straps to locate and fix foot. The moving platform of the robot is shaped to form a shoe of varying size and shape. This MP or the shoe is attached to the leg support using a special mechanism which provides three rotational degrees of freedom to the shoe and has its center of rotation coincident with the ankle joint. As the air muscles can only pull and can not push hence to maintain the tension in all the cables it is desired to have redundant actuation. In fact all the cable based parallel robots have redundant actuation (Pusey et al., 2003) which means the robot needs '(n+1)' actuators to achieve 'n-dof' motion of the manipulator. Hence to obtain 3-dof from the robot we have used a set of four air muscles and four cables. Coordinated and antagonistic actuations of air muscles will ensure desired changes in the wire lengths and pose of the moving platform subsequently for a range of ankle exercises.

The proposed robot is required to perform specified motions in 3-D space. To accomplish this, the end effector is moved in the workspace along a predefined trajectory. The position and the orientation of the robot end effector at a specified workspace location can be obtained by controlling the displacements of the cables. This calls for a mathematical model of the robot which can define the relationship between end effector motions and the cable displacements in the domains of time and space. This mathematical model is called kinematic model and the derivatives of this model describes mechanics of the motion without taking forces into account. The forces and/or the torques are considered with the mechanics of motion in the dynamic analysis. The kinematic and the dynamic studies are essential tools for the design of proposed robot. Kinematic model describes the position and orientation of the end effector with respect to the FP and the dynamic model relates the applied forces and/or torques to the resulting robot motion. The kinematic parameters of joints are of two types, fixed parameters and the variable parameters. Fixed parameters are the location of the connection points on the two platforms and the variable parameter is the length of links between the platforms. The kinematic model can be completely defined by providing the information about both types of kinematic parameters of the links or cables. The kinematic and the geometrical modeling of the proposed robot are described in the following sections.

4.3 Robot Kinematic Modeling

The kinematic model describes the spatial motion of the end effector in time domain, about a fixed reference frame (or fixed platform). The kinematic study includes two types of models, namely the Forward Kinematic (FK) model and the Inverse Kinematic (IK) model. The forward kinematic model provides position and orientation of the end effector as a function of variable and constant kinematic parameters of all the links. Similarly the inverse kinematic model helps to find the set of joint parameters that would bring the end effector in a desired location in the workspace. The desired task of the end effector is specified in terms of its position and orientation in the workspace. The joint variables, to accomplish this task are found using IK analysis. Joint variables, in turn are used to find the instantaneous coordinates of the end effector using FK analysis.

It has been well established (Innocenti and Parenti, 1990) that for parallel robots the IK solution exists in close form but the FK solution is not possible in close form. While doing a FK analysis for parallel robots one ends up with a set of highly coupled nonlinear equations for which unique solution is not possible. Both types of kinematic models are normally required to study manipulator differential motion, its statics and to implement a desired control scheme for the end effector. Kinematic analysis is also essential to estimate the feasible workspace of the robot and to perform singularity analysis. A brief discussion on the kinematic modelling of the proposed robot is presented in the following section.

4.4 Inverse Kinematic Analysis

The inverse kinematics of our proposed cable driven robot is relatively simple and provides a unique solution of cable lengths for given end effector pose. In the following discussion the wire lengths have been determined in terms of the pose of the moving platform.

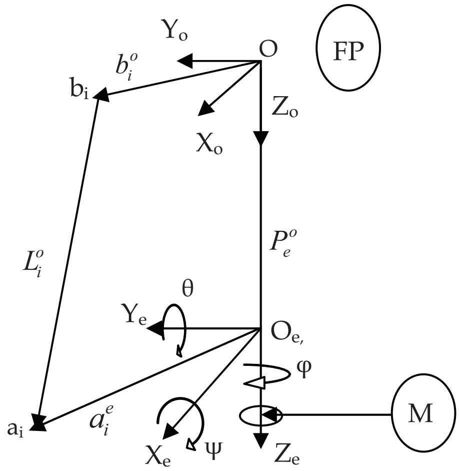


Fig. 2. A sketch of cable and position vectors of connection points on FP and MP

The diagram presented in Figure 2 shows the position vectors of the cables in the proposed SPM. The connection points on MP and FP are denoted by a_i's and b_i's respectively. The connection points on the fixed platform are all in the same plane ($Z_O = 0$) and are placed at a radial distance 'r_b' from the coordinate system which is located at O. The position vectors (b_i^o) of point b_i's on the FP are defined by

$$b_i^o = \begin{bmatrix} r_b \cos \beta_i \\ r_b \sin \beta_i \\ 0 \end{bmatrix} \tag{1}$$

where, r_b is the radial distance from the base coordinate frame O. The variable β_i denotes the angular position of point b_i on FP with respect to their respective axes. The moving platform similarly has a set of connection points located on the circumference of a circle of radius r_a and the coordinate frame attached to O_e is about 60mm above the center of mass (M_C) of the MP (with reference to the position of ankle joint which is approximately 60mm above moving plate level). The position vectors (a_i^e) of the four connection points on the moving platform can be given as follows:

$$a_i^e = \begin{bmatrix} r_a \cos \alpha_i \\ r_a \sin \alpha_i \\ -60 \end{bmatrix}. \tag{2}$$

The variable α_i is the angular position of point a_i on the MP with respect to their respective axes. The position vectors of the cable lengths in terms of end poses can be expressed as a system of four equations described below:

$$L_i^o = P_e^o + R_e^o \, a_i^e - b_i^o \quad i = 1,\dots 4. \tag{3}$$

where P_e^o represents the position vector of point O_e with respect to O. R_e^o is the rotational transformation matrix of MP with respect to FP using a fixed axis rotation sequence of ψ, θ and ϕ about X_o, Y_o and Z_o axes, respectively, and can be written as below.

$$R_e^o = \begin{bmatrix} \cos\phi\cos\theta & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \sin\phi\cos\theta & \cos\theta\cos\psi + \sin\phi\sin\theta\sin\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix} \tag{4}$$

4.5 Forward Kinematics

Forward kinematic (FK) mapping for parallel manipulators is difficult compared to serial manipulators as it involves highly coupled nonlinear equations and their closed-form solution is not possible. However it is important to use forward kinematic solution since it is a key element in closed loop position and force control of parallel manipulators. It is also an essential block in the trajectory control of a manipulator.

Quite a few approaches are found in the literature to solve FK of parallel manipulators and a few representative works have been studied during present work. In one of the earlier works (Innocenti and Parenti, 1990) the closed form solution of FK has been simplified by merging the connection points at the two platforms which reduces the degree of polynomial representing FK. Generally in practical control and other applications, a unique solution of FK is required and various numerical approaches (Deshmukh and Michael 1990) to solve non-linear equations can be used for this purpose. Several researchers have been able to linearize few of the non linear equations obtained from the kinematic analysis or have been able to reduce the degree of the set of polynomial equations (Nam and Park 2004). The system of equations of reduced order has further been solved using one of the numerical methods. Neural networks (NN), genetic algorithms (GA) and their variants have also been used by researchers to solve FK problem. Using inverse kinematics (IK) solution, one can create a database of end effector orientations and corresponding joint variables. This database can be used to train a NN and a weight matrix can be obtained for further predictions. A cascaded CMAC (Cerebella Model Arithmetic Computer) based NN to solve FK problem has been proposed by (Geng and Haynes 1991) and they have stated that the proposed algorithm is faster and more precise compared to popular back propagation algorithm. A floating point GA (Genetic Algorithm) using IK analysis has been proposed by (Boudreau and Turkkan 1996) to solve FK problem, formulating it as an optimization problem. A simple feed forward network has been used by (Yee and Lim 1997) and an accuracy of 0.017^0 and 0.017mm, in predicting end effector pose, has been achieved. NN tuned FK model has also been used by (Oyama et al. 2002) in visually guided hand position control. A comparison of NN with various numerical methods to solve FK has been provided by (Sadjadian and Taghirad 2005). BPNN (Back Propagation Neural Network) has been used by (Li, Zhu and Xu 2007) where they have employed PSO (Particle Swarm Optimization) to further train NN to achieve accuracy of the order of 0.001 degrees.

4.6 Geometric Modeling

The kinematic models establish the correlation between the joint displacements and the position and orientation of end effector of a robot. This correlation can only be used for the static control of manipulator in the workspace. For our proposed robot the final desired angular pose of the manipulator is important and at the same time the angular velocity by which it has traversed to reach the final location is also equally important. Thus it is essential to obtain a mapping between joint velocities and end effector velocity. This mapping can be defined by a matrix, which is called the robot Jacobian matrix. The Jacobian matrix depends on robot configuration and linearly maps the Cartesian velocity in to joint velocities. It is interesting to note that the Jacobian matrix defined for parallel robots corresponds to the inverse Jacobian of serial robots. To determine the Jacobian matrix of parallel robots, two approaches, namely geometric approach and analytical approach can be used. In the present chapter we have used a geometric approach as discussed below.

To determine geometric Jacobian matrix using robot geometry, initially a relation between cable lengths and end effector pose is formulated. For the subject robot this relation is given by Eq. (3). The magnitude of the cable lengths can be calculated for a given set of end effector orientation. The magnitude of each L_i^o vector can be given by l_i^o as shown below.

$$(l_{\mathrm{i}}^{\mathrm{o}})^2 = (L_{\mathrm{i}}^{\mathrm{o}})^T \quad L_{\mathrm{i}}^{\mathrm{o}} \cdot \tag{5}$$

Taking the time derivative of the above kinematic constraint equations we obtain the following equation.

$$2l_{\mathrm{i}}^{\mathrm{o}}\dot{l}_{\mathrm{i}}^{\mathrm{o}} = (\dot{P}_e^o + \dot{R}_e^o\, a_i^e)^T (P_e^o + R_e^o\, a_i^e - b_i^o) + (P_e^o + R_e^o\, a_i^e - b_i^o)^T (\dot{P}_e^o + \dot{R}_e^o\, a_i^e) \cdot \tag{6}$$

Using linear algebra identity, if ‘ c ’ and ‘ d ’ are column vectors, following holds true,

$$\mathrm{c}^{\mathrm{T}} d + d^T c = 2c^T d = 2cd^T \ . \tag{7}$$

This further implies that the Eq. 6 can be rewritten as below.

$$2l_{\mathrm{i}}^{\mathrm{o}}\dot{l}_{\mathrm{i}}^{\mathrm{o}} = 2(P_e^o + R_e^o a_i^e - b_i^o)^T (\dot{P}_e^o + \dot{R}_e^o a_i^e) \cdot \tag{8}$$

Since the end effector is constrained to only rotational motion and the spacing between the platforms remains constant, the time derivative of P_e^o should be zero. Setting $\dot{P}_e^o = 0$ and further simplifying Eq. 8, we get,

$$l_{\mathrm{i}}^{\mathrm{o}}\dot{l}_{\mathrm{i}}^{\mathrm{o}} = (L_i^o)^T (\dot{R}_e^o\, a_i^e) \tag{9}$$

but since

$$\dot{R}_e^o = \omega_e^o \times R_e^o \tag{10}$$

$$l_{\mathrm{i}}^{\mathrm{o}}\dot{l}_{\mathrm{i}}^{\mathrm{o}} = (L_i^o)^T (\omega_e^o \times R_e^o\, a_i^e) \tag{11}$$

where ω_e^o is the angular velocity vector of the end platform with respect to the coordinate system of the fixed platform. Further since a_i^e and a_i^o are related as shown in Eq. 12, Eq. 11 can be rewritten as Eq. 13. Rearranging the variables, Eq. 13 can be presented as Eqs. 14 & 15.

$$a_i^o = R_e^o\, a_i^e \tag{12}$$

$$l_{\mathrm{i}}^{\mathrm{o}}\dot{l}_{\mathrm{i}}^{\mathrm{o}} = (L_i^o)^T (\omega_e^o \times a_i^o) \tag{13}$$

$$\dot{l}_{\mathrm{i}}^{\mathrm{o}} = \frac{1}{l_{\mathrm{i}}^{\mathrm{o}}}[(L_i^o)^T (-a_i^o \times \omega_e^o)] \tag{14}$$

$$\dot{l}_{\mathrm{i}}^{\mathrm{o}} = \frac{1}{l_{\mathrm{i}}^{\mathrm{o}}}(a_i^o \times L_i^o)^T \omega_e^o \tag{15}$$

If $\dot{q}$ is the vector of link velocities and ‘ t ’ is the twist vector of the end platform, the Jacobian matrix $J(q)$ of the robot can be defined as,

$$\dot{q} = J(q)t \tag{16}$$

where 't' is a vector of angular velocities of the end platform and is given by following.

$$t = \begin{bmatrix} 0 \\ \omega_e^o \end{bmatrix} \tag{17}$$

Eq. 15 can be rearranged and compared with following matrix equation (Eq. 18) to find the Jacobian matrix as given by Equations 16 and 19.

$$X\dot{q} = Yt \tag{18}$$

$$J = X^{-1} \times Y\,. \tag{19}$$

Finally the Jacobian matrix of the proposed cable based robot can be written such that its ith row is given by following equation.

$$J_i = \left(a_i^o \times \frac{L_i^o}{l_i^o} \right)^T \qquad where \quad i = 1,2,...., 4\ \cdot \tag{20}$$

The robot Jacobian matrix is an important parameter and is extensively used for the kinematic and dynamic analysis of the robot. This matrix shall be further used in the workspace analysis and in the actuator force analysis in the following sections.

4.7 Workspace Analysis

The workspace of the proposed cable based robot is difficult to analyze for the two major reasons. Firstly, the translational and orientation workspaces are achieved through coupled motion of its links or cables and the both kind of workspaces can not be evaluated independently. Hence the workspace is defined simply as the space where the inverse and forward kinematic solutions exist. Secondly, for cable based robots (Pusey, 2003) or SPM's, their workspace is the space where sets of positive cable tensions also exist. In SPM's positive cable tensions are needed to constrain the moving platform all the time regardless of any external wrench. In other words, a manipulator pose belongs to the feasible workspace if there is at least one set of positive cable tensions forming a force closure.

Thus in SPM's, it is not only necessary to solve the closure equations but it is also essential to verify that equilibrium can be achieved with non-negative actuator (cable) forces. Workspace analysis of the SPM's is interesting due to the fact that it is constrained by more than one requirement and thus has attracted many researchers. (Stump and Kumar 2006) have approached the problem of evaluating the reachable workspace for a cable-driven parallel platform by using the tools of semidefinite programming (convex optimization) to obtain closed-form expressions for the boundaries of this workspace. Similarly in (Pusey et al. 2003), the design and workspace of a 6–6 cable-suspended parallel robot has been discussed and workspace volume is characterized as the set of points where the centroid of the moving platform can reach with tensions in all suspension cables at a constant orientation. The main contribution of (Pusey et al. 2003) is in establishing that for any geometry of platforms, the largest workspace volume occurs when the moving platform (MP) is the same size as the base platform (BP). The proposed device basically has four actuated links and accommodates patient's ankle joint which acts as a central strut in the parallel device. The air muscles which are the actuating links are all in their half contracted positions initially to facilitate the antagonistic actuation of moving platform. The cables connecting both platforms are given some Pre-tension (in present study it is assumed to be 50N), by adjusting the cable lengths. Force experienced by the central strut due to this pre-tension in the cables is called spine force and is denoted by F_S here. This helps in keeping all the cables tensionable in the workspace. The static force and moment balance on the MP are given as below.

$$F_S + \sum_{i=1}^{4} U_i = 0 \tag{21}$$

$$\sum_{i=1}^{4} ({}^{o}m_i \times U_i) + \sum M_{ext} = 0 \tag{22}$$

Though there are no external moments applied to the MP, due to ankle stiffness, finite moments are required to move ankle joint passively. To realize the ROM exercises air muscles works antagonistically and applies the required moments at the end effector. To find corresponding tensions in individual cable the dual relationship between kinematics and statics can be used as follows:

$$M_{ext} = J^T U \tag{23}$$

where 'U' is the vector of cable forces, J is the geometric Jacobian matrix of the robot and M_{ext} is a 3 dimensional vector containing the required moments given by Eq. (24). The rows in Eq. (24) represent the moments required to orient ankle joint about the X_o, Y_o and Z_o axes, respectively.

$$M_{ext} = \begin{bmatrix} M_{x_o} & M_{y_o} & M_{z_o} \end{bmatrix}^T \tag{24}$$

Now, to obtain the equations for the force in each of the cable, Eq. (23) is rearranged in the following manner.

$$U = \overline{J} M_{ext} \tag{25}$$

where we denote $$\overline{J} = (J^T)^{-1} . \tag{26}$$

Next, at each point within the possible workspace, the equation describing the force in each cable is used to see if tension is obtainable. The actuators have a limited stroke length and hence workspace points which are not reachable by the actuators are not considered. A Matlab program is written to search the entire workspace and check for the condition of tensionability of cables and link length constraint.

The proposed robot is redundantly actuated, i.e. to achieve three degrees of freedom four actuators are used. Further since J^T is a 3×4 matrix, its null space solution must exist and it should have one degree of freedom when J is full rank. The pretension in the cables takes care that all the cables remain under tension at all times. The resulting positive actuator forces for the flexion trajectory have been plotted in Figure 3 against different sets of manipulator poses. Apparently the four actuator forces such as T1, T2, T3 and T4 are all positive during the flexion trajectory. Similarly the actuator forces for other trajectories are also evaluated.

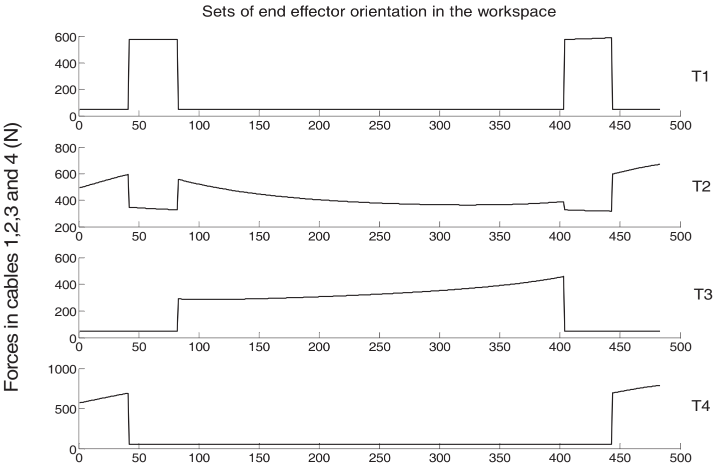


Fig. 3. Static cable forces at different end effector orientations

5. Design Criteria of the Ankle Rehabilitation Robot

In light of the unidirectional nature of cable forces, design of the proposed cable-based manipulator is more complicated than the rigid-link parallel robots. There are certain criteria, specific to the cable based parallel robots which require more attention. These design criteria are discussed below:

1. Maximum workspace criterion
2. Near unity condition number criterion
3. Singular value based criterion
4. Minimum force norm based criterion
5. Other criteria

An explanation of these measures is important to state their significance.

5.1 Workspace Criterion

Workspace is a vital parameter in the domain of kinematic analysis and workspace analysis of the proposed robot is discussed in section 4.7. The feasible workspace volume depends on the geometrical configuration of the robot such as the size of the platforms and placement of connection points on them apart from other constraints discussed in the previous section. Thus by changing the geometrical parameters it is possible to change the volume of the feasible workspace. It is desired that the workspace of a robot under given constraints should be as large as possible for greater maneuverability. Furthermore, unlike serial robots, workspace of parallel robots is unevenly shaped (Stump and Kumar, 2006) due to their complex kinematics. This further contributes in lowering the size of the feasible workspace. Apart from the size, the quality of the workspace is also important and it is desired that the workspace should be free from singularities. In our study the feasible workspace is represented by an index as given below.

$$I = \frac{\varphi_f}{\varphi_T} \tag{27}$$

Here φ_f is the number of workspace points (sets of manipulator orientation) which are reachable with the restricted stroke length of the actuators, and φ_T is the total orientation workspace of the manipulator. As mentioned before, the manipulator orientations about X$_o$, Y$_o$ and Z$_o$ axes are given by ψ, θ and ϕ respectively. The limiting values of these orientations are taken as ψ_{-30}^{+40}, θ_{-10}^{+25} and ϕ_{-20}^{+20}, considering a step size of 2°, the total orientation workspace given by Eq. 28, has 49000 points to evaluate.

$$\varphi_T = (\psi_{\max} - \psi_{\min})(\theta_{\max} - \theta_{\min})(\phi_{\max} - \phi_{\min}) \,. \tag{28}$$

In the proposed work, the robot design parameters have been optimized to achieve maximum permissible workspace volume.

5.2 Condition Number Criterion

As discussed in the previous sections, the Jacobian matrix 'J' of a robot maps joint rates to the Cartesian velocities of the manipulator. The condition number of this matrix is a measure of its sensitivity to changes in the kinematic variables of the robot. A robot design with near unity condition number is desirable (Khatami and Sassani, 2002) since it minimizes the error in the end effector wrench due to input errors in joint torques. The condition number can also be used to evaluate the workspace singularities. This number also reveals how far a robot is from its present configuration to the nearest singular configuration. Stiffness of the end effector due to joint stiffnesses can also be obtained using condition number. Thus it is evident from the above discussion that the condition number is a vital design parameter and the robot configuration should be optimally designed to produce a minimum condition number close to unity. The condition number 'k' is defined as the ratio of the largest singular value 'σ_l' to the smallest singular value 'σ_s' of the matrix 'J' for a fixed orientation of the manipulator. The singular value can be further defined as the square root of the eigenvalues of $\overline{J}\overline{J}^T$ and $\overline{J}^T\overline{J}$. The range of condition number is described as below.

$$1 \leq k < \infty \tag{29}$$

When the condition number approaches unity, the matrix 'J' is said to be well conditioned or far from singularities. On the contrary, if the condition number is higher, the matrix is said to be ill conditioned. An ill conditioned Jacobian matrix will further magnify the kinematic or dynamic error present in the robot motion. Sometimes to avoid an infinite right hand side bound, an inverted form of the condition number referred to as the conditioning index ($C.I.$) can also be used.

$$C.I. = \left(\frac{1}{k}\right) \tag{30}$$

where $0 \le C.I. \le 1$

The condition number is generally obtained at each individual point in the feasible workspace region, for a fixed orientation of the end effector. Though condition numbers at individual manipulator orientations are useful information, the Global Conditioning Number (GCN) (Khatami and Sassani, 2002) is normally used to analyze the behavior of the condition number over the entire workspace volume. In the present work robot performance has been defined by Global Condition Number (GCN) as given by,

$$GCN = \frac{\int_W (k)dw}{\int_W dw} \tag{31}$$

where 'k' is the condition number at a given orientation and 'W' is the feasible workspace. Since it is difficult to calculate the exact solution to the integrals mentioned above, GCN is discretely defined and expressed as below.

$$GCN = \frac{\sum_{i=1}^{n} (k)}{n} \tag{32}$$

Here n is the total number of discrete *feasible* points constituting the workspace and the numerator is the sum of condition numbers obtained at different points in the feasible workspace volume grid. The GCN is bounded by the range as given.

$$1 \le GCN \le \infty \tag{33}$$

Here, when the GCN is a large number the entire workspace tends to be ill conditioned and when the GCN is near one the entire workspace is said to be well conditioned. GCN further depends on the robot configuration which is defined by arrangements of connection points at both the platforms and the link lengths. Hence there exists an optimum robot configuration for a good GCN and performance thereof. To ensure that all the points in the workspace provides a condition number within certain range, the maximum value of the condition number for a particular robot design over the entire workspace can be obtained and minimized. Once the maximum GCN is minimized, it can be ensured that

1. The final GCN represents the average behavior of condition number over the feasible workspace.
2. The condition number all over the feasible workspace is always less than the maximum GCN value.

5.3 Singular Value Based Criterion

Singular values are important measure of kinematic behavior of the robot and provides an assessment on its controllability. The manipulator loses or gains extra degrees of freedom when it enters in to a singular configuration. A kinematic singularity occurs when the determinant of the Jacobian matrix becomes zero or loses rank at a particular configuration in the workspace.

$$\det(J) = 0 \tag{34}$$

Referring to Eq. (20), it is apparent that when '*J*' is singular and its null space is non-zero, there will be certain non-zero Cartesian vectors (t) resulting in zero joint vectors ($\dot{q}$). This further means that, despite the joints being locked, the end effector can still have some infinitesimal motion in a particular direction and gains one or more degrees of freedom. Hence it is essential to optimize the design parameters of the robot, in order to minimize the number of singular points in the workspace. The limiting values of certain kinematic parameters are constrained by the minimum or the maximum singular values. Thus a criteria based on singular values shall yield a desirable robot design.

5.4 Minimum Actuator Force Based Criteria

Since we intend to design a wearable robot for ankle joint rehabilitation treatment, it is desired to keep the length of the robotic device similar to the length of the patients' leg. The length of the robot is governed by the length of its actuators hence the actuators should be kept as small as possible. Further the size of these actuators depends on the cable forces calculated (Eq. 25) in Section 4.7. Longer air muscles are required for higher forces in the individual links. The lengths of the actuators can be minimized by lowering the actuator force requirements. Higher actuator forces may cause the cables to break and these forces may also produce undesired cable elongation affecting the positional accuracy adversely. Once again it is apparent that the actuator forces are the functions of robot's geometrical parameters. By selecting connection points farther from the axis of rotation, the forces can be greatly reduced. To minimize the actuators force vector it is convenient to summarize the set values of force vector as a single number. Vector norms are generally used to represent the vector in a single value. Three types of vector norms are generally used such as, 1-norm, 2-norm or ∞-norm. 2-norm or Euclidean length is preferred (Hassan and Khajepour, 2008) over the other two norms because it is more sensitive towards changes in larger force components. 1-norm is equally sensitive to all the force components and ∞-norm is only sensitive to the changes in the largest force component. In the present study, 2-norm or the Euclidean distance of the actuator forces has been considered and can be given as below:

$$\|U\|_2 = \|0 - U\|. \tag{35}$$

5.5 Miscellaneous Criteria

There are other design criteria which are specific to the application of the robot. Such criteria are size of the manipulator and the fixed platform, range of motions of the robot, material selection based on strength criteria, etc.

Despite the advantages discussed in previous sections the parallel manipulator has some innate problems owing to its closed chain mechanism. A parallel robot basically has several serial robots connected in parallel and its feasible workspace is the intersection of the individual workspace contribution of its joints. Hence the feasible workspace of our robots is small in size. Apart from the size and the shape, the workspace is also affected by the kinematic singularities. The proposed parallel robot due to its complex mechanism has three kinds of singularities such as inverse kinematic singularity, direct kinematic singularity and the combined singularities.

The design of cable based parallel robot is again more challenging since it has an added constraint on the workspace in the form of tensionability. The feasible workspace of this robot is the euclidean space where the robot manipulator can reach with positive tension in all its cables. Thus apart from the singularities the condition of tensionability is also required to be checked to evaluate the feasible workspace for the manipulator.

The proposed robot is to be used for ankle joint rehabilitation treatments and the subjects are supposed to fix their foot and the ankle into the moving platform. It is evident that the device has to be robust to take care of the physical differences in the shape and size of different patient's foot and ankle. Differences in the foot of different subjects also amount to the small variations in the robot kinematic and dynamic parameters. Furthermore, when in actual use, the effective location of the ankle joint in the robotic device also changes to some extent. This is due to the fact that the ankle joint is approximated by two joints (as explained in section 3.1) in series and different ankle motions are the result of a coupled rotation about these joints.

The kinematic and geometric parameter variations in the joint space results in the corresponding Cartesian parameters variations. Owing to this fact, the Jacobian matrix of the robot, which relates the joint and Cartesian rates, is required to be well conditioned. It is apparent that if the Jacobian matrix is not well conditioned then the small changes in the joint variables will result in very large changes in the Cartesian variables. This will further lead to difficulty in manipulator control and large trajectory following errors. Nevertheless, by choosing optimal geometric parameters of the robot, the condition number of the Jacobian matrix can be improved and the design can be made robust to parametric variations.

Another problem with cable based parallel robot is that the cables can not be subjected to very high forces. Higher forces in the cables may cause breakage or undesired elongation of the cable which will adversely affect the positional accuracy. Moreover to achieve higher forces, the required length of the air muscles also increases proportionally (as discussed in Section 5.4) which is not desirable. By optimizing the geometrical parameters of the robot, the force requirements can also be minimized.

It can be concluded from the above discussion that the design of the robot is required to be optimized to maximize its workspace, and minimize the condition number and the actuator force requirements. The dimensional analysis and optimization of the robot design may include the key design parameters such as the shapes and sizes of the parallel platforms, locations of the connection points of the cables on both these platforms and the lengths of cables. Such a design optimization is essential to reduce the above mentioned shortcomings of the robot while maintaining its inherent merits.

6. Design Optimization

Though parallel robots are in use in a wide range of applications, their potential has not been completely exploited. The reasons are obvious and have been discussed in the previous section. To address the issues of the workspace, singularity and the robust design of parallel robots, trial and error approach is normally used. This approach is based on rigorous experiments or simulation runs and intuitive judgments on the results thereafter. The main drawback of this approach is that with an increase in number of tunable parameters the required number of simulation runs increases exponentially. Moreover tuning of all the design parameters simultaneously is difficult and time consuming. Some of the researchers have tried to optimize one or more of the design objectives using some numerical methods. Several performance indices such as manipulability, isotropy, dexterity index, conditioning index, global conditioning index and global isotropy index have been defined by different researchers (Khatami & Sassani, 2002) to compare the performances of competing manipulator designs.

In a recent study (Hassan & Khajepour, 2008) the actuator forces in a cable based parallel manipulator have been optimized using a minimum norm solution. Authors have used Dykstra's projection method to optimally distribute the actuator forces among the cables and the redundant limbs. Though the force distribution among links has successfully been optimized to provide minimum norm solution, the geometrical parameters have not been tuned to minimize the actuator forces. Genetic algorithms (GA) have been used by (Sergiu et al., 2006) to optimize the workspace of a 2-DoF parallel robot using a mono-objective function. A novel kinematic design method has been implemented by (Liu, 2006) and various performance indices such as global conditioning workspace, global conditioning index and global stiffness index have been used to obtain the design parameters. A multi-criteria optimization based on conventional weighted average approach has been performed by (Lemay & Notash, 2003). The authors have proposed a combination of GA and simulated annealing algorithms to optimize workspace, dexterity and mass & size of the manipulator simultaneously. Workspace and stiffness of a modular parallel robot have been studied and optimized by (Merlet, 2003). The author has proposed a branch and prune type algorithm to improve the robot performance. A new performance index called *space utilization* has been proposed by (Stock & Miller, 2003) to evaluate the optimal kinematic design of a linear Delta robot. They have used an exhaustive search minimization method to optimize mobility, workspace and manipulability. Global conditioning index has been optimized as a result of altering the length of links by (Khatami & Sassani, 2002). The authors have used a nested implementation of two GA to obtain a mini-max genetic solution. GA has been used with constraints defined as penalty functions by (Su et al., 2001) to minimize the minimum condition numbers in the entire workspace. To validate and verify the algorithm, results obtained from GA have further been compared with the Quasi-Newton method. Architectural optimization of a 3-dof parallel robot has been performed by (Tsai & Joshi, 2000) to maximize the global conditioning index.

Mathematical Formulation

There are three major issues to be addressed in the proposed design. For better controllability the robot should have the condition number value close to unity. Thus the Global condition number is required to be minimized. The workspace of parallel robots is a critical parameter and it is required to be maximized. Lastly for a good design the norm of the actuators force vector is required to be minimized.

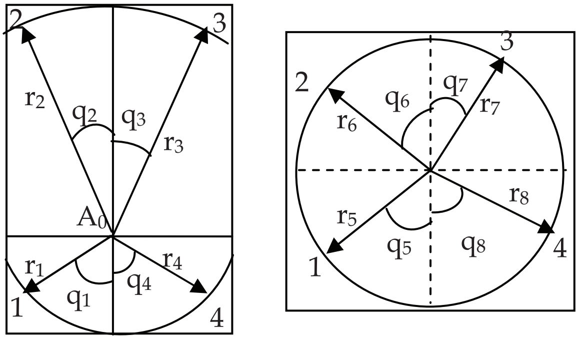


Fig. 4. Rectangular end platform and circular fixed platform with their respective geometrical parameters

Thus we have three objectives to optimize and out of these two are required to be minimized and one objective function is required to be maximized. Since the robot’s performance is very sensitive (Merlet, 2003) to its geometry, we have identified sixteen geometrical variables and have varied them to obtain different robot configurations. These variables are the polar coordinates of eight connection points on both the platforms. The formulation of the optimization problem is discussed in the following section.

6.1 The Objective Functions

(1) Minimize Global Condition Number (GCN):

$$f(q_p, r_p) = GCN\ (i) + P(i, j) \tag{36}$$

$$GCN\ (i) = \frac{\int_W (k)dW}{\int_W dW} \tag{37}$$

Here, the condition number of a particular robot design at a given orientation is denoted by ‘k’ and ‘W’ is the feasible workspace explained in the previous section 4.7. The eight sets of polar coordinates (q_p, r_p) consisting of angular positions and radial distances are used to specify the location of eight connection points (Figure 4) on the two platforms, '*i*' is the number of robot designs and $j \in W$. Further $P(i, j)$ is a penalty term defined as below.

$$P(i, j) = \left(\frac{\sigma_{\max} (J(i, j))}{\sigma_{\min} (J(i, j))} \right)_{\max} \tag{38}$$

Where $\sigma_{\min}$ the smallest singular value and $\sigma_{\max}$ is defined as largest singular value of the robot's jacobian matrix J for each workspace point.

(2) Maximize the workspace utilization index (*I*)

$$I = \frac{\varphi_f}{\varphi_T} \tag{39}$$

Here φ_f represents the number of workspace points which are reachable with the restricted stroke length of the actuator, and φ_T is defined as below.

$$\varphi_T = (\psi_{\max} - \psi_{\min})(\theta_{\max} - \theta_{\min})(\phi_{\max} - \phi_{\min})$$

Here ψ, θ and ϕ are roll, pitch and yaw angles for the moving platform.

(3) Minimize Tension norm in the workspace

$$\|U\| = \|\overline{J} M_{ext}\| \tag{40}$$

Where

$$\overline{J} = (J^T)^{-1} \tag{41}$$

And

$$M_{ext} = \begin{bmatrix} M_{x_0} & M_{y_0} & M_{z_0} \end{bmatrix}^T \tag{42}$$

and 'U' is the vector of cable forces present in each cable, J is the Geometric Jacobian matrix of the robot and M_{ext} is a 3 dimensional vector containing the required moments to move ankle joint.

6.2 The Kinematic Constraints

(1) Workspace Constraints: To find the optimum location of the connection points on the platforms their polar coordinates are varied to search the complete feasible area of the platforms. It is apparent that by changing the polar coordinates, the whole platform area can be investigated. However the points are varied within certain limits (Equations 43 & 44) so that the patient can comfortably place his foot in the robot. Variables mentioned in the Eq. 43 have been shown in Figure 4. Here γ_1 and γ_2 are two convenient limits (70 mm and 140 mm) describing the variation in the size of both platforms.

$$\frac{\pi}{6} \le q_1, q_4 \le \frac{\pi}{2}, \frac{\pi}{4} \le q_2, q_3 \le \frac{\pi}{2}$$
$$\text{and} \quad \frac{\pi}{4} \le q_5, \ldots, q_8 \le \frac{\pi}{2} \tag{43}$$

$$\gamma_1 \le r_1, \ldots, r_8 \ge \gamma_2 \tag{44}$$

(2) Stroke length check for m^{th} link: Since the air muscle actuator has a limited stroke length, it is essential to ascertain that for all different orientations of end effector the change in the link length or the stroke length is always within permissible limits.

$$l_m^{\min} \le l_m \le l_m^{\max} \tag{45}$$

(3) Tensionability Constraints: To ensure that all the cables remain in tension for all the required orientations of end effector while carrying out ROM exercises. U_m is the value of tension in the m^{th} cable.

$$U_m \ge 0 \tag{46}$$

(4) Checking maximum tension in a link: Once again, the air muscle actuators being used have a limited force capacity (β_1 =700N) and hence it is required to check that the maximum tension ($[U_m]_{\max}$) in a link should not exceed this value.

$$[U_m]_{\max} \le \beta_1 \tag{47}$$

This is a multi-objective and multivariable constrained optimization problem. To solve such a problem there are two main approaches mentioned in the literature (Deb, 2005) namely, the weighted formula approach and the Pareto optimal solution approach. To carry out the optimization, conventionally, direct search or gradient decent methods are used but these techniques become less efficient when the search space is large and is finely discretized. Further these methods sometimes get trapped into local minima and fail to provide a global optimum solution. It has been observed that when the objective function does not change over certain points in succession, these traditional methods become less effective. We propose to use Genetic Algorithms in the present optimization problem. GA works with population of points and processes them simultaneously and hence is more likely to give a global solution. As is evident from Figure 8-10, the values of condition number are quite similar in the neighborhood of any given point and the variation is very smooth. Thus discretizing the workspace does not lead to any information loss and this fact further supports our choice of GA.

7. Multi-criteria Optimization Techniques

7.1 The Weighted-Formula Approach

Conventionally to solve a multi-objective problem it is transformed it into a single-objective problem. This is typically done by assigning a numerical weight to each objective (evaluation criterion) and then combining the values of the weighted criteria into a single value by either adding or multiplying all the weighted criteria. That is, the value V of a given candidate model is typically given by one of the two kinds of formula:

$$V = w_1 \times f_1 + w_2 \times f_2 + ... + w_n \times f_n \tag{48}$$

Or

$$V = f_1^{w_1} \times f_2^{w_2} \times .... f_n^{w_n} \tag{49}$$

Here w_n denotes the weight assigned to function f_n and n is the number of evaluation criteria. A weighted formula approach has been used by (Lemay and Notash, 2003) to construct a configuration engine for robot design optimization. This approach has several associated benefits, such as, the importance of one objective over the rest can be controlled using appropriate weights and the finally obtained result is usually a Pareto optimal solution (a solution satisfying all the objectives equally). Further the approach is fairly simple and easy to use and thus is very popular. However it suffers from certain drawbacks and worst of them is the ad-hoc selection of weights for different objectives. Generally the choice of weights is based either on some trial and error experiments with different weights or on the intuitive judgment of the user. Both these choices are subjective and lack a logical ground. Further, different quality measure of varying units and scales are required to add in a single objective function which is not correct mathematically. However appropriate normalizing procedures can be used to address this problem.

7.2 The Population Based Approach

This approach entails the use of population of Evolutionary Algorithms (EA) to expand the search without using the idea of Pareto optimality. The Vector Evaluated Genetic Algorithm (VEGA) proposed by (Schaffer, 1985) is an example of this approach. This basically is the simple GA but with different selection process adapted to perform multi-objective optimization. Subpopulations for each of the objective function are selected separately based on their individual fitness. These subpopulations later, are mixed and other genetic operators such as crossover and mutation are performed on them as usual. Hence this approach has an advantage over weighted formula method that the non-commensurable or the criteria of different units are treated separately with diverse subpopulation. However the implementation of this approach is complex compared to the previous method. Further drawback of this approach is the absence of Pareto dominance concept in the selection process. Thus a design which offers a good compromised solution for all the criteria but fails to provide best solution for one of them, get discarded. Nevertheless, for the problems having large number of objectives or when the objectives are of similar nature, this approach is useful since it creates a bias in the selection process.

7.3 The Pareto Approach

The basic idea of the Pareto approach is that, instead of transforming a multi-objective problem into a single-objective function, all the objective criteria are evaluated simultaneously for a population. The dominating solutions emerging from the population are selected and ranked, based on their dominance. Two design solutions (sol$_1$ and sol$_2$) are compared and one of them (i.e. sol$_1$) is termed as dominated only if it is strictly better than the other solution (i.e. sol$_2$) with regard to at least one of the objectives. Further sol$_1$ should not be worse than sol$_2$ for other objectives. A number of Pareto approaches are mentioned in the literature and a comprehensive survey of these can be found in (Coello et al., 2002). In the present study a simple genetic algorithm based approach has been used to generate a Pareto optimal front. Despite their obvious advantages over other approaches, Pareto optimal solutions do suffer with the problem of subjectivity in choosing the best solution from the Pareto optimal solution set.

7.4 Genetic Algorithm Methodology

The fitness function and parameter selection have been discussed here and for a detailed reading on GA, the book (Deb, 2005) is recommended.

7.4.1 Generation

Initial population of 20 binary coded strings of 160 bits was generated using Knuth's random number generator (Knuth, 2000). Sixteen variables $q_1, \ldots q_{16}$ as shown below, one for each connection point location have been defined. The binary string of 160 bits has 10 bits assigned for each of the sixteen variables, thus the solution accuracy of the order of 10^{-3} can be achieved.

$$\underbrace{1101..1}_{q_1}\ \underbrace{1010..0}_{q_2}\ \underbrace{0110..1}_{q_3}\ \underbrace{1011..0}_{q_4}\ \underbrace{1010..1}_{q_5}\ \text{.........}\ \ \underbrace{1001..0}_{q_{16}}$$

The binary fractions of individual variables are converted into decimal numbers and the three performance criteria are evaluated to finally calculate the fitness function. The steps involved in GA have been further explained in subsection 7.5.

Remark: In the proposed work initially 20 robot designs are generated randomly and are represented in a binary string. The benefit of using GA is that the initial population needs not to be very large because the initial population keeps evolving and changing after each iteration.

7.4.2 Fitness Evaluation

In order to evaluate each string in the population, its fitness must be calculated. Generally GA is used for maximization problems and hence for the present case of multi-modal and multi-criteria optimization all the objectives have been converted to maximization functions. The binary strings are decoded to real numbers and the three performance criteria are evaluated to finally calculate the fitness function.

7.4.3 Reproduction

The population initially selected may not have all the good strings in terms of their fitness values. Thus good strings have been analyzed based on individual fitness and their multiple copies have been selected in the mating pool. The Roulette-Wheel Selection method (Deb, 2005) has been used for reproduction.

7.4.4 Crossover and Mutation

A four point crossover with 0.95 probability has been used and mutation has been performed with 0.01 probability. Generally crossover probability is kept close to one so that all the parent strings may get a chance to crossover. Mutation helps in finding a global optimum solution but to avoid a random search its probability is kept low.

7.5 Proposed GA Algorithm

In the proposed work efforts have been made to explore the advantages of two of the approaches discussed in subsections 7.1 & 7.3, such as, weighted formula and the Pareto optimal approach. The steps involved in the proposed algorithm are explained below.

Step 1. Choose termination criteria (either number of epochs or the error function could be used). In the present case we have chosen an error function which is defined as $(\varepsilon = 1 - Fitness)$ and the algorithm terminates when the error function is 0.01 or the fitness is 0.99.

Step 2. Initialize a random population of 20 binary strings with 160 bits in each binary. There are sixteen design variables in the present problem and each of them are represented by ten bits hence every binary corresponds to an individual robot design.

Step 3. Convert binary values of sixteen design variables to the decimal values taking their universe of discourse into account as shown below.

$$x_i = x_i^L + \frac{x_i^U - x_i^L}{2^{l_i} - 1} \times (decoded \quad value \quad of \quad binary \quad string) \tag{50}$$

Where x^L and x^U are the lower and the upper limits of the design variable, l_i is the length of binary string which is 10 in the present case.

Step 4. Evaluate the three objective criteria for these 20 robot designs and normalize them. The values of the criteria are scaled in the range 0 to1 using following linear mapping. For example GCN of the ith robot design is normalized as below:

$$GCN_{normalized} = \frac{GCN_i - GCN_{\min}}{GCN_{\max} - GCN_{\min}} \tag{51}$$

Since GA can only maximize a function and can not minimize, care has been taken to convert all the criteria into maximization functions by inverting the function which require minimization. Three constraints for the limiting values of three criteria are defined in the algorithm and any solution outside these values is discarded. The maximum values are selected on the basis of design limitations and are as below.

$$0.25 \le \varphi \;, \quad GCN \le 4.0 \quad and \quad \|U\| \le 2000$$

Step 5. Calculate the fitness function (Eq. 51) with equal priorities i.e. $w_1 = w_2 = w_3 = 0.3333$.

$$Fitness\,(i) = w_1 \times GCN\,(i) + w_2 \times \|U\|(i) + w_3 \times \varphi(i) \tag{52}$$

Step 6. Evaluate the individual fitness of the initial 20 robot designs. Using roulette wheel selection method (subsection 7.4.3) create multiple copies of binaries in the mating pool, based on their individual fitness.

Step 7. Search for any non-dominated solution as discussed in section 7.3 and store their copies in a vector.

Step 8. Perform crossover operator on randomly selected parents and later perform mutation operator on the crossed-over binaries.

Step 9. Check for the termination criteria, if not met, go to Step 3 else terminate.

The sensitivity of the three optimization criteria has been checked by changing their weights in the fitness function. The results of sensitivity analysis from the above algorithm have been shown in Table 2.

8. Results and Discussions

The weighted criteria used and the final results obtained in the present multi-criteria optimization problem are shown below in Table 2.

	Priorities (w_n)	Global Condition Number (GCN)	Actuator Force Norm (N)	Permissible Workspace Fraction (φ)
Case I	$w_1 = w_2 = w_3 = 0.333$	2.497	1537.7	0.80
Case II	$w_1 = 0.8, w_2 = 0.1, w_3 = 0.1$	2.11	1612.9	0.84
Case III	$w_1 = 0.1, w_2 = 0.8, w_3 = 0.1$	2.225	1439.9	0.70
Case IV	$w_1 = 0.1, w_2 = 0.1, w_3 = 0.8$	2.45	1657.18	0.95

Table 2. Results from four experiments with different weighted criteria.

Further, the optimum design parameters obtained from the equal priority objectives are also shown in Table 3. Experiments with different priorities (Table 2) reveal that there is a trade off between workspace and actuator forces. When minimization of actuator forces is given more weight (w_2 =0.8), the available workspace reduces to 70%. Conversely when the workspace maximization is given higher weight (w_3 =0.8), the actuator force norm increases to 1657.18 N. However the global condition number seems to be less sensitive for change in priorities. A close look at the design parameters (units are degrees and millimeters) shown in Table 3 reveals that the designs are closely symmetric. Also it can be observed that when the actuator force minimization is given more weight the connection points, as expected move farther from the center of rotation. On the other hand when workspace utilization is given more priority the connection points move close to the ankle joint.

	q1	q2	q3	q4	q5	q6	q7	q8	r1	r2	r3	r4	r5	r6	r7	r8
Case I	46	35	34	50	65	72	71	63	97	133	136	99	81	89	86	81
Case II	25	23	45	37	75	69	61	57	85	98	102	95	89	78	106	73
Case III	46	22	20	48	66	60	59	64	96	149	138	86	97	100	117	88
Case IV	48	37	31	58	66	57	58	57	81	141	143	85	71	81	89	71

Table 3. Designs from the equal priority objective optimization.

The design of the proposed robot is expected to be left-right symmetric because it is intended to be used for both right and left ankle joint rehabilitation treatments. Though the symmetry requirement was not considered as a constraint in the design optimization process, we finally get designs which are almost left-right symmetric and deviate moderately.

While using the weighted formula method all the objective functions were normalized and scaled in the range, from zero to unity. Initially all the objectives were given equal priorities and the objective function comprising the three objectives was optimized using GA. The optimized design as can be seen from the Table 2, has a global condition number as 2.4972 which is close to one and is acceptable. The design so obtained is able to use 80% of the total workspace volume and requires a force norm of 1537.7 N to move ankle joint passively. Though the condition number and the workspace utilization are acceptable but the force norm requirement is higher and is not adequate. One possible approach to address this problem is to use a different weighted formula which gives more priority to the force norm minimization compared to the workspace utilization and the condition number. Thus to explore more possibilities we further used three different types of weighted formulas giving priorities to the three objectives respectively. The results from these three objective formulations are shown in the Table 2. The difficulty in choosing one of these designs is that while giving more priority to a particular objective, we have to compromise on part of other performance criteria. For instance, if the minimization of the force norm is given more priority we do not get good condition number and the workspace remains under-utilized. This fact is evident from the results provided in Table 2. To overcome this problem we need more number of designs, displaying better performances, to make a decision on the final design. This led us to the concept of the Pareto optimal set solutions. Using non-dominated solutions approach in the present algorithm, we were able to obtain a number of good solutions providing acceptable solutions for all the three objectives simultaneously.

We have created a vector (Step7 in Section 7.5) in our algorithm to store the designs which have dominating performance in each epoch. These designs are later analyzed and the finally dominating designs out of these are selected using the same measure of dominance as discussed in Section 7.3. A plot of dominant designs is shown in Figures 5-7 and the dominating designs with their performance criteria are listed in Table 4. Design parameters from the finally dominating solutions are also displayed in Table 5. For clarity of the presentation, the Pareto optimal surface obtained for the three performance criteria is shown discretely in three plots (Figures 5-7). It is evident from the results shown that the robot performance is sensitive to its geometry and the performance criteria are also interdependent particularly the workspace and the actuator forces.

The Pareto optimal front provides at least eleven sets of solutions to choose from. Figure 5 shows solutions 2, 4, 7 & 11, providing minimum GCN and actuator forces. Design solutions 6, 7 & 10 are shown in Figure 6, which provide maximum workspace and minimum GCN. Similarly the optimal design solutions 1, 2, 3, 5, 8, 9 &10 offering minimum actuator forces and maximum workspace are shown in Figure 7. Though all of these solutions are acceptable, solution sets 1, 8 and 6 appears to be better solutions. However the workspace utilization for these solutions is low and is 70%-80% of the total workspace volume. The workspace utilization index, used in the present work is the ratio of the points satisfying actuator stroke length constraint and the total assumed workspace volume. It was found that by increasing lengths of the air muscles by 20% the workspace utilization can be increased significantly. A suitable trade-off between the performance criteria is required to establish a particular design from the available Pareto optimal solution sets.

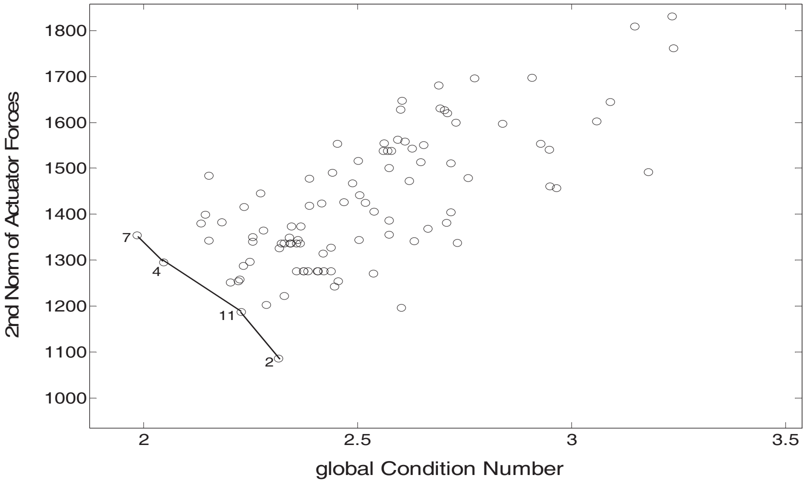


Fig. 5. Pareto optimal front for the norm of actuator forces (N) and the global condition number

Further the quality of the available workspace can also be checked with the behavior of the condition number. Hence an exhaustive search was performed in the entire workspace of the final design (D8). The variation in the condition number was plotted against X and Y Euler's angle as shown in Figures 8-10. It was found that the range of condition number variation in the entire feasible workspace was from 2.3322 to 3.2241, which is within the required limits.

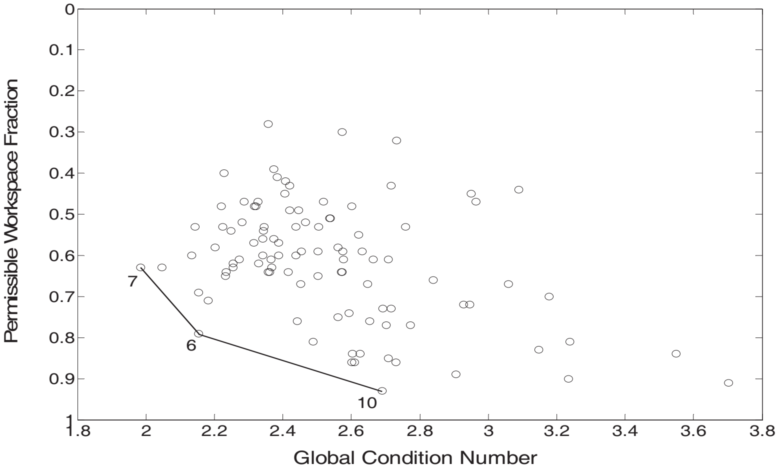


Fig. 6. Pareto optimal set for permissible workspace fraction and the norm of actuator forces (N)

Design Index	Global Condition Number	Actuator Force Norm (N)	Permissible Workspace Fraction
1	2.182	1382.098	0.71
2	2.315	1085.602	0.57
3	2.6266	1543.118	0.84
4	2.0468	1295.654	0.63
5	2.1531	1342.863	0.69
6	2.1534	1484.412	0.79
7	1.9843	1353.789	0.63
8	2.4875	1467.597	0.81
9	2.2324	1288.175	0.65
10	2.6893	1680.771	0.93
11	2.2278	1186.855	0.4

Table 4. Performance criteria for the designs of the Pareto optimal front.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
q1	43.6	43.6	66.5	43.6	32.1	50.4	40.1	72.2	50.4	59.0	43.6
q2	28.7	35.5	35.0	22.4	35.0	22.4	22.4	31.0	22.4	50.4	36.7
q3	33.8	36.1	33.8	21.2	32.7	21.2	21.2	33.8	21.2	38.4	36.1
q4	41.8	45.9	69.9	37.3	37.3	44.7	37.3	69.9	44.7	37.3	45.9
q5	66.5	72.2	64.8	69.9	69.9	64.8	64.2	64.8	60.2	69.9	65.9
q6	63.1	67.1	72.2	64.8	67.1	72.2	70.5	72.2	61.3	70.5	69.4
q7	67.6	51.0	51.0	63.6	51.0	51.0	59.0	52.2	58.5	52.2	63.6
q8	68.8	72.8	72.8	68.8	72.8	58.5	54.5	61.9	58.5	66.5	56.2
r1	80	100	100	80	100	70	100	100	100	90	100
r2	120	130	130	120	130	130	150	130	120	110	130
r3	140	130	130	140	130	130	130	130	140	110	130
r4	80	80	90	90	80	80	90	90	120	90	90
r5	80	100	80	90	80	80	80	80	90	80	100
r6	100	140	90	100	90	90	90	90	100	90	140
r7	100	130	80	100	110	80	110	80	110	80	120
r8	80	90	80	80	80	80	80	90	80	80	90

Table 5. Designs from the Pareto optimal front.

Due to difficulty in plotting results with respect to three orientations, results have been provided for three fixed yaw orientations (ϕ) such as -20°, +20° and 0°.

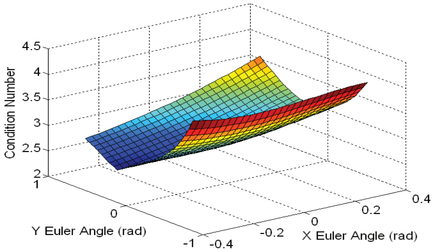


Fig. 8. Condition number vs. orientation at ($\phi = -20^o$)

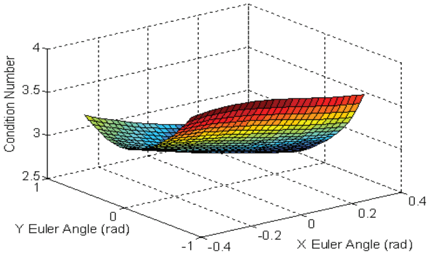


Fig. 9. Condition number vs. orientation at $\phi = 20^o$

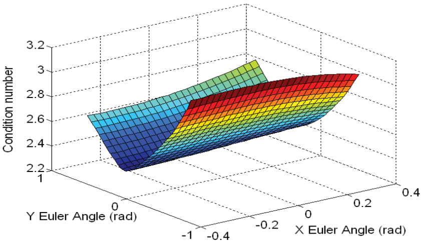


Fig. 10. Condition number vs. orientation at ($\phi = 0^o$)

It is evident from the results shown (Figures 8-10) that the condition number is a small number close to the center of rotation and increases steadily when the manipulator is moved to the workspace extremities. Further, it is also apparent from these figures that the condition number variation in the neighborhood is also smooth and the workspace is relatively free from the singularities.

9. Conclusion and Future Work

A wearable soft parallel robot for ankle joint rehabilitation has been proposed after carefully studying the complexities of human ankle joint and its motions. The proposed device is an improvement over existing robots in terms of simplicity, rigidity and payload performance. The proposed device is very light in weight (total weight is less than 2 kg excluding the weight of support mechanism) and is inexpensive. The kinematic and workspace study is carried out and the performance indices to evaluate the robot design are discussed in detail.

It is highlighted that to exploit the potential of parallel manipulators and overcome their shortcomings their design should be optimized. Three important criteria, the global condition number, the workspace volume and the norm of the actuator forces, are considered to evaluate the robot design. A multi-criteria, multi-variable optimization problem is formulated and the suitable constraints are defined. The feasibility of genetic algorithms to solve such multi-criteria optimization problem is emphasized and a brief discussion on existing techniques is also provided. We have attempted to use an algorithm which maximizes a fitness function using weighted formula approach and at the same time provides us with Pareto optimal solutions. The results obtained are discussed and several important inferences are drawn which will be helpful for the future course of this research. After carefully studying the proposed algorithm it was revealed that the possibilities of finding better solutions compared to the one presented in this chapter can not be ruled out. The present algorithm improves solutions in the direction of improving fitness of the aggregate function and does not attempt to improve the dominating solutions. Hence it will be interesting to use existing multi-objective evolutionary algorithms such as VEGA (Vector Evaluated Genetic Algorithms) and NSGA-II (Non-dominated Sorting Genetic Algorithms) which attempt to improve the dominated solutions. We have adopted GA parameters which are normally used in the literature, it will be interesting to change these parameters and study their effects on the convergence of our algorithm. Further, the Pareto solutions pose a difficulty in making a right choice from available alternate solutions. It is proposed to use Fuzzy Inferencing to assist the designer in making post Pareto decision.

It is evident from the results that the optimal design is left-right symmetric and hence connection points on the left and right side of the platforms can be considered to be the mirror images. In light of the symmetric design, it is proposed that while experimenting with newer algorithms in future, a reduced number of variables may be used.

10. Acknowledgement

The work is supported by Faculty of Engineering, Research and Development Fund, University of Auckland, New Zealand.

11. References

Alp B. & Agrawal S. K. (2002). Cable suspended robots: Design, planning and control, *Proceedings of 2002 IEEE International Conference on Robotics and Automation*, 4275-4280, 10504729 (ISSN), Washington, D.C., May 11-15, 2002.

Boudreau, R. & Turkkan, N. (1996). Solving forward kinematics of parallel manipulators with a genetic algorithm. *Journal of Robotic Systems*, Vol. 13, No. 2, 111-125.

Coello C. A. C. and Lamont G. B. (2002). *Applications of multi-objective evolutionary algorithms*, World Scientific, 981-256-106-4, Singapore.

Dai J. S.; Zhao T. & Nester C. (2004). Sprained ankle physiotherapy based mechanism synthesis and stiffness analysis of a robotic rehabilitation device, *Autonomous Robots*, Vol. 16, 207–218.

Deb K. (2005). *Optimization for engineering design, algorithms and examples*, Prentice Hall of India, 81-203-0943-X, New Delhi.

Deshmukh G. & Michael P. (1990). A modified Powell method for six-degrees-of-freedom platform kinematics, *Computers and Structures*, Vol. 34, No. 3, 485-491.

Deutsch E. J.; Lewis J. A. & Buedea G. (2007). Technical and patient performance using a virtual reality-integrated telerehabilitation system: preliminary finding, *IEEE Transactions on Neural systems and Rehabilitation Engineering*, Vol. 15, No.1, 30-35.

Dul J. & Johnson G.E. (1985). A kinematic model of the human ankle, *Journal of Biomedical Engineering*, Vol. 7, 137-143.

Geng Z. & Haynes L. (1991). Neural network solution for the forward kinematics problem of a Stewart platform, *Proceedings of the IEEE International Conference on Robotics and Automation*, 2650-2655, Sacramento, California.

Girone M. J.; Burdea G. C. & Bouzit M. (1999). The Rutgers ankle orthopedic rehabilitation interface, ASME*, Dynamic Systems and Control Division (Publication) DSC*, Vol. 67, 305-312.

Girone M. J.; Burdea G. C., Bouzit M. & Popescu V. (2000). Orthopedic rehabilitation using the Rutgers ankle interface, *Studies in health technology and informatics*, Vol. 70, 89-95.

Hassan M. & Khajepour A. (2008). Optimization of actuator forces in cable-based parallel manipulators using convex analysis, *IEEE Transactions on Robotics*, Vol. 24, No. 3, 736-740.

Hesse S.; Schmidt H., Werner C. & Bardeleben A. (2003). Upper and lower extremity robotic devices for rehabilitation and for studying motor control. *Current Opinion in Neurology*, Vol. 16, No. 6, 705-710.

Innocenti C. & Parenti C. V. (1990). Direct position analysis of the Stewart platform mechanism, Mech*. Mach. Theory*, Vol. 25, No. 6, 611–621.

Krebs H. I. (2003) Rehabilitation robotics: performance-based progressive robot- assisted therapy, *Autonomous Robots* , Vol. 15, No. 1, 7–20.

Khatami S. & Sassani F. (2002). Isotropic design optimization of robotic manipulators using a genetic algorithm method, *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, 562-567, Vancouver, October 27-30, Canada.

Knuth D. E. (2000). *The art of computer programming: Vol.2*, Addison-Wesley, 0201896834, R*eading, Massachusetts.*

Lemay J. & Notash L. (2004). Configuration engine for architecture planning of modular parallel robots, *Mechanism and Machine Theory*, Vol. 39, No. 1, 101–117.

Li L.; Zhu Q. & Xu L. (2007). Solution for forward kinematics of 6-dof parallel robot based on particle swarm optimization, *In Proceedings of the IEEE International Conference on Mechatronics and Automation*, 2968-2973. Harbin, China.

Liu G.; Gao J, Yue H, Zhang X & Lu G (2006). Design and kinematics analysis of parallel robots for ankle rehabilitation, *Proceedings of the International Conference on Intelligent Robots and Systems*, October 9 - 15, Beijing, China.

Liu X. J. (2006). Optimal kinematic design of a three translational DoFs parallel manipulator, *Robotica*, Vol. 24, 239-250.

Merlet J. P. (2003). Determination of the optimal geometry of modular parallel robots, *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, 1197-1202, September 14-19, Taipei, Taiwan.

Nam Y. J. & Park M. K. (2004). Geometric approach to forward kinematics of casing oscillator, *Proceedings of Annual Conference of IEEE Industrial Electronics Society*, 2582-2587, Busan, Korea.

Oyama E.; Maeda T., Tachi S., MacDorman K. F. & Agah A. (2002). On the use of forward kinematics model in visually guided hand position control analysis based on ISLES model, *Neuro computing*, Vol. 44-46, 965-972.

Parenteau C. S.; Viano D.C. & Petit P.Y. (1998). Biomechanical properties of human cadaveric ankle-subtalar joints in quasi-static loading. *Journal of Biomechanical Engineering*, Vol.120, No. 1, 105-111.

Pusey J. L.; Fattah A., & Agrawal S. K. (2003). Design and workspace analysis of a 6-6 cable-suspended parallel robot, *Mechanism and Machine Theory*, Vol.39, No. 7, 761-778.

Sadjadian H. & Taghirad H. D. (2005). Comparison of different methods for computing the forward kinematics of a redundant parallel manipulator. *Journal of Intelligent and Robotic Systems*, Vol. 44, No. 3, 225-246.

Schaffer J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms, *Proceedings of the first international conference on Genetic Algorithms*, 93-100, Lawrence, Erlbaum.

Sergiu S.; Maties V. & Balan R. (2006). Optimization of workspace of a 2 DOF parallel robot, *Proceedings of the 2006 IEEE, International Conference on Mechatronics and Automation*, 165-170, June 25 - 28, 2006, Luoyang, China.

Siegler S.; Chen J. & Schneck C. D. (1988). The three-dimensional kinematics and flexibility characteristics of the human ankle and subtalar joints- Part I: Kinematics, *Journal of Biomechanical Engineering*, Vol. 110, 364-373.

Stock M. & Miller K. (2003). Optimal kinematic design of spatial parallel manipulators: application to linear delta robot, *Transactions of ASME*, Vol. 125, 292-301.

Stump E. & Kumar V. (2006). Workspaces of cable-actuated parallel manipulators, *Journal of Mechanical Design ASME*, Vol. 128, 159-167.

Su X. Y.; Duan B. Y. & Zheng C. H. (2001). Genetic design of kinematically optimal fine tuning Stewart platform for large spherical radio telescope, *Mechatronics*, Vol. 11, 821-835.

Tsai L. W. & Joshi S. (2000). Kinematics and optimization of a spatial 3-UPU parallel manipulator, *Journal of Machine Design*, Vol. 122, 439-446.

Yee C. S. & Lim, K. B. (1997). Forward kinematics solution of stewart platform using neural networks, *Neuro computing*, Vol. 16, 333-349.

Yoon J.; Ryu J. & Lim K. B. (2006). Reconfigurable ankle rehabilitation robot for various exercises, *Journal of Robotic Systems* , Vol. 22 (Supplement), S15–S33, 2006.