Faculty of Science and Technology
Control and Instrumentation Research Group
Engineering Department

THE DEVELOPMENT OF A MULTI-ARM MOBILE ROBOT SYSTEM FOR NUCLEAR DECOMMISSIONING APPLICATIONS

Mohamed Jeylani Bakari
MSc., PG Dip., BSc.

A thesis submitted for the degree of Doctor of Philosophy

December 2008

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

ABSTRACT

This PhD thesis is based in the field of robotics and introduces a case study of the design and development of a multi-arm mobile robot system for nuclear decommissioning (MARS-ND). A key premise underlying the research was to develop intelligence in the robot that is similar to the cooperation and communication between the human brain and its two arms; hence the human body was adopted as the starting point to establish the size and functionality of the proposed system. The approach adopted for this research demonstrates the development, integration and configuration of a multi-arm robot system which consists of two human arm-like off-the-shelf manipulators whose joints are controlled using potentiometer sensors and hydraulic actuators. Using the manipulators' sensor feedback, a wide variety of complex tasks found in the rapidly expanding field of nuclear decommissioning can be undertaken. The thesis also considers the issue of collaboration, collision detection and collision avoidance between the two arms of MARS-ND. As part of the final stage of this research the author participated in a collaborative research project with the Sugano Laboratory at Waseda University, Tokyo, Japan. The three major research issues addressed in this thesis are:

1. The selection and integration of off-the-shelf hardware in the development of MARS-ND using the latest technology available for robotic systems
2. The creation of a suitable control system for the robot arms; and the building of an advanced, user-friendly interface between the robot system and the host computer
3. The investigation and implementation of collaboration, coordinated motion control and collision detection & avoidance techniques for the robot arms

The hardware and software integration for the whole robotic system is explained with the proposed software architecture and the use of National Instruments (NI) functions and tools to

control the movement of the arm joints and the performance of a selected decommissioning task. This thesis also examines the operational software applied within the research through its discussion of four interlinked areas:

1. The control software and hardware interface for the MARS-ND and the controller architecture
2. The application of an NI Compact FieldPoint controller and FieldPoint I/O modules to facilitate wireless communication between the Multi-Arm Mobile Robot system and the user interface in the host PC
3. The use of Measurement and Automation Explorer (MAX) and LabVIEW software tools for calibration and the building of user interfaces required for sending and receiving the signals needed to control the robot arm joints accurately
4. The application of a PID toolkit in LabVIEW for the design of a simple PID controller for the individual arm joints with a potentiometer sensor fitted inside each joint in order to provide a feedback signal to the controller

The thesis concludes that MARS-ND is a good example of a robotic system specifically designed for hazardous nuclear decommissioning applications. It demonstrates the complexity of such a system from a number of aspects such as the need for mobility, control, sensor and system design, and integration using modern tools that are available off-the-shelf. In addition the use of these modern tools allows a single mechatronics engineer to design, integrate, interface and build a motion control system for MARS-ND as compared to the traditional way of building a similar robot by a team of specialised engineers. The contribution this research makes to the design and building of multi-arm robot system for nuclear decommissioning industry concerns its size and mobility using a mobile platform to transport the multi-arm robot system. In addition links have been made between Lancaster University and Waseda University in the context of the development of multi-arm robot systems.

DECLARATION

I declare that this thesis consists of original work undertaken solely by myself at Lancaster University between 2004 and 2008; where work by other authors is referred to and has been properly referenced.

Mohamed J. Bakari

December 2008

PUBLICATIONS

Journal Paper:

Bakari, M. J., Zied, K. M., and Seward, D. W., 2007. Development of a Multi-Arm Mobile Robot for Nuclear Decommissioning Tasks. *International Journal of Advanced Robotic Systems (ARS)*, Vol. 4, No. 4, pp. 387 – 406.

Conference Papers:

Bakari, M. J., Seward, D. W., Joyce, M. J., and Mackin, R. O., (2008). "The Development of Tele-Operated Two-Armed Mobile Robot System for Nuclear Decommissioning Operations". *2nd International Joint Topical Meeting on Emergency Preparedness & Response and Robotic & Remote Systems*. Albuquerque, New Mexico, USA.

Bakari, M. J., Seward, D. W., Shaban, E. M., and Agate, R. Y., (2006). "Multi-Arm Mobile Robot for Hazardous Nuclear Decommissioning Tasks". *23rd International Symposium on Automation and Robotics in Construction (ISARC)*. Tokyo, Japan.

Agate, R., Pace, C., Seward, D., Shaban, E., and Bakari, M., (2006). "Control Architecture Characteristics in Autonomous Mobile Construction Robots". *23rd International Symposium on Automation and Robotics in Construction (ISARC)*. Tokyo, Japan.

Bakari, M. J., and Seward, D. W., (2006). "HUMAN-LIKE MECHANICAL MANIPULATOR, The Design and Development of Multi-Arm Mobile Robot for Nuclear Decommissioning". *3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Setubal, Portugal.

Seward, D. W., and Bakari, M. J., (2005). "The use of Robotics and Automation in Nuclear Decommissioning". *22nd International Symposium on Automation and Robotics in Construction (ISARC)*. Ferrara, Italy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I always feel my life is indebted to Allah, the most gracious and the most merciful.

I would like to express my deepest thanks to my supervisor **Professor Derek W. Seward** for his excellent supervision, support, encouragement and valuable guidance throughout this research work.

Special thanks to the Engineering & Physical Science Research Council (EPSRC) for the financial support in the form of sponsoring my PhD research.

Also special thanks to the Japan Society for the Promoting of Science (JSPS) for the financial and travel support in the form of sponsoring me to undertake a collaborative research project with the Sugano Laboratory at Waseda University in Tokyo, Japan.

I would like to express my deepest thanks to all the staff of Engineering Department at Lancaster University, especially **Mr. Mark Salisbury**, **Mr. Andrew Verden**, **Mr. Andrew Gavirluk** and **Mr. Bob Mackin** for their valuable help and advice during the research work.

Finally, I would like to express my gratitude and to dedicate this work to my wife **Allison** and our beautiful daughter **Miryam** for their continual support in my life.

ABBREVIATIONS

ANL	Argonne National Laboratory
ASM	Advanced Servo-Manipulator
APF	Artificial Potential Field
BSM	Bilateral Servo Manipulator
CAD	Computer Aided Design
cFP	Compact FieldPoint
D&D	Decontamination and Decommissioning
DAWM	Dual-Arm Work Module
DOF	Degree of Freedom
D-H	Denavit-Hartenberg
DAQ	Data Acquisition
FE	Finite Element
FK	Forward Kinematic
GUI	Graphical User Interface
IK	Inverse Kinematic
IAEA	International Atomic Energy Agency
JNC	Japan Nuclear Cycle
LS	Least Square
MARS-ND	Multi-Arm Robot System for Nuclear Decommissioning
mSv	millisievert
mtbf	Mean-time-between-failure
MAX	Measurement and Automation Explorer
ND	Nuclear Decommissioning
NDA	Nuclear Decommissioning Authority
NRC	Nuclear Regulatory Commission

NI	National Instruments
NED	Networked Embedded Device
ORNL	Oak Ridge National Laboratory
PID	Proportional Integral Derivative
PIP	Proportional Integral Plus
RTDP	Robotics Technology Development Program
RRT	Rapidly-Growing Random Trees
SDK	Software Development Kit
SRAST	Symbolic Robot Arm Solution Tool
SE	System Engineering
URD	User Requirement Document
VME	Virtual Machine Environment
VI	Virtual Instrument
VRML	Virtual Reality Modelling Language
WAGR	Windscale Advanced Gas Cooler Reactor

NOMENCLATURE

Roman Alphabetic Symbols

a_i	the distance between z_{i-1} and z_i axis measured along the negative direction of x_i-axis (DH convention)
d_i	the coordinate of the origin O_i on z_{i-1}-axis (DH convention)
d	influence distance
d_u	gain distance
F_a	attractive force
F_r	repulsive force
F	Force
g	acceleration gravity
I	identity matrix
J	Jacobian matrix
J_e	end-effector Jacobian matrix
J_o	collision avoidance point Jacobian matrix
J_e^+	pseudo-inverse of J_e
J^T	transpose Jacobian
K	spring constant
m	mass of object
M	inertia matrix of the object
M_o	joint mobility
T	transformation matrix
T_{i-1}^{i}	homogeneous transformation matrix for joint i relative to i-1
$\dot{x}_e$	end-effector velocity vector
$\dot{x}_0$	collision avoidance point velocity vector
w_0	angular velocity of the object

Greek Alphabetic Symbols

α_i	twist angle (DH convention)
θ	the vector of joints angle
θ_i	The i^{th} link angle
$\dot{\theta}$	the vector of joints velocity
Ψ	arm angle

CHAPTER 1

INTRODUCTION

1.1 Introduction

This thesis examines the development of a Multi-Arm Mobile Robot System (MARS) for specific application in the nuclear decommissioning (ND) industry. For this research MARS-ND consists of a Brokk 40 demolition robot (Brokk 2005) and two seven – function manipulators called Hydro-Lek (Hydro-Lek 2005). The development of MARS-ND for this research is comprised of three practical stages:

1. The selection of pre-tested off-the-shelf hardware and its integration in the development of MARS-ND
2. The creation of a suitable control system for the robot arms and the building of user interfaces between the robot system and the host computer
3. The investigation and implementation of collaboration, coordinated motion control, and collision detection and avoidance techniques between the two robot arms

Three key research questions emerge from these three stages and form the basis of this thesis:

- How to use modern commercially available off-the-shelf tools to build MARS-ND in a robust and cost-effective manner
- How to develop a communication interface between the different tools used to develop MARS-ND through one software application

- How to create motion coordination algorithms for successful collaboration and collision avoidance between the two arms when carrying out a specific task, such as dismantling an object or pipe cutting. These are discussed in further detail in the following sections.

1.2 The application of robotic systems for decommissioning and dismantling tasks

Over the next two decades hundreds of nuclear facilities will come to the end of their working lives and require decommissioning (Waste and Decommissioning 2006). These range from nuclear power stations, submarines, fuel processing plants and mines. The UK Nuclear Decommissioning Authority (NDA) estimate that the total cost of dealing with the nuclear legacy in the UK is nearly $100Bn (NDA 2006).

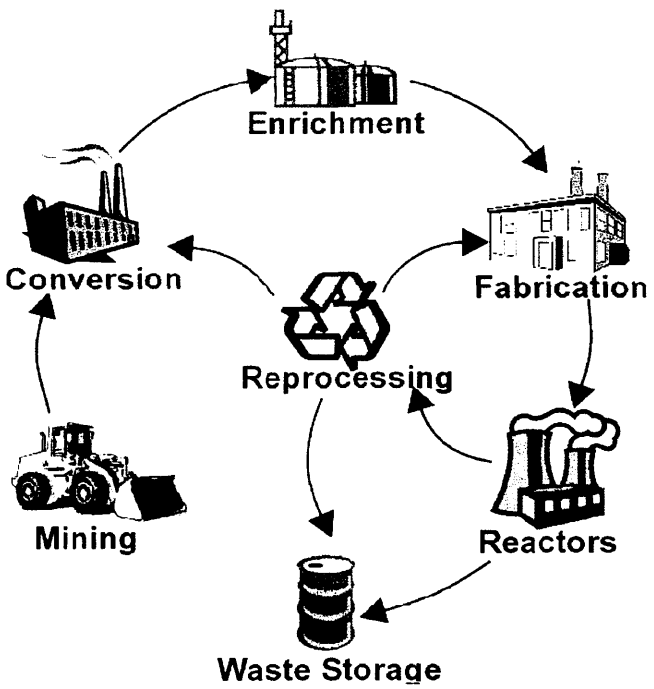


Figure 1.1 The nuclear fuel cycle (World Nuclear Association 2005)

The majority of the decommissioning process uses well established demolition techniques, however there is an overwhelming complication in the case of the decommissioning of

nuclear facilities because of the hazard of radiation release and its potential impact on workers, the general public, and the environment. There is considerable political pressure to undertake nuclear decommissioning tasks quickly and, in many cases, the only means of facilitating this is through the use of automation and robotics in order to reduce the dose exposure of workers. The obvious type of facility that requires decommissioning is the redundant nuclear power station but a nuclear power station is only one part of a complete fuel cycle as shown in Figure 1.1.

During the process of decommissioning considerable effort must go into containing the spread of contamination to other parts of the facility by reducing air-born dust and fluids. For this reason many operations take place inside specially constructed cocoons. Figure 1.2, shows methods for minimising radioactive waste from D&D of nuclear facilities.

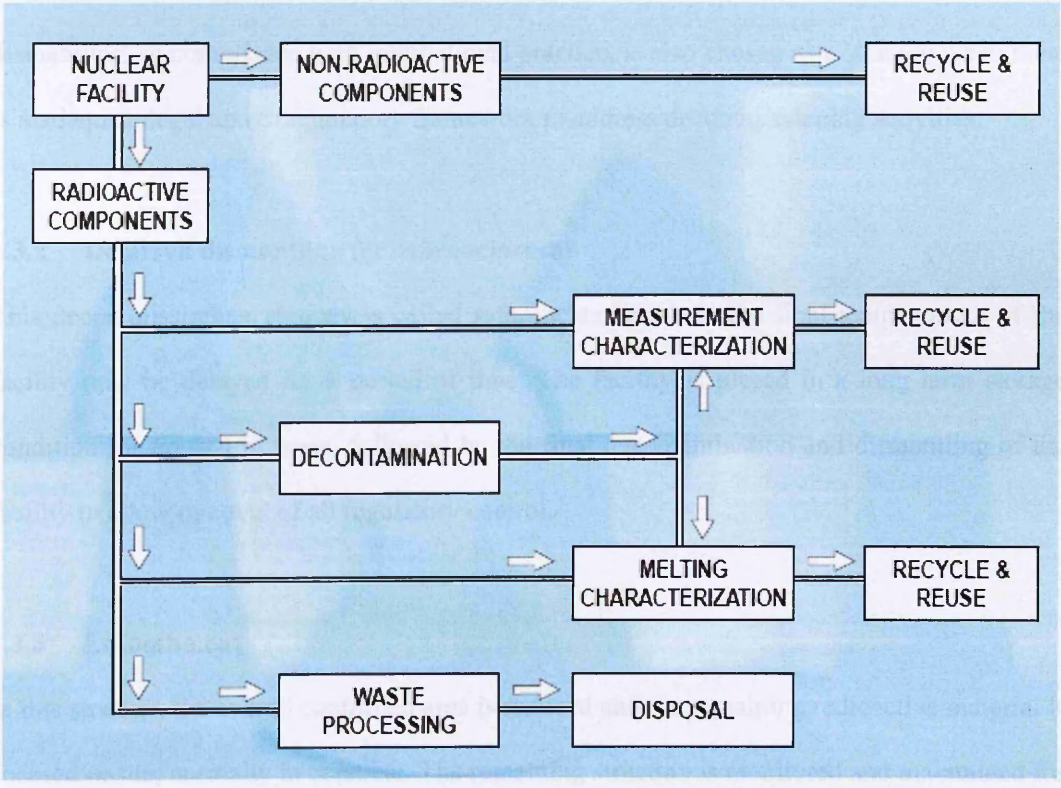


Figure 1.2 Methods for minimising radioactive waste from D&D of nuclear facilities

(IAEA-401 2001)

1.3 The increased demand for 'immediate demolition' as opposed to a 'safe enclosure or deferred dismantling' period

In the mid 1990s, the International Atomic Energy Agency (IAEA) adopted three decommissioning strategies, as described below; these strategies are currently applied in all IAEA safety standards (IAEA 2003, IAEA 1999, IAEA 2001). The strategy chosen can have an impact on the development of new technologies necessary for the dismantling of facilities, their characterisation or their decontamination.

1.3.1 Immediate dismantling

The implementation of an immediate dismantling strategy normally begins very soon after shutdown of the plant, usually within five years. This strategy leads to the development of remote control equipment or robotic systems that can access difficult to reach areas of the plants with the objective to reduce the radiation exposure to the workers. Immediate dismantling, in compliance with international practice, is also chosen as a strategy when there is inadequate legal and or regulatory framework to address decommissioning activities.

1.3.2 Deferred dismantling (or safe enclosure)

This decommissioning strategy is called safe enclosure where the final temperament of the facility may be delayed for a period of time. The facility is placed in a long term storage condition for up to 100 years, followed by the final decontamination and dismantling of the facility to allow removal of all regulatory control.

1.3.3 Entombment

In this strategy, the overall controlled area is reduced and the remaining radioactive material is encased onsite, normally in concrete. The remaining structure is monitored and maintained for an indefinite of time.

The stages of decommissioning and the end state of the facility are mostly governed by regulatory, economic, or environmental issues. The strategy chosen to reach these conditions depends on factors that include the existence of decommissioning funds or the availability of requested and financially sound technology. For example, the existence of adequate technology in a country faced with a specific, early elaborated and well substantiated decommissioning plan will lead to immediate dismantling of the facility. A lack of technological means, however, can lead to deferred dismantling with the hope that technological development will tackle the issues in the future. Although long term safe enclosure does not usually require sophisticated decommissioning and dismantling methods and techniques, dismantling operations performed in the extended future can prove to be more difficult than expected due to the degradation of equipment (IAEA 1999). In practice few references indicate that availability of technology is a major factor influencing the selection of decommissioning strategy. On the contrary, the strategy chosen may have an impact on the development of new technology such as robotic systems necessary for the dismantling of facilities.

In current practice the preferred strategy is immediate dismantling. There are several reasons for this. Immediate dismantling minimises costs associated with deferred dismantling and allows immediate decommissioning to occur using remote control equipment such as robotic systems. Furthermore immediate dismantling means that new nuclear power plants can be built on the same site to meet the increasing demands of energy. The immediate dismantling strategy paves the way for the development of sophisticated robotic systems from widely commercially available off-the-shelf technologies. This process proved difficult in the past because the technology was unable to support the requirements of the decommissioning industry in relation to robotic systems as is discussed further in Chapter Two.

Another important factor in the selection of a decommissioning strategy is the radiological risk (IAEA 1995). In the context of deferred dismantling, radioactive decay over time leads

directly to a reduction in the level of radiological risk to the workers and the public. Radiological risks can be higher during early dismantling because of the higher radiation levels. The application of remote equipment can reduce these risks and facilitate immediate dismantling and decommissioning with the associated advantages outlined above.

1.4 The need to protect workers from radiation

A major objective of decommissioning is to dismantle plant systems and resize, package and ship components to waste disposal facilities. This process can expose workers to high doses of radiation as well as other significant safety risks. The Nuclear Regulatory Commission's regulation 10 CFR 20 states that an occupational worker cannot receive more than 50 mSv per year for the full body dose (NRC 1999; Code of Federal Regulations 1993), once this dose has been reached the worker has to stop working immediately. This necessitates an increased number of workers to be employed in order to accomplish the necessary task. By using robots the number of workers is minimised which also creates many additional savings including a reduction in the quantity of protective clothing needed, and decreased administration costs. A study published by Marian and Rowan (1987) indicates that worker exposure costs are more than $500,000 per man-Sv. A utility executive writing in Nuclear Engineering International (1990) stated that every dollar spent on robotics is doubled in return.

Robots are now used widely in the nuclear industry and their primary use in decommissioning applications is to reduce the radioactive dose levels to which workers are exposed by executing hazardous tasks that are dangerous to workers, to perform automated and repetitive work, or access areas that are difficult to reach or present life threatening hazards. In nuclear science, protection of workers has become a catalyst for the development of robotics, but profitability is also a strong motivator. It has become more economically feasible and desirable to use robots for a variety of decommissioning activities compared to conventional, manual work methods because of higher standards for worker safety, more severe regulatory and judicial penalties for

violations and injuries, and increased capabilities and versatility of robotic devices. Furthermore, a remote system is often the only way to enter a very high radiation field. Nearly all Decontamination and Decommissioning (D&D) activities that are too hazardous for direct human contact are presently executed using robotic systems. Many of these robotic systems are custom-designed for specific projects, but this makes them expensive, often unreliable and limited (IAEA-395, 2001). Whereas conventional industrial robots have a mean-time-between-failure (mtbf) of 60,000 hours, a typical customised one-off solution has a mtbf of only 5-6 hours.

1.5 Concerns regarding the application of robotics for decommissioning tasks

Common applications of robotic systems in industry are driven by requirements to move parts, tools and materials through pre-programmed sequences to perform a variety of tasks. Although not a pre-requisite for robotic systems, many applications are also driven by additional requirements to automate processes that are in some way too dangerous and too costly for human's to perform manually. Many decommissioning contractors have experienced significant problems with complex customised robotic systems and are therefore sceptical about their deployment. The reasons are as follows:

- Technology limitations. Some contactors believe that existing technology in robotics is too basic to solve decommissioning problems reliably.
- Inflated claims and unrealistic customer expectations. Robotic companies often exaggerate the capabilities of their current and future robotic systems for commercial and profit reasons. This has made contractors wary of using these systems for decommissioning tasks.
- Acceptance, fear and prejudice. Many contractors have fears concerning the application of robotic systems for decommissioning purposes and are unable to see

the potential of robotic systems to make decommissioning processes safer and reduce the risks to workers. These fears include:

1. Potential of the robotic systems
2. Reliability and cost of specialist repairs
3. Slow performance compared to humans
4. Fear of job replacement by the robotic system

Even though many contractors have these fears and questions, companies still continue to utilise and develop complex robotic systems for D&D tasks.

1.6 Traditional and modern robots for nuclear decommissioning

Many of the facilities currently being decommissioned are thirty to forty years old. Robotic technology was in its infancy when these plants were designed and built; they were therefore not constructed for robotic operations and are not robot friendly. They are typified by a complex interwoven web of pipework and vessels, and can be considered as semi-structured. One type of robot that was initially deemed suitable for nuclear decommissioning, were those produced for the factory automation market. Many factory automation robotic technologies have been adapted and used for the decommissioning of nuclear facilities. For example, Challinor 1996 and Mort 1998 discuss the use of industrial robots to dismantle vessels removed from buildings in the process of being decommissioned. Factory automation robotic technologies have many technical attractions such as good reliability, high accuracy and the ability to replicate activities, they are also relatively inexpensive; but they have many disadvantages some of which are outlined below.

The main objectives of industrial robots and those designed for nuclear decommissioning are different. Factory robots were designed for factory automation and therefore have features designed within their hardware, software and firmware to address factory automation issues. These features can be at odds with the requirements of machines used for nuclear

decommissioning. In decommissioning it is necessary to take into account the large variety of items that have to be dismantled and the geometric changes that occur during dismantling; factory automation is not designed to tackle these types of problems therefore modification and the adoption of a different type of hardware and software are required. In addition robot functions are pre-programmed for factory automation, whereas decommissioning robots require a system that can be programmed rapidly. Factory robots provide a means of increasing productivity and lowering production costs. In the decommissioning process it is necessary to use remote methods wherever it is difficult or impossible for the human workforce to enter, this often increases the decommissioning costs but there is no alternative but to use a remote system. In the future it is likely that there will be more stringent regulatory controls which will limit man-entry further creating the need for an increase in remote techniques.

The technology that is currently most applicable to nuclear decommissioning are Telerobotic Systems. Within these systems there is some form of human control within the loop, this is an important element for the effective use of a robotic system in nuclear decommissioning. There is a lot of research currently underway to extend the limited functionality of existing robot characterisation systems in order to meet the needs of increases in decommissioning in the future. One example is architecture proposed by Cragg and Hu (2003). In this research a Brokk machine has been selected and modified in order to suit the specific nuclear decommissioning task being examined, which is pipe cutting.

1.7 Multi-Arm robot systems used for D&D tasks

Until recently multi-arm robot systems have been deployed by either a rigid boom overhead transporter, or by a crane. Below are a selection of examples of multi-arm robot systems with a summary of their control systems and levels of intelligence. There are no technical

specifications provided with many of these systems due to the classified nature of their use within the nuclear decommissioning industry.

1.7.1 Dual-Arm Work Module

The Dual-Arm Work Module (DAWM) shown in Figure 1.3 is based on two Schillings Titan II hydraulic manipulators (Schillings 2005) mounted to a 5 degree of freedom (DOF) base. All actuation is hydraulic. The 5-DOF articulation provides centre torso rotation, linear actuation to change the separation between the arms, and arm base rotation joints to provide individual elbow up, elbow down or elbow out orientation for each arm. DAWM is mounted to the bottom of a rigid boom overhead transporter; but is also sometimes deployed by a crane.

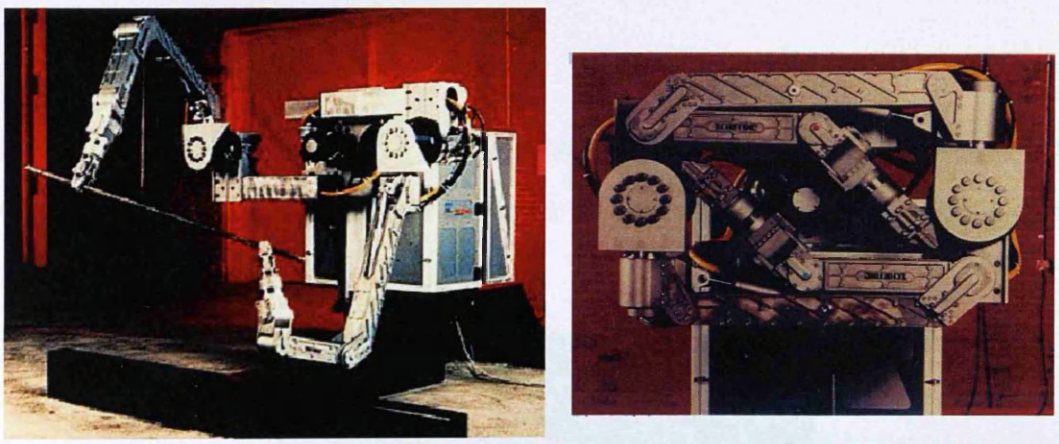


Figure 1.3 Dual Arm Work Module (DAWM) (Noakes 1999)

The control architecture for DAWM is based on:

- Unix operating system
- Operator interfaces
- Virtual Machine Environment (VME) backplane-based multiprocessor VxWorks targets
- Five single-board computers

The five single-board computers are fitted in the VME backplane with one each for asynchronous control and communications, left master, left slave, right master and right slave.

All computer processing unit boards are located in the master backplane and the slave side backplane is connected to the master via a VME-VME bus repeater. The real-time software is written in C++ and the operator interfaces are run on two separate monitors. The DAWM level of intelligence is master-slave force reflecting tele-operation. It uses joysticks and switchboard for the control of the camera pan tilt and lens. DAWM was developed at Oak Ridge National Laboratory (ORNL) by Robotics Technology Development Program (RTDP) (McKay 1993). DAWM has been used at Argonne National Laboratory (ANL) in Chicago for study issues related to multi-arm manipulation, control, automation, operations and tooling. Some of the disadvantages of DAWM is that it uses unsophisticated hardware and software, and the operator interface is not user friendly (Noakes 1999).

1.7.2 RODDIN

RODDIN as shown in Figure 1.4 is a crane deployed work platform, designed by CYBERNETIX, France. This system is used for pipe and metal cutting at decommissioning sites.

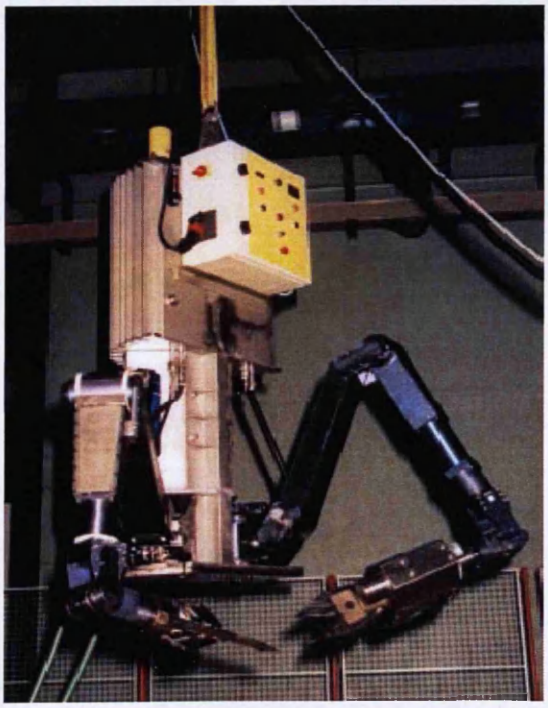


Figure 1.4 RODDIN (Desbats 2005)

The RODDIN platform is equipped with two 6-DOF hydraulic manipulators (SAMM or MAESTRO) with an onboard or remote hydraulic power unit. The MAESTRO is a 6-axis manipulator with force feedback. It has a payload of 80kg and a length of 2.4m.

1.7.3 Advanced Servo-Manipulator

The Advanced Servo-Manipulator (ASM) as shown in Figure 1.5 is a remotely operated servo manipulator system designed and built at the Oak Ridge Laboratory (ORNL), Chicago, USA (Kuban 1987).

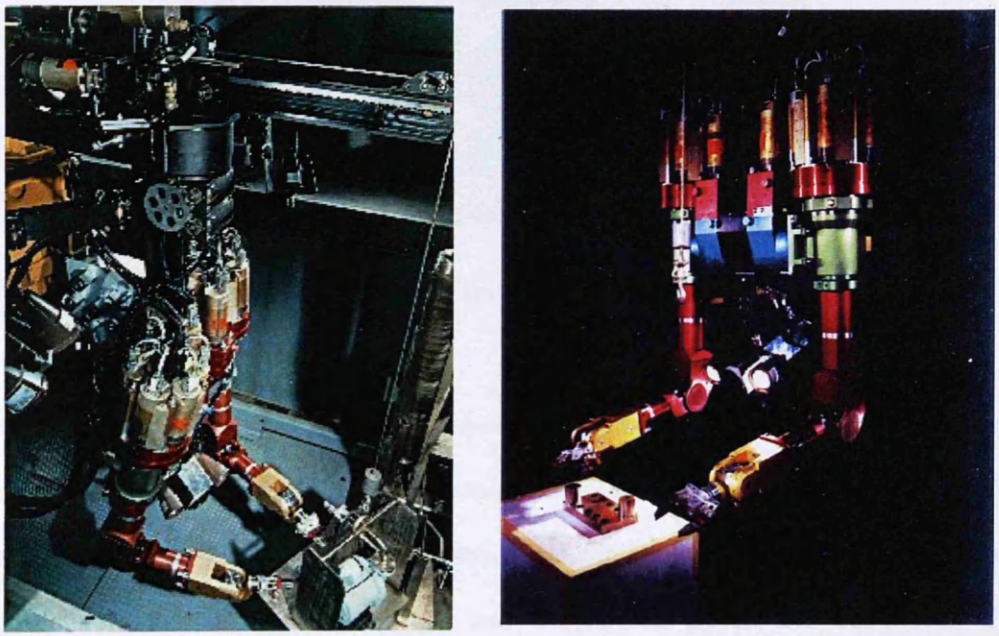


Figure 1.5 Advanced Servo-Manipulator (ASM) (Parker and Draper 1998)

ASM has been used to support nuclear fuel processing applications and to dismantle components including tubing jumpers, instruments, motors and tanks. The remote manipulator is a dual arm system that provides force feedback to the operator performing the task. The manipulator system is attached to an overhead crane and each manipulator has 6-DOF. Remote television cameras are fitted with the manipulators and can be repositioned by the operator using the same set of controls provided for the servo manipulator.

1.7.4 Dual-Arm Mobile Working Platform

The Dual-Arm Mobile Working Platform (PTM) shown in Figure 1.6 is a dual-arm intelligent tele-operation system. It has been developed by the Atomic Energy Commission (AEC) of France.

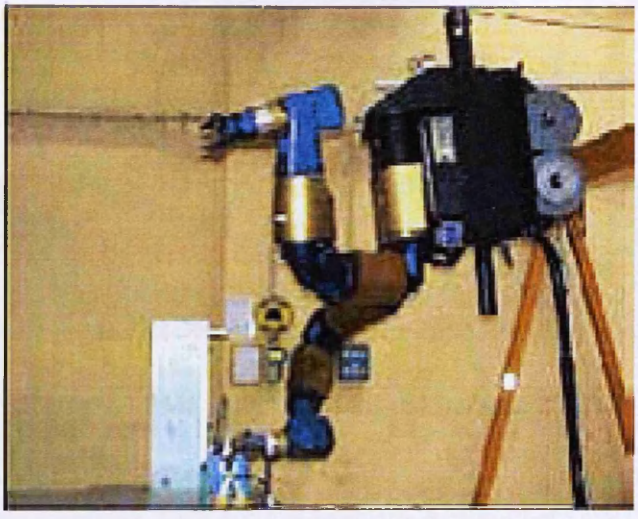


Figure 1.6 Dual-Arm Mobile Working Platform (PTM) (Desbats 2005)

PTM is specifically designed to carry out maintenance tasks in fuel reprocessing facilities where standard wall-mounted mechanical manipulators cannot be used. The arm in the PTM system is a general purpose manipulator developed for remote tele-operation and robotic applications in the area of maintenance and intervention on process equipment. The manipulator is a 7-axis redundant manipulator with 25kg payload capacity and force feedback control.

1.7.5 LMF Vehicle

The LMF robot shown in Figure 1.7 was developed by the Cybernetix Group in France (Fidani and Baraona 2001). It is a modular vehicle for remote controlled intervention in hazardous internal and external environments and is used for purposes such as surveillance, inspection, maintenance, and decommissioning in nuclear facilities.

Figure 1.7 LMF vehicle (Fidani and Baraona 2001)

The mobile base is designed to go over obstacles and is equipped with a heavy duty hydraulic tele-manipulator for master-slave control with force feedback. The control and data are video images which are transmitted by an umbilical cable. The dimensions of LMF are as follows: Width: 850mm, Height: 1900mm, Length: variable. The control station for LMF includes a monitor for displaying 3D video images; a monitor showing a 3D simulation of the vehicle with collision detection; a control desk for the mobile platform; a control desk for the control of the tele-manipulator; and one master arm which can be used as a joystick for moving the LMF arms.

1.7.6 M-2 Manipulator System

The M-2 manipulator system consists of Master and Slave manipulators as shown in Figure 1.8. The M-2 system has the following features:

- Force reflecting master-slave manipulator
- Dual 6-DOF arms
- 100lb lifting capacity

- Removable wrist assembly and geared azimuth drive 540°
- Position-position bilateral force reflection mode
- Automated slave position calibration
- Camera positioning with 4-DOF, power zoom, focus, and iris

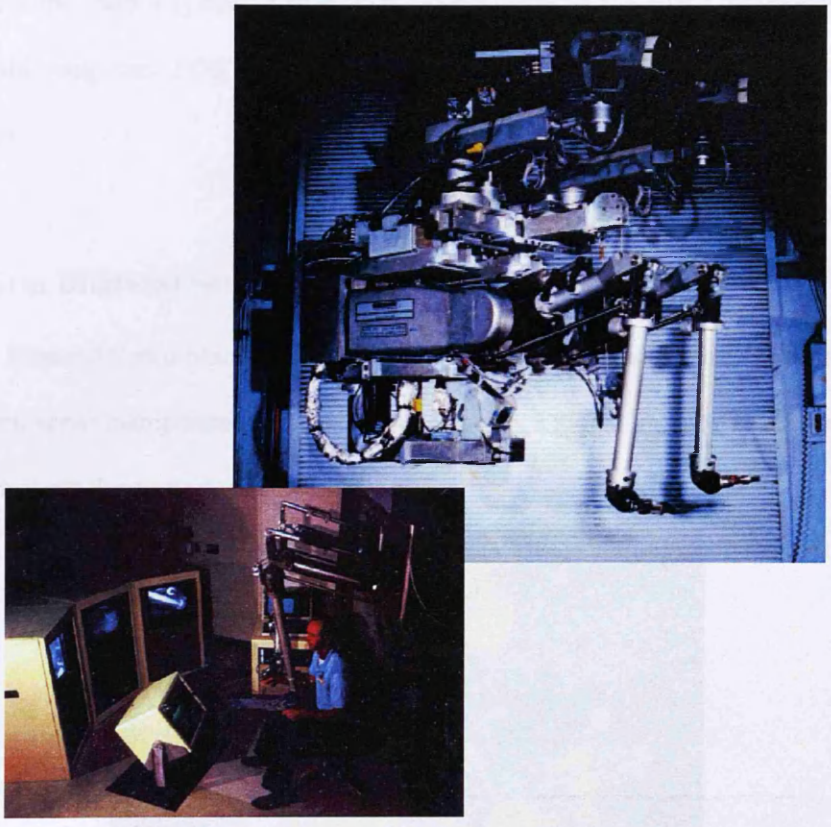


Figure 1.8 M-2 Manipulator System (Killough *et al* 1986)

The M-2 system is a force reflecting master-slave servo-manipulator developed by ORNL as research study for the consolidated fuel reprocessing program (Killough *et al* 1986). The M-2 system provides force feedback for both arms. The force feedback facilities allow the operator to estimate the forces at the work site and gain a sense of the feel characteristics of the mechanical manipulator. The force reflection makes both the slave and master friction appear at the master. The M-2 system manipulators use metal tape and pulley power transmissions which provide very low friction. The original control for the M-2 system was built entirely from analog electronics. Each joint had its own servo control loop containing the basic position-position servo loop which limited the performance and reliability of the M-2 system.

In addition, because of the drift characteristics of the potentiometers, capacitors and resistors the analog circuit had to be retuned weekly when the manipulator was used frequently. A digital control system was later used to replace the analog system and this has improved the force-reflection performance, reliability, maintenance and operational features of M-2 system. The hardware for the control system is built from general-purpose bus-based microcomputers with single board computers. FORTH programming language is used as the main software to control the arms.

1.7.7 Two-Arm Bilateral Servo Manipulator System

The Two-Arm Bilateral Servo Manipulator System (BSM) as shown in Figure 1.9 consists of two arm bilateral servo manipulators, a transporter, cameras, a signal and power transmission system, and a man-machine interface subsystem.

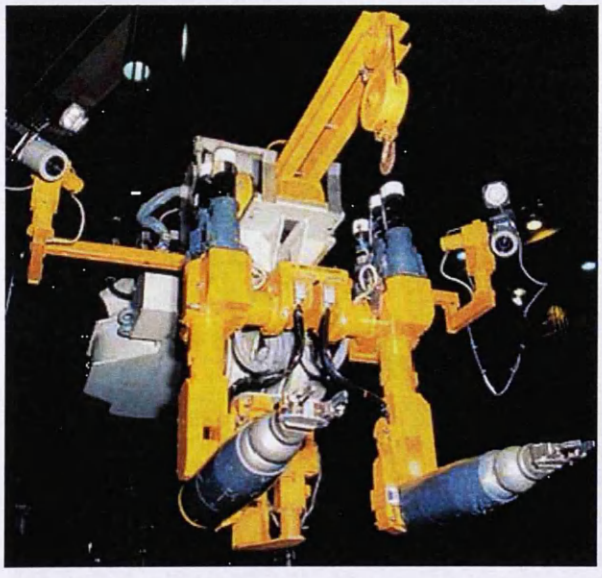


Figure 1.9 Two-Arm Bilateral Servo Manipulator System (BSM) (Nakamura 2005)

BSM was developed by the Japan Nuclear Cycle Development Institute (JNC) (Nakamura 2005). It was designed for full remote maintenance in a large cell in order to increase both the facility-operation-ratio and reduce the operator's radiation exposure. The BSM is used to repair broken equipment or to replace the equipment and rack inside the cell.

1.7.8 B212 Decommissioning Machine

The B212 decommissioning machine as shown in Figure 1.10 is developed by James Fisher Nuclear Ltd for use for decommissioning applications in the nuclear industry.

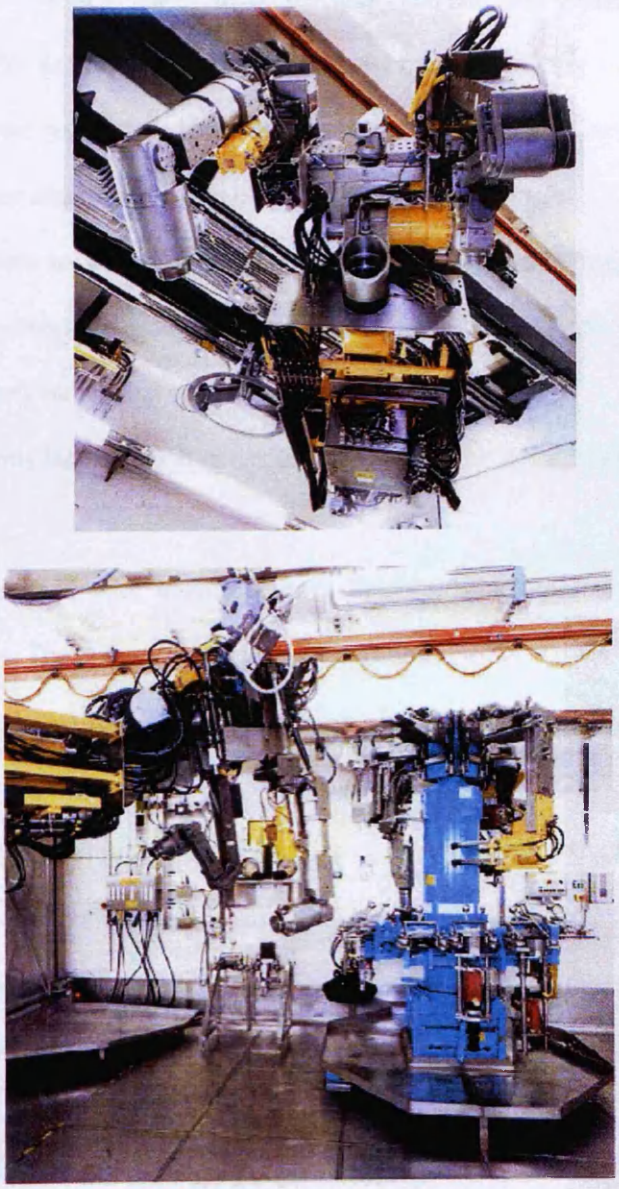


Figure 1.10 B212 decommissioning machine (JFN 2006)

This machine supports two 6-DOF manipulators on the tilt table. The tilt table consists of two station supporting working manipulators, remote manipulator installation and removal, hydraulic rotary actuator and hydraulic and electrical service connections.

There are many design and control limitations in the robotic systems discussed above; consideration of these limitations has influenced the design of MARS-ND. The main limitations can be summarised as follows:

1. Most of the robotic systems are deployed by cranes and can only be used, therefore, if cranes are available and there is sufficient space for manoeuvrability
2. Most of the robotic systems are large and heavy and therefore require a large amount of hydraulic power and cannot be used for tasks in confined spaces
3. The robotic systems use old hardware and software with unfriendly user interfaces
4. Simple control systems are used to control the motion of these robotic systems. These control systems rely on manual control by the operators
5. The control systems lack collision detection and collision avoidance capability

MARS-ND has been designed with unique features in order to attempt to overcome the limitations outlined above. The main new features are as follows:

1. It is operated using two separate systems that are integrated using a universal bracket and a single hydraulic system
2. It is small enough to be used for tasks found in confined spaces and in locations with little space for manoeuvrability
3. It consists of two manipulators that posses a design configuration similar to the human arm in terms of size, scale and dexterity. This facilitates increased ease of control
4. Sophisticated hardware and software are used to operate the multi-arm system
5. A single operating software is used to control the motion of every joint in both arms
6. No crane is required for its mobility as a mobile platform is used
7. Collision detection and collision avoidance are considered and established within the control system
8. A joystick is interfaced and used for the movement of the arm joints allowing control of the movement of the arm

9. A single operator interface is used this creates ease of monitoring and a user friendly system

1.8 Summary and advantages of the use of Multi-Arm robot systems for D&D tasks

Multi-arm robots have many advantages over single-arm robots as has been noted by researchers and decommissioning agencies around the world (Alford & Belyeu 1984; Cox *et al* 1995; Miyabe *et al* 2004). One of the key advantages is that a multi-arm robot system can emulate human arm anatomy and physiology in the sense that one arm can manipulate the work in process while the other manipulates a tool as can be seen in Figure 1.11 on page 20.

The dynamics of single, serial manipulation has been thoroughly investigated over the past decade, and many of the issues related to their control and programming are well understood (Thomas and Tesar 1982). A multi-arm robot system has the ability to perform two distinct operations either simultaneously or separately. Both arms also have the ability to perform the same processing operation in a coordinated manner or to share a task, such as holding and cutting a pipe. Multi-arm robot systems are advantageous over single-arm robot systems in the context of tele-robotic application areas including the manipulation of complex objects, assembly, micro-manipulation, remote space applications, and operation in hazardous environments. Multi-arm robot systems are becoming increasingly popular in the area of industrial automation and space technology. There are many reasons for this, for example multi-arm robot systems can meet the payload capacities and the load balance requirements for particular applications as versatile assembly operations where a single-arm robot system is unable to satisfy these demands. There are also additional benefits such as sharing a heavy load. One of the most important features however, is that in a constrained environment multi-arm robot systems are directly tractable to multi-manual controllers for tele-operation. From a

human perspective advantages of using multi-arm robots within D&D tasks include removing personnel from hazardous environments and reducing the quantity of labour intensive work.

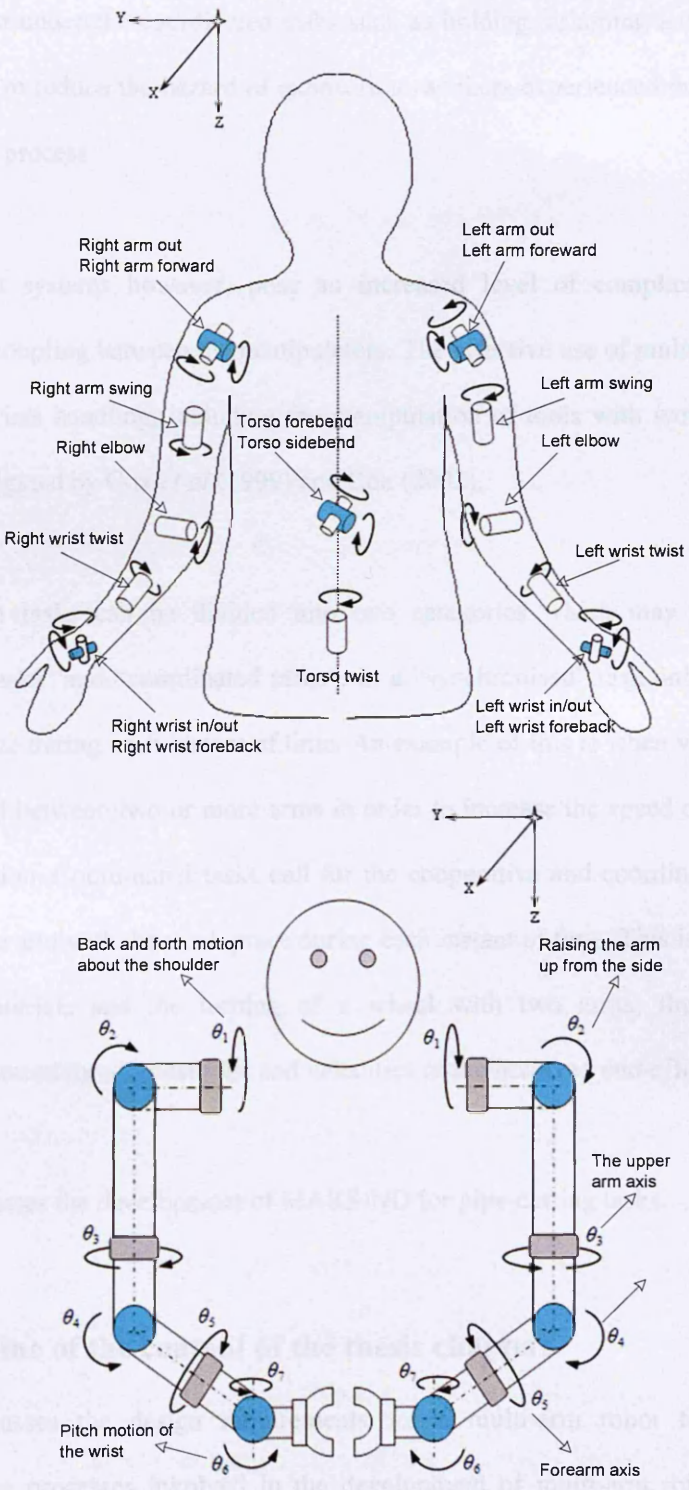


Figure 1.11 Human arms and robot arms (Bakari 2005)

For example, current removal methods for piping systems within the decommissioning process are labour intensive, time consuming, costly and often represent a significant challenge to D&D decision makers. This creates motivation to utilise a remote controlled machine that can undertake coordinated tasks such as holding, crimping and cutting sections of pipe in order to reduce the hazard of exposure to workers experienced during the baseline manual removal process

Multi-arm robot systems however, pose an increased level of complication due to the interaction and coupling between the manipulators. The effective use of multi-arm robotics for hazardous materials handling, including the manipulation of tools with work in process has also been investigated by Cox *et al* (1999) and Cox (2002).

Multi-arm robot tasks can be divided into two categories which may be addressed as "synchronised tasks" and "coordinated tasks". In a "synchronised" task only one arm works on the work piece during each instant of time. An example of this is when various successive tasks are divided between two or more arms in order to increase the speed of operation in an assembly operation. Coordinated tasks call for the cooperative and coordinated involvement of more than one arm with the work piece during each instant of time. This is illustrated in the handling of materials and the turning of a wheel with two arms; these tasks require coordination between forces, positions and velocities of the involved end-effectors.

This thesis discusses the development of MARS-ND for pipe-cutting tasks.

1.9 Brief outline of the content of the thesis chapters

Chapter 2 discusses the design requirements for a multi-arm robot for use in D&D applications. The processes involved in the development of multi-arm robot system using modern tools and their systematic application using engineering principles are also explained.

Finally the chapter outlines examples of robots developed and used for decommissioning tasks in the nuclear industry.

Chapter 3 introduces one of the core aspects of this thesis, the characteristics of modular hardware and operation software for a robot system. This chapter introduces a case study concerning the selection of pre-tested off-the-shelf hardware, its integration in the development of MARS-ND, and justification of the system. It examines the integration and application of the Brokk hydraulic system to operate the Hydro-Lek manipulators separately and together using solenoid valve packs; and considers the sensors used to obtain feedback from the robot joints. The chapter then reflects on the NI hardware adopted for the operation of MARS-ND and their advantages over traditional methods used to control other robot system. Chapter 3 also examines the operational software applied within this research project through its discussion of the control software and hardware interface for the MARS-ND and the controller architecture; the application of PAC hardware, Compact FieldPoint and FieldPoint I/O modules, to facilitate wireless communication between MARS-ND and the user interface in the host PC; and the use of Measurement and Automation Explorer (MAX) and LabVIEW software tools for calibration and the building of user interfaces required for sending and receiving the signals needed to control the robot arm joints accurately.

Chapter 4 outlines the results of structural identification and low-level control of the Hydro-Lek arms. It specifically discusses the use of the Denavit-Hartenberg (D-H) convention as a means of defining the constant parameters of the Hydro-Lek manipulator structure and practical issues arising during the experiments; and use of robotic simulation software with the D-H convention to compute and obtain the transformation matrix T needed to calculate the forward and inverse kinematics of the Hydro-Lek arms. The closed form solution and numerical solution for the inverse kinematics problem are briefly outlined. The application of PID toolkit in LabVIEW for the design of a simple PID controller for the individual arm joints with a potentiometer sensor fitted inside each joint in order to provide a feedback signal to the

controller is explained. The chapter also highlight the problems that were observed with respect to the mathematical complexity of the closed form solution, due to the design of the robot arm; and identifies an alternative solution to the design configuration in order to ease the finding of a closed form solution for the inverse kinematics for the precise control of the Hydro-Lek arm end-effector.

Chapter 5 analyses the role of 3D simulation in the conduction of a given task by an actual robot in the real environment. In this chapter, the research present the use of a simulation technology currently in use called *Workspace 5* robotic simulation software, and its comparison with a new mechatronics tool called NI LabVIEW-SolidWorks Mechatronics toolkit. This toolkit is designed to enable virtual machine prototyping; and the use of electromechanical simulation to develop multi-axis motion profile for the robot arms, detect collisions, and validate them using 3D simulation. A USB joystick is integrated and programmed within LabVIEW software and used with the Mechatronics toolkit in order to control the axis of the robot joints individually.

Chapter 6 demonstrates the kinematics control algorithm developed at Lancaster University for MARS-ND, and discusses its experimental results based on software simulation. This chapter also outlines the tests undertaken at Sugano Laboratory using the algorithm developed at Lancaster University; and the improvements and limitations that became evident through these tests. Finally the chapter compares the results between the two tests in the UK and Japan and gives recommendations for further research and application.

The concluding chapter, chapter 7 assesses the approach adopted in this research for the development of a multi-arm mobile robot system for D&D tasks including the building of hardware, software and control systems; and collision detection and avoidance. It also considers the experimental results obtained and discusses further improvements which could be made for example, improved performance through the use of additional improved hardware

and software including cameras, laser scanning and 3D virtual prototyping. Finally this chapter considers the contribution this thesis has to make to current research in the development of multi-arm mobile robot systems for D&D applications; and recommends areas for further development.

CHAPTER 2

THE DEVELPMENT OF ROBOTICS FOR D&D APPLICATIONS

2.1 Introduction

The UK government and the Nuclear Decommissioning Authority (NDA) are currently involved in D&D of a high number of ageing nuclear power stations and other facilities. These nuclear plants often contain radioactive and other hazardous materials that are harmful to humans. Much of the decommissioning process utilises well established demolition techniques although there are complications within this process due to the risk of exposure to workers and the wider environment from radiation. Nearly all D&D activities that are too hazardous for direct human contact are presently executed using robotics systems, however many of these robotic systems are custom-designed for specific projects and are deployed by a crane rather than a remote vehicle (Richardson et al 1995). The NDA is looking for new and innovative technologies that will allow D&D operations in the UK to be faster, safer, and cost-effective and reduce the radioactivity dose levels to which workers are exposed. The application of a multi-arm robot system within D&D tasks has the potential to meet these requirements.

The use of robotic equipment has become readily accepted for a variety of applications in the field of decommissioning. The initial development of robotic machines in commercial nuclear power focused on applications for situations that presented extreme hazards to personnel and

workers. Robots were developed in the 1980s for remote inspections as well as survey and sampling capabilities in nuclear emergencies. These devices were designed to perform reconnaissance in post accident radiological conditions that would be life threatening to humans. As the domain of robotic automation is widened, hazardous tasks that require human sensor perception, intelligence and dexterity in an unstructured environment are increasingly being considered for robots (Colbaugh and Jamshidi 2007). Modern commercial off-the-shelf tools permit engineers to build sophisticated robotic systems, such as MARS-ND, that can be easily reconfigured for various decommissioning tasks. These tools also possess open architecture environments which can be adapted easily for new applications and allow higher intelligence to be added. Many problems that arose in the past can now be overcome for example, the time spent developing and redeveloping the communication and subsystem interface infrastructure; and limitations placed on future development by the initial system infrastructure. It is therefore now easier and faster to assemble, program and control a robotic system for the rapid development of automated solutions that suit the nature of decommissioning activities.

2.2 Design requirements for a Multi-arm robot for use in D&D tasks

In order to perform a wide range of D&D tasks remote robotic system need to have the following attributes (Noakes *et al* 2002):

- Easily maintained manipulators
- Easy to operate
- Operate correctly first time
- Quick change-outs for the manipulator system
- Operable by one operator from a single location
- Reliable
- Flexible
- Low cost but heavy duty robot system

- Task orientated
- A reconfigurable control architecture to suit different tasks
- Friendly and easy to use user interfaces with which to control the robot system

This research used two 7- DOF arms. A multi-arm robot system that is composed of two 6 to 7 DOF arms with a manoeuvrable base requires system architecture with the following features:

- Standardised high load capacity actuators
- Bus communication among all components
- Standardised interface
- Operating software for all configurations of the system
- Open architecture system controller
- A minimum set of modules to provide repair by easy replacement
- When used in a decommissioning site, wash down capacity to enable decontamination
- When used in a decommissioning site, radiation tolerance for reasonable operational life
- A universal man-machine interface for human intervention and training

The layout shown in Figure 2.1 aims to reveal the requirement document for tasks undertaken by a multi-arm manipulator system. There are three areas that are highlighted for the structuring of any given D&D problem, as follows:

- The task to be carried out
- The tool to be used
- The manipulator requirement

It is known that each D&D problem has a unique identification with regards to the task, tool and manipulator specifications. This common framework however, can lead to a modular,

redundant robotics technology that has on-site reliability sustained by multiple performance characteristics selected and monitored by human operators. Multi-arm robot systems must have effective cooperative manipulation strategies and use of tools and equipment to perform D&D tasks. This is the basis for creating a sophisticated control system suitable for D&D tasks as outlined below.

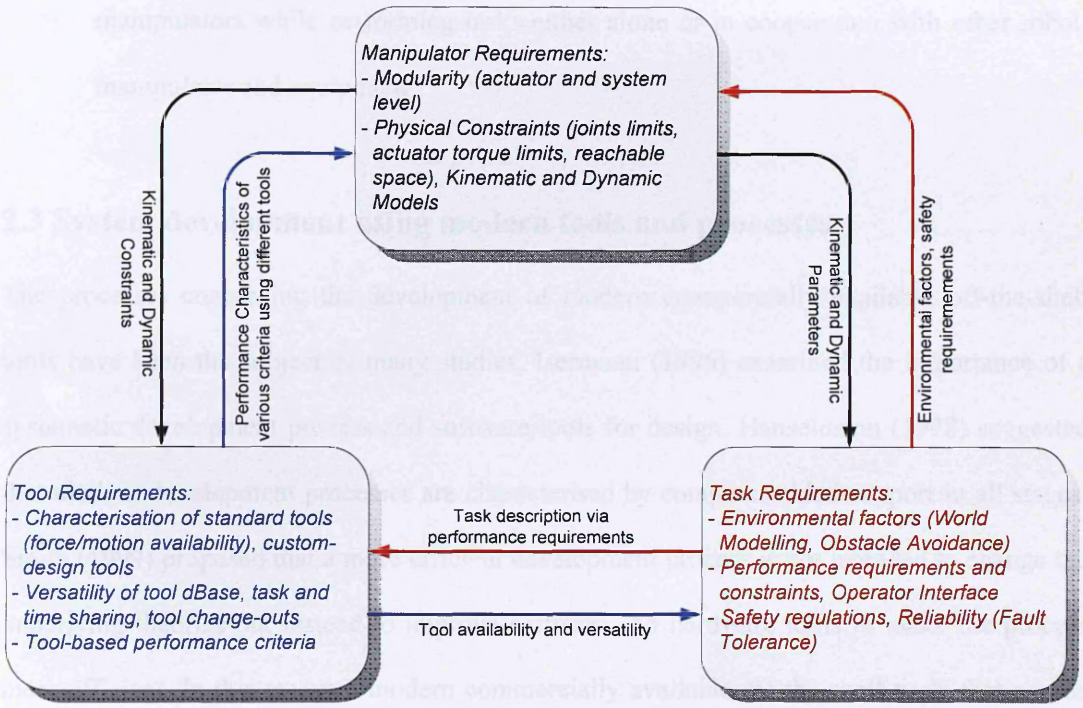


Figure 2.1 Structuring of D&D problems

A reconfigurable multi-arm robot system should ideally have the following operational features:

- Human scale reach and dexterity
- Heavy duty payload
- Force control
- Obstacle avoidance system
- Remote tele-operated control for both arms
- Cooperative manipulator ability
- High reliability
- High maintainability

The successful development of modular robotics at both the system and the actuator levels depends on two technology tasks which are:

1. High performance actuators which drive the joints of the robot manipulator with precision and endurance
2. Operational software that controls manipulator movement of reconfigurable robot manipulators while performing tasks either alone or in cooperation with other robot manipulator and equipment

2.3 System development using modern tools and processes

The processes concerning the development of modern commercially available off-the-shelf tools have been the subject of many studies. Isermann (1996) examined the importance of a systematic development process and software tools for design. Hanselmann (1998) suggested that modern development processes are characterised by computer-aided support in all stages. Smith (1999) proposed that a more efficient development process is not intended to change the underlying theories but instead to improve software and hardware tools to make the process more efficient. In this research modern commercially available off-the-shelf tools that can be reconfigured have been used to build MARS-ND.

The development of robotic systems for ND activities has faced many barriers which have affected the implementation of these systems for decommissioning sites (DOE 2001). One of these barriers can be classified as technological. Technological barriers occur because a decommissioning robot must cope with the complexity of the decommissioning process which may involve an unstructured and continuously evolving site, and multiple tasks with different characteristics and very little repetition. This is in addition to the need for the robotic system to be able to replicate subtle human actions such as handling and dismantling objects, which require the system to have a certain level of intelligence such as an advanced control system, user interface and an intelligent sensory system.

The development of a modern robotic system such as MARS-ND requires:

1. Standardised bus communication among all devices
2. Standardised interfaces
3. Universal operating software for configuration of the system
4. Open architecture system controller

These requirements are discussed in further detail below.

2.3.1 Bus communication among robotic system components

In the past, the control systems applied for robotics in the unstructured environments were old and unreliable. Most of the robotic systems were controlled by a mainframe computer which communicated with a number of sensors, actuators and third party devices (field devices) by point-to-point connections. Mainframe computer-resident processes could communicate with each other through variables shared in a common memory. A centralised architecture of this kind involves various problems which can be summarised as follows:

a. Complex wiring is needed for point-to-point connections to allow the control systems to exchange information with the field devices which can lead to problems in the maintenance and efficiency of the communication system.
b. In order to develop increasingly sophisticated control systems greater processing power are required by the mainframe computers which can lead to high costs.
c. Mainframe computers may contain critical fault points within which a great part of the control activity is concentrated.
d. The closed nature of the system in mainframe computers means that it is impossible to interchange between the various elements, and difficult to upgrade the communication system to meet new control requirements

A mainframe computer system can still be used for simple applications with a limited number of field devices. When centralised systems such as mainframe computers are used in current

practice in industrial plants and the nuclear industry, sophisticated control systems are used in order to control the large number of variables and correlate them by means of computational algorithms. The control functions are also distributed over several processing nodes, each dedicated to a specific part of the control process, and to a group of field devices in order to enable point-to-point communication. The nodes cooperate with each other, communicating through a shared physical channel which generally has a bus topology as shown in Figure 2.2.

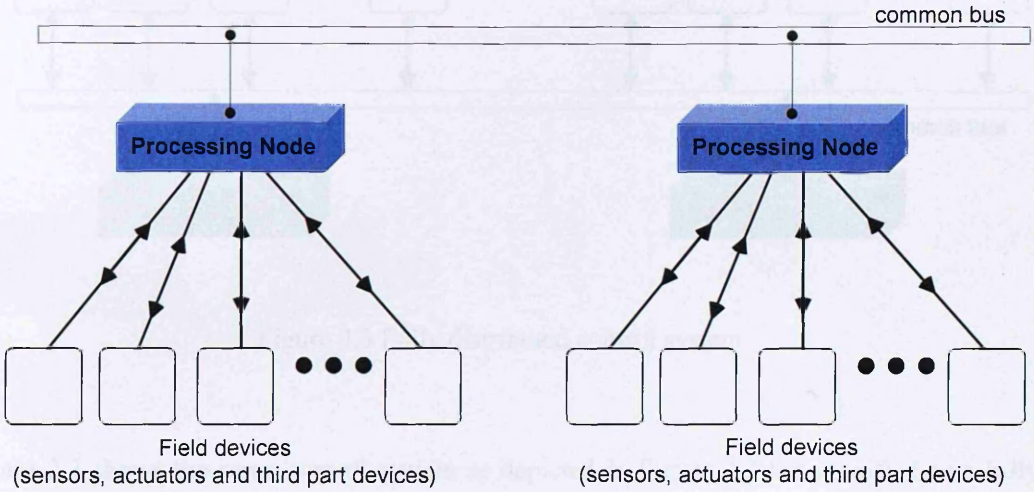


Figure 2.2 Point-to-point control system

The indirect communication in the point-to-point control system shown in Figure 2.2 can cause the following shortcomings:

1. Coding and timing problems
2. Difficulties in distributing information from a sensor to several consumer processes
3. Increased critical wiring

In order to overcome the problems of point-to-point control systems, fully distributed control systems have been developed which allow several control activities to be combined in an integrated environment so as to meet increasingly critical time requirements and specific applications. In the architecture of a fully distributed control system all the elements, both control and field devices, are connected to a single communication channel (Pleinevanx &

Decotignie 1988; Pimental 1990). The field devices are therefore resources shared by all the processing nodes each of which still controls a part of the control process as can be seen in Figure 2.3 below:

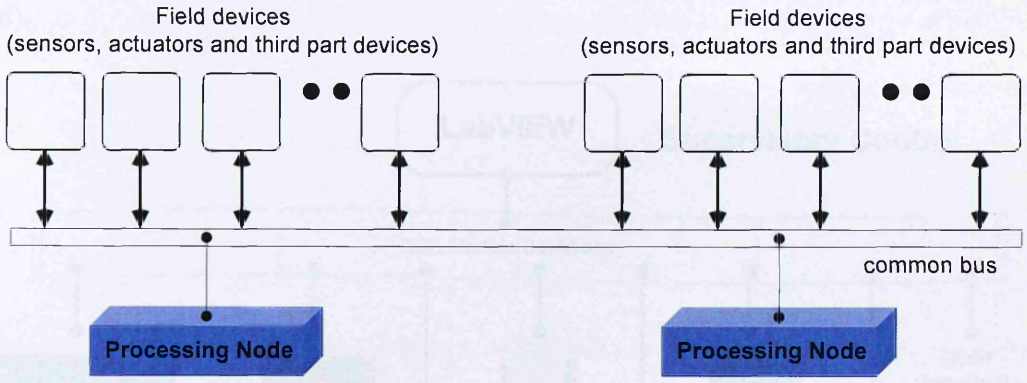


Figure 2.3 Fully distributed control system

Figure 2.3 shows the same control system as depicted in Figure 2.2 but modified as a fully distributed control system. All the information handled by the field devices can be seen as objects which can be reached by every processing node through the communication channel.

In current practice, the level of integration of robotic system components determines the effectiveness of modern technical systems (Mäntylä and Andersin 1998). With respect to technology, integration has three principal constituents: standardisation, automation and rationalisation. A fully distributed control system develops when application of the system occurs alongside functional and modular decomposition. In this way physical and, or logically distributed components that constitute the system facilitate a common goal. Communication between such components in a robotic system is a crucial issue. A key property of modern automated systems is the intensive cross-communication and interaction between elements within the system and their changing environments. Automated systems now comprise of a number of off-the-shelf devices from different commercial outlets that can all be governed by a single controller. Implementing a distributed measurement and control system allows

engineers to optimise the processes running on each machine and over the network, creating a more reliable and higher performance system. LabVIEW is a general-purpose programming system that contains a complete library of built-in elements for open connectivity and system integration. Figure 2.4, shows a distributed system using LabVIEW (National Instruments 2006).

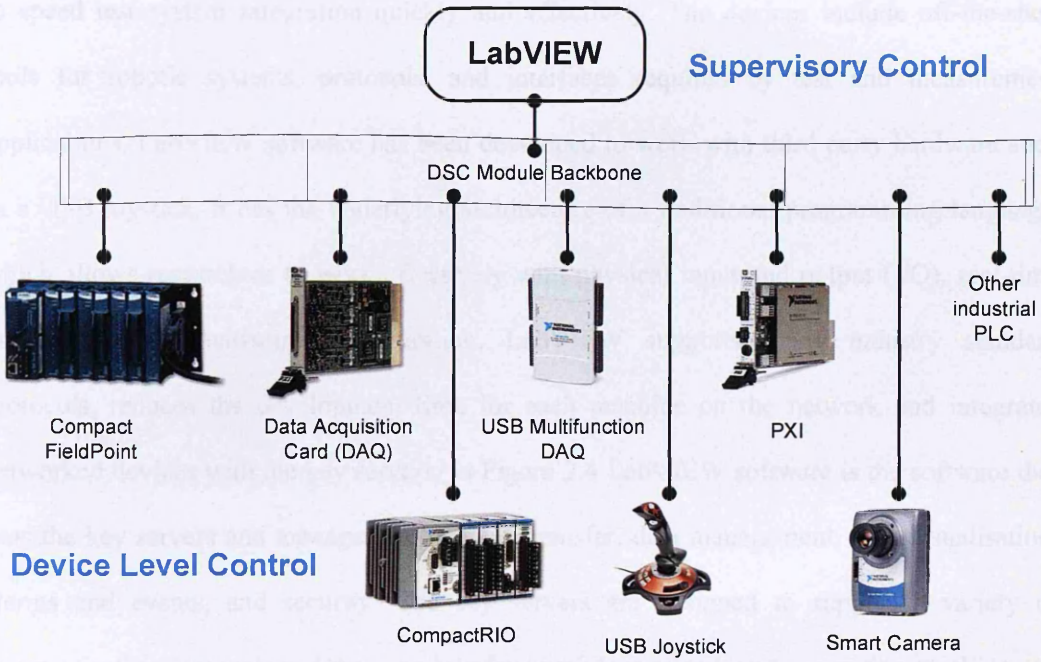


Figure 2.4 Modern distributed system

The distributed system shown in Figure 2.4 can be separated into two parts, the backbone of the system and the nodes. The backbone is the top level of the system and can be simplified to the key servers and the network. The most difficult and time consuming part of building a system is often the integration of components into the backbone. Until recently it was necessary to gather the requirements of each component and identify software tools at the beginning of the process to facilitate the integration process. Tools that are open and flexible are crucial to making this process easier. Open software tools use industry protocols, such as Application Programming Interface (OPC) and Transmission Control Protocol/Internet Protocol (TCP/IP), and work with other third party hardware from a variety of vendors to make integration easier for the end user. One of the key characteristic of the backbone is that

it can communicate with the rest of the hardware through a common protocol, such as TCP/IP. In addition, the individual software used at each machine in the network supports the same communication protocols.

LabVIEW is designed for the development of open connectivity with a broad range of devices to speed test system integration quickly and effectively. The devices include off-the-shelf tools for robotic systems, protocols, and interfaces required by test and measurement applications. LabVIEW software has been developed to work with third party hardware such as a USB joystick. It has the underlying architecture of a traditional programming language, which allows researchers to work effectively with physical input and output (I/O), real-time constraints, and hardware configuration. LabVIEW supports many industry standard protocols, reduces the development time for each machine on the network and integrates networked devices with the key servers. In Figure 2.4 LabVIEW software is the software that runs the key servers and manages the network transfer, data management, data visualisation, alarms and events, and security. The key servers are designed to support a variety of communication protocols and they can interface with legacy and next generation machines on the system. The hardware with specific tasks, such as the robot motion controller, is an example of the node level of the system. With certain hardware components, intelligence can be incorporated at the I/O level. In this PhD research, LabVIEW software has been used as the operating software and the main graphical programming software to interface and build motion control for the MARS-ND system.

2.3.2 Standardised interfaces

It is a challenge to develop a communication interface between all the different tools within a complex robotic system through single operating software. In addition it is necessary to create a graphical user interface where all these tools can be operated in a friendly manner, and simple steps can be used for reconfiguration and modification of the robot to achieve different

aims. A real-time graphical simulation is considered to be one of the most important tools for advanced tele-operation systems because it gives the operator a number of options of operating modes which allow tasks to be achieved with relative ease. Traditional control interfaces for robotic systems have a hard-wired interface for example buttons, switches and dials, then a command-line interface. The operator of such an interface is required to know each and every function, which can become difficult if there are many buttons and switches placed on the same control panel.

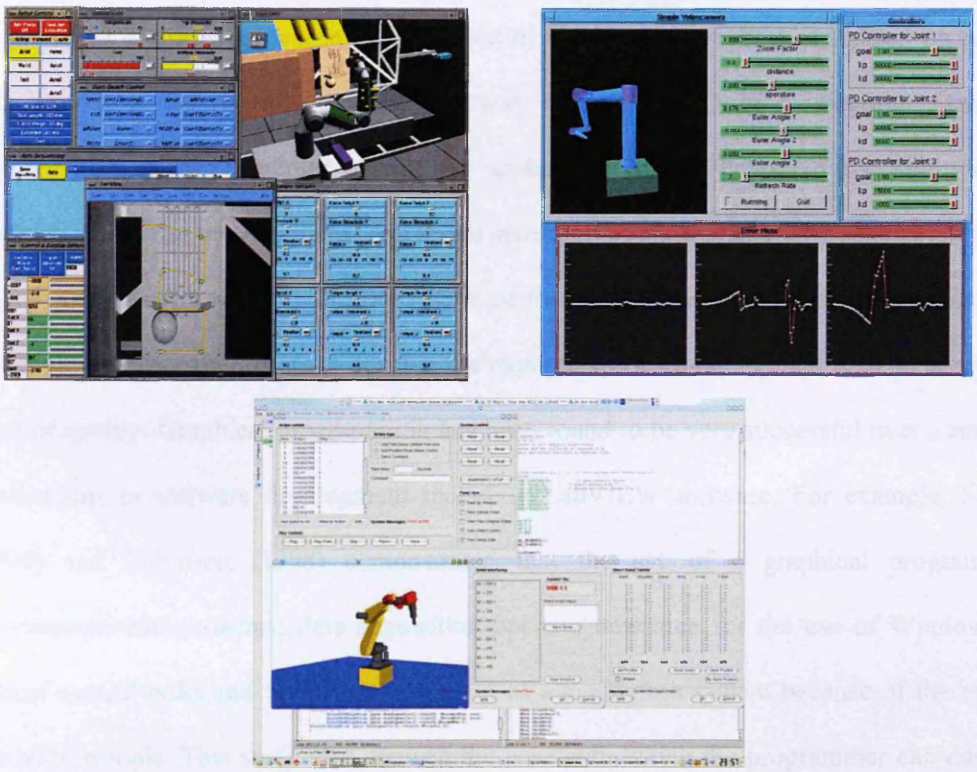


Figure 2.5 Graphical user interfaces

A graphical user interface (GUI) can facilitate a more engaged human interaction with the tools adapted to control the robot system, constrain user inputs to valid ranges and units, supply tubular and plot-based output where needed, and provide users with data from sensors on the robots used for control. The demands of a robotic system mean that the software applied must create an environment that enables modularity, integration and the possibility of reprocessing. In ND advanced tele-operation systems are applied for most of the hazardous and complex applications such as materials handling and pipe cutting. Real-time graphical

simulation is considered to be one of the most important tools for these applications to allow effective communication between the robot system and operator (Bicknell and Hardey 1998). The aim of using advanced graphical user interface software for a tele-operation system is to provide the operator with a variety of operating modes through which tasks can be achieved as easily as possible. Figure 2.5 below shows examples of real-time graphical user interfaces that use graphical programming for the control of robotic manipulators.

Graphical or iconic programming is a method of writing custom programs based on placing and writing a variety of graphical function objects (icons) in a diagram. Graphical programming is very modular in that the application can be broken down into multiple modules which can be constructed and tested separately. This characteristic enables the re-use of any module with other applications. One of the advantages of graphical programming is that it allows non-expert users to develop the required software rapidly and with an acceptable level of quality. Graphical programming has been found to be very successful over a range of applications in software development including LabVIEW software. For example, Shreier (1999) and Robinson (1998) demonstrated that the use of a graphical programming environment with real-time data acquisition opens a new area for the use of Windows for critical control tasks and for the development of a simulation system because of the easy to use built-in tools. This study also showed the ease with which the programmer can design a user interface for a complex system. Similar studies were carried out by Fountain 1999 and Hoadley *et al* 1998. Baroth and Hartsough 1995 demonstrated that the use of Graphical Programming to accomplish a programming task requires up to 10 times less time than the time required for textual programming.

2.3.3 Universal operating software

Until recently robot joints were controlled in real-time at high sample rates by embedding control loops in customised hardware. For example the PUMA 560 robot controller has two

customised boards for Proportional Integral Derivative (PID) control of each DOF using 12 boards in total (Edward *et al* 1992). In current practice many robotic companies and researchers are able to move the control of the robot from hardware to software control which creates the following advantages:

1. Lower costs as they can run on general purpose computers instead of specialised hardware
2. Increased flexibility because control algorithms can be modified
3. More powerful control algorithms can be implemented
4. Allow upgrading to more powerful computers without changing software

Universal operating software is an operating system that manages computer resources and provides programmers with an interface to access those resources. The operating system performs basic tasks such as controlling and allocating memory, prioritising system requests, controlling input and output devices, facilitating computer networking and managing files.

LabVIEW was chosen as the universal operating software for this research because of its connectivity with a broad range of devices, protocol, and interfaces required for measurement applications. It contains libraries of functions and development tools specifically designed for data acquisition, instrument control and automation control. LabVIEW also uses Virtual Instruments (VI) for its graphical programming through which the object in graphical programming simulates the actual instruments. VI has been useful in this research because it allowed the operator to use the LabVIEW libraries which contain all the necessary icons needed to build the control model for the MARS-ND. This has eliminated the need for expertise in programming language such as C++. An example of LabVIEW's VI layout is shown in Figure 2.6.

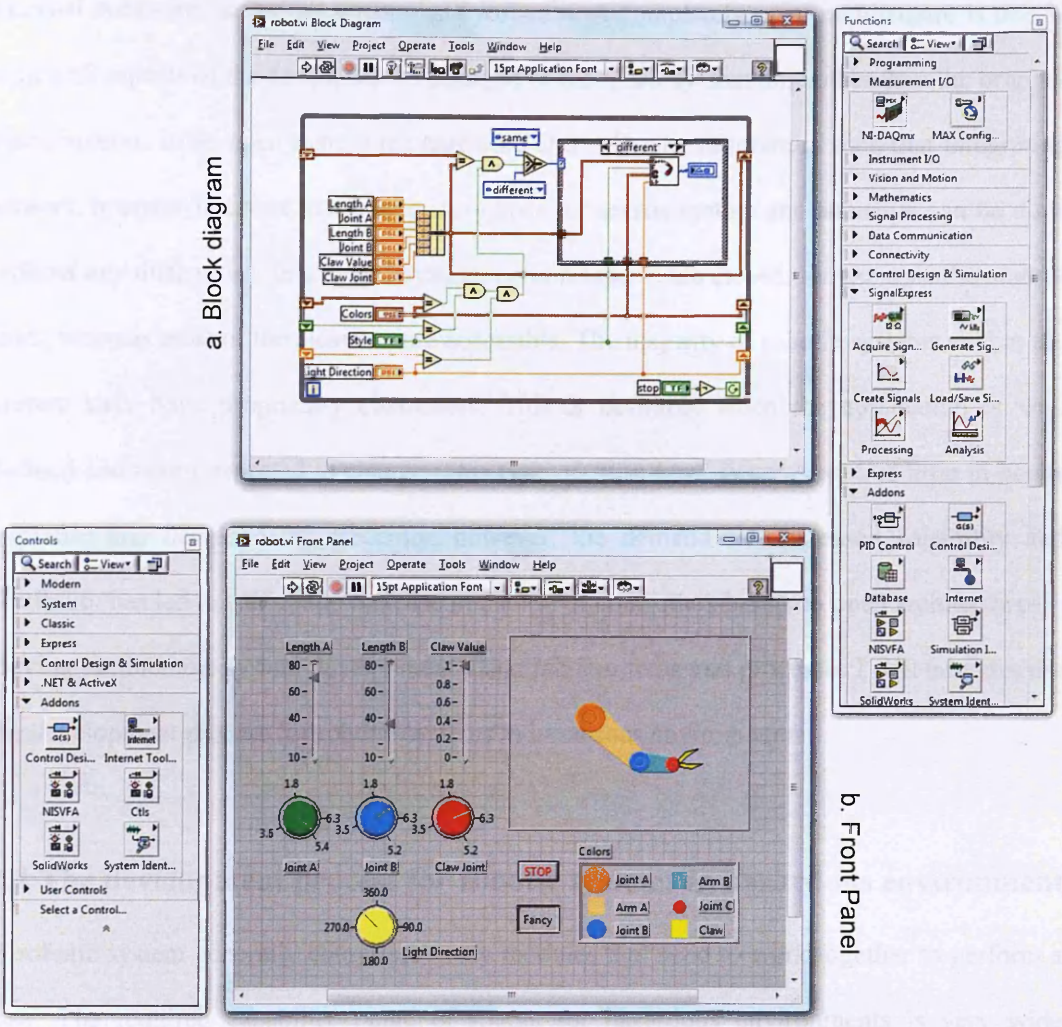


Figure 2.6 LabVIEW VIs (NI LabVIEW 2005)

The VI structure consists of: *Front Panel*, which is an interactive user interface for data entry and output visualisation. *Block diagram*, which represents the source code for the application and consists of icons connected together by wires through which the data flows. The iconic functions in the block diagram can be other VIs, called sub VIs.

2.3.4 Open architecture system controller

Robot controllers can be classified into three main categories: proprietary (also called closed), open, and hybrid systems. A proprietary system is one in which it is very difficult to integrate

external hardware, including sensors and software. A completely open architecture is one in which all aspects of the design can be changed or modified by someone other than the original manufacturer. In an open system the hardware and software structure is such that integrating sensors, operator interface and new control laws for servos motors and actuators can be done without any difficulties. In a hybrid system, certain aspects are closed, for example the control laws, whereas most of the elements are accessible. The majority of robotic systems sold at the current time have proprietary controllers. This is desirable when the application is well defined and is not expected to change. This type of "turn-key" system requires little in-house expertise and development. Recently, however, the demand for increased capability and flexibility has led to a dramatic increase in the use of controllers based on open architectures. Having outlined the system development using modern tools and processes I will now discuss the development process for robotic systems in hazardous environments.

2.4 The development process for robotic systems in hazardous environment

A robotic system generally comprises many modules that need to work together to perform a task. The required capability range of robots for hazardous environments is very wide depending upon factors such as the nature of the task, the degree of structure in the environment and the level of hazards. The development of advanced technology requires both a study of economic feasibility and an assessment of available technology. The identification of the required capabilities that a robotic system should have is not an easy task and it requires the use of a systematic approach that enables the developer to gather concrete information to help satisfy the end user requirements.

The adoption of a systematic approach in the development of any system requires the use of an appropriate development model, which takes into consideration the unique nature of the particular tasks. The many modules of a robotic system for decommissioning make it a multidisciplinary system in which the selection of a development model based on only one

module is not appropriate (Zied, K. 2004). Seward (1999) has noted that an important aspect of the development process is the difference between the development of a system for research purposes and a system for commercial purpose. The latter has well-defined end results which allow the formulation of specific requirements, but the former is open-ended which means that it is difficult to identify specific requirements. It therefore becomes necessary to identify a starting point and an ending point for the research project.

The use of systems engineering (SE) principles allows a research project to identify the user and the system requirements in a general manner, and create architectures for future research. This permits different teams to collaborate even in different time periods. The adoption of a development model such as the SE Model (Stevens, R., *et al* 1998) allows for the implementation of a partially developed system and supports the continuity of the research project. In a commercial project, the same principles can be applied but, as mentioned above, it is necessary to define the endpoint of the project. Stevens presents in detail the development process using the SE principles. All of the models identified are based on the basic sequential model. The sequential model steps are adopted to tailor a development model that suits the development of construction robotic systems.

2.4.1 Implementation of a partially developed system in a new system

The SE process is designed to cope with complex systems; part of this complexity can come from the use of partially developed systems. It is necessary to emphasise the difference between legacy systems and partially developed systems. Legacy systems, as identified by Stevens, R., *et al* (1998), have been developed often without good documentation and are already in use as final products. Partially developed systems are systems whose development has been stopped for unknown reasons. The partially developed system can be documented and re-engineered for further development in terms of SE principles. Figure 2.7 shows the implementation of a partially developed system in different stages of the development process.

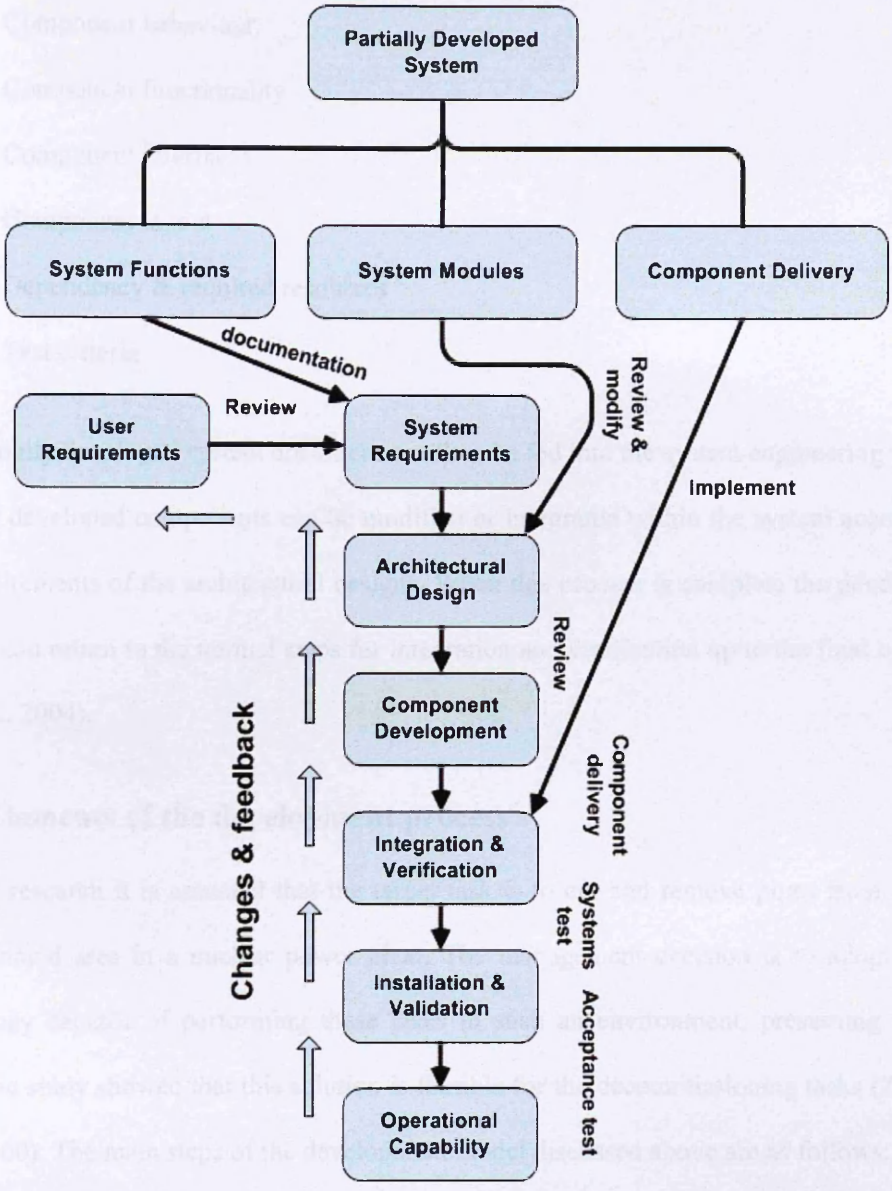


Figure 2.7 The use of a partially developed system in the development of a system using systems engineering principles

In order to identify the system functions it is necessary to perform a complete analysis of the partially developed system, system architecture and the system components. The output from this analysis is a document that describes the system objectives and component functionality. It thus becomes possible to recall the format of the architectural design document as a model for the output of the partially developed system analysis in terms of:

1. Component behaviour
2. Component functionality
3. Component interfaces
4. Component layout
5. Dependency & required resources
6. Test criteria

The partially developed system document can then be fed into the system engineering process. Already developed components can be modified or integrated within the system according to the requirements of the architectural designs. When this process is complete the development process can return to the normal steps for integration and verification up to the final operation (Zied, K. 2004).

2.4.2 Elements of the development process

For this research it is assumed that the target task is to cut and remove pipes from inside a contaminated area in a nuclear power plant. The management decision is to adopt robotic technology capable of performing these tasks in such an environment, presuming that the economic study showed that this solution is feasible for the decommissioning tasks (Zied, K., *et al.* 2000). The main steps of the development model discussed above are as follows:

2.4.2.1 User requirements

The user requirement capture process can be employed to produce the User Requirement Document (URD). The environment variables discussed by Bahr, N. (1997) can be represented pictorially to identify the working environment and all adjacent subsystems involved in the traditional methods and robotic solution (See Figures 2.8a and 2.8b). Figure 2.8a explains the interaction between people and their working environment. Figure 2.8b explains the interaction between hardware systems such as robotic system and its working environment.

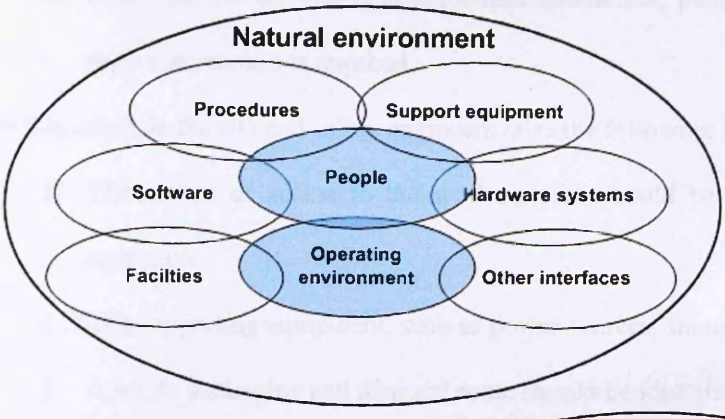


Figure 2.8 (a)

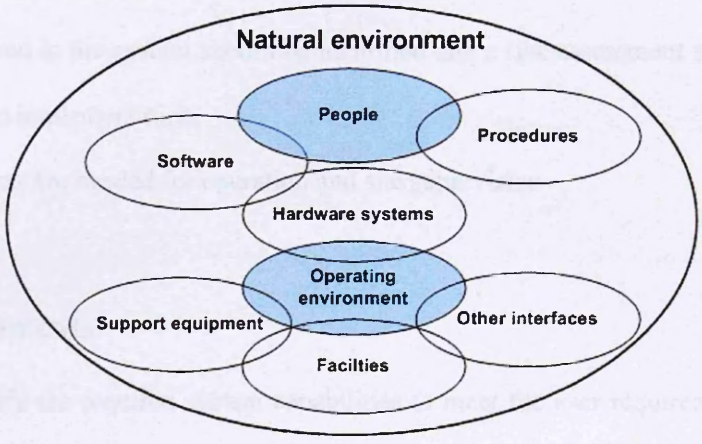


Figure 2.8 (b)

Figure 2.8 Environmental variable: (a) Traditional method and (b) Robotic solution (adapted from Bahr, N. 1997)

This process helps to define the user requirements and the system requirements for the new system. In the following example, the client (the contractor who wishes to introduce the new technology) raises the following six issues:

1. The site is highly contaminated and requires the avoidance of direct human intervention
2. Pipe cutting should be as quick as possible with reasonable accuracy using existing saws and other supporting equipment
3. The system should be easily decontaminated following completion of the task
4. The cost of the new system should be within the agreed limit
5. A user interface should be provided together with information concerning operation and safety

6. A compatible off-line system for task simulation, planning, scheduling, costing and report generation is required

In this example the site and safety engineers raise the following issues:

1. The means of access to the working area should be identified prior to the actual operation
2. The supporting equipment, such as power sources, should be identified
3. A waste packaging and disposal route should be identified
4. The hazards involved in the system should be identified and a risk assessment should be prepared prior to implementation.
5. Surveillance cameras are needed for operation and site supervision

2.4.2.2 System requirements

It is now possible to identify the required system capabilities to meet the user requirements. The user requirements can be translated into system requirements which can be defined in terms of different functions. The main system is divided into two subsystems, the first is the off-line system, and the second is the on-board system.

The main functions of the off-line system are:

a) To receive the contract information
b) To create simulations for the system components and the working site
c) To identify tasks
d) To plan tasks
e) To prepare task schedules
f) To evaluate the contract costs
g) To generate a descriptive report for the whole contract including prices
h) To review the resources required
i) To review safety issues, codes of practice and regulations

j) To arrange for tools and logistics dispatch

k) To issue working orders

The on-board system function is divided into six functions; these functions collaborate to perform the overall function. There are interdependencies between the systems involved in undertaking these separate functions. These functions can be decomposed as follows:

1. Handling input and output from and within the system
    a. Receiving work instruction and task details from the off-line system
    b. Sending and receiving information from and to different components of the system
    c. Monitoring the performance of the individual system components and assuring harmony
    d. Task monitoring and task sequence control

2. Handling platform position
    a. Selecting the platform
    b. Moving the platform to a desired position
    c. Providing a stable position for the platform
    d. Interacting with other functions
    e. Logging of all operations to enable full traceability of the process

3. Handling environment information
    a. Monitoring the working area
    b. Checking for collisions in the working area
    c. Perceiving relative position and orientations with regard to the working area
    d. Issuing safety warnings
    e. Interacting with other functions

4. Handling motion
    a. Receiving position information
    b. Sending signals for position modification
    c. Rectifying position
    d. Path planning
    e. Interacting with other functions

5. Handle end-effector position
    a. Receiving commands
    b. Handling tools
    c. Providing a stable platform for tools
    d. Providing resources for other functions
    e. Providing desired configurations
    f. Interacting with other functions

6. Handling tools
    a. Providing multiple tool docks
    b. Providing a stable platform for tools
    c. Providing easy engagement and disengagements of tools
    d. Sending tool status

2.4.2.3 Architecture design

The architectural design concepts can be represented in three forms, all of which give increased understanding of the system under development. The first form is the system structure which defines the major component organisation and decomposition. The second form is the system behaviour which defines the inherent dynamics in the system and shows how the system will behave during operation and the system layout. Finally the system layout

defines the physical interrelationship between the system components and their relative positions.

(a) System structure

For a robotic system it is possible to define the major components based on the functional decomposition presented in the system requirements. It is however difficult to make a relative structural decomposition or a hierarchy of the major components because all of them are distinctive and cannot be subsystems of each other. The decomposition of each component can be identified based on the sub-functions involved in the main function of the component.

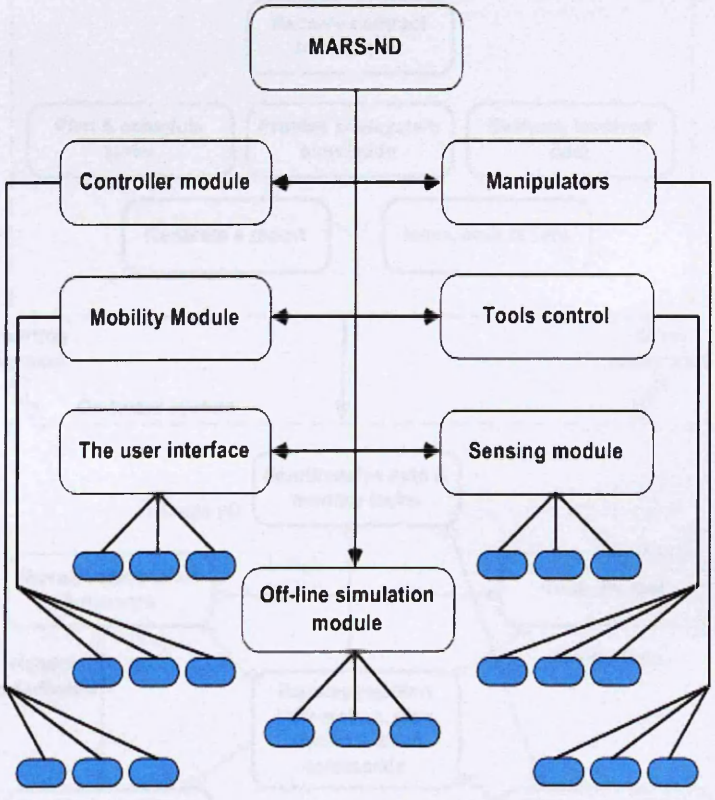


Figure 2.9 MARS-ND structure

Figure 2.9 shows the major components of a robotic system structure which include: off-line simulation module, the user interface, the controller module, the sensing module and the manipulator & end-effector module.

(b) System behaviour

The system behaviour model shown in Figure 2.10 illustrates the top-level behaviour of the major components of the system this includes handling the plan & schedule tasks, the costs involved for the whole project, and the layout of the on board system. Clearer system behaviours may be illustrated by lower level subsystems for example communication behaviour in the controller module, where data is exchanged within the subsystem itself as well as between other subsystems in the architecture. Figure 2.11 shows an example of the controller module structure and hierarchy. Figure 2.12 shows a behavioural model of the controller module

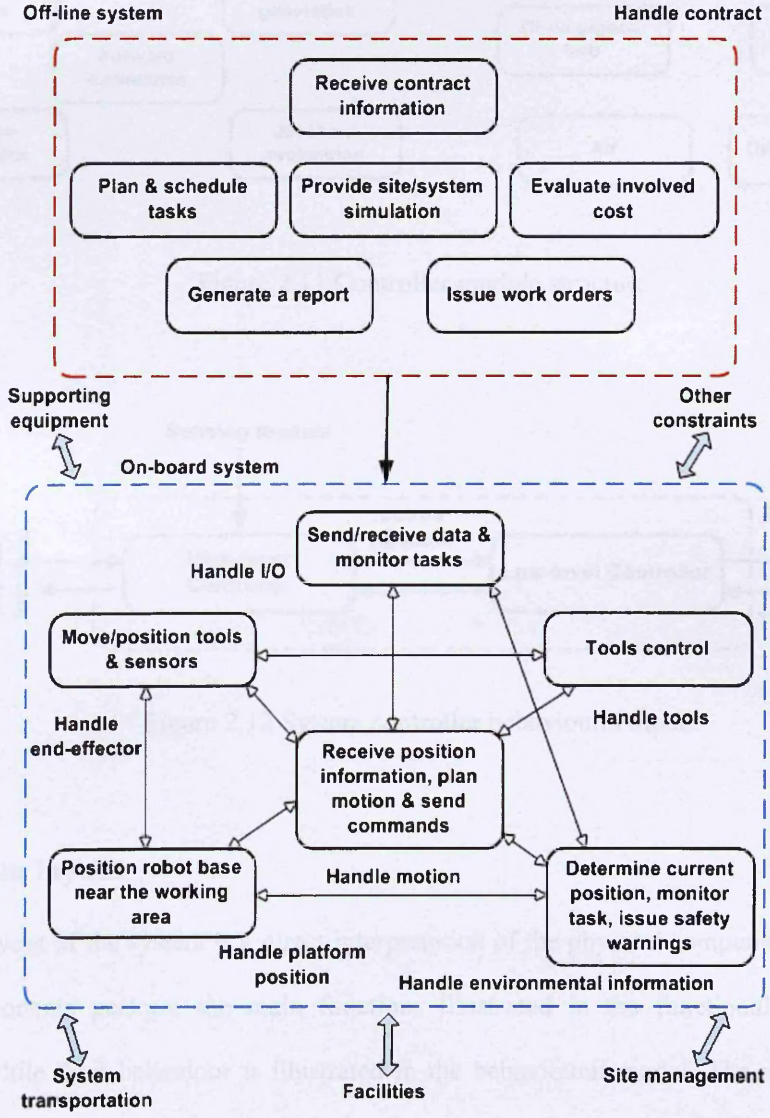


Figure 2.10 System top level behaviour model

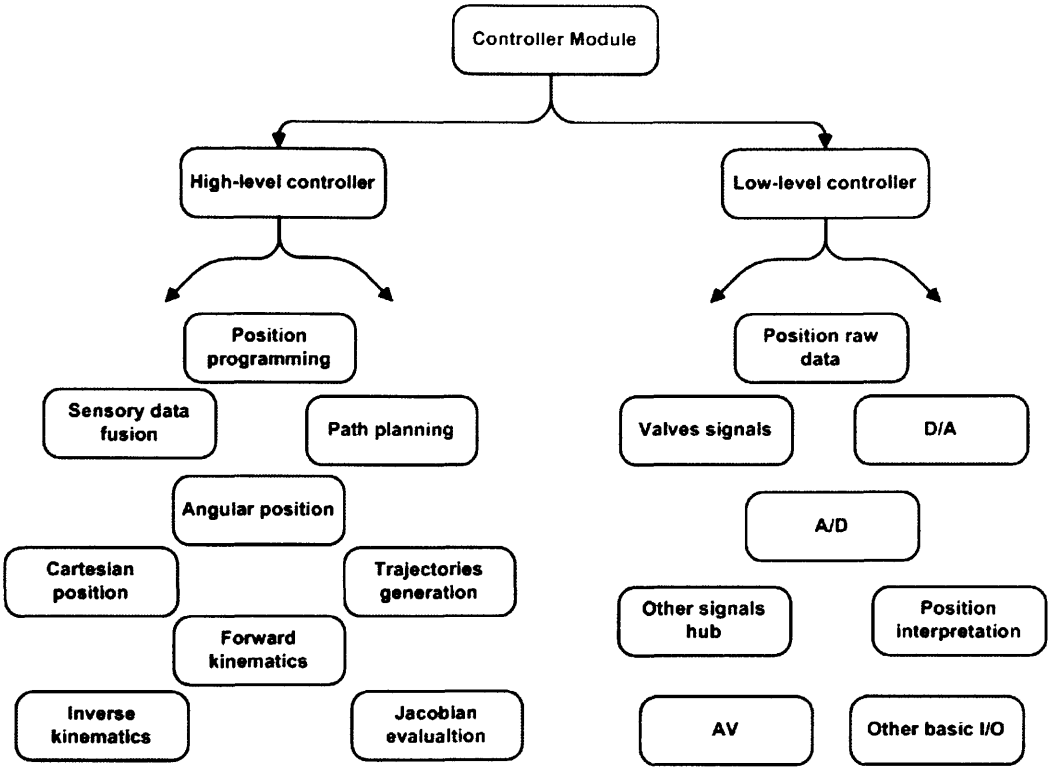


Figure 2.11 Controller module structure

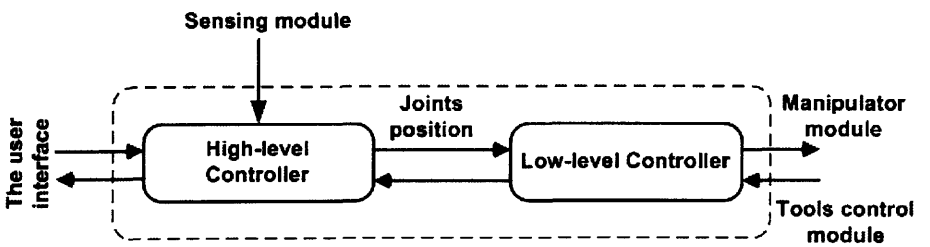


Figure 2.12 System controller behavioural model

(c) System layout

The top layout of the system is a direct interpretation of the physical component arrangement. The components perform the main functions illustrated in the functional decomposition diagram while their behaviour is illustrated in the behavioural model. The on-board system layout as represented in Figure 2.13 gives details of the final working system. The major components shown in Figure 2.13 are the six modules already identified in the behavioural

module but with an emphasis on the components used in the design. The initial aim that underpinned the building of MARS-ND was to reproduce human capability in a robotic system, in terms of movement, range and forces, to allow that system to undertake anything that a human can do.

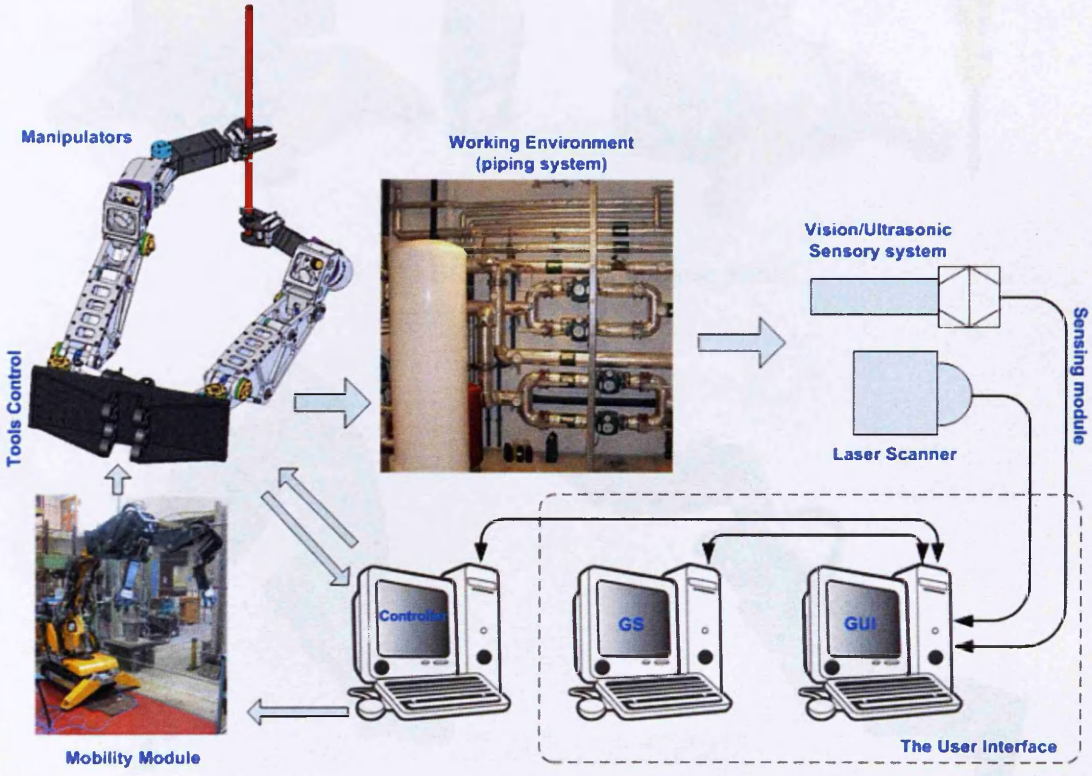


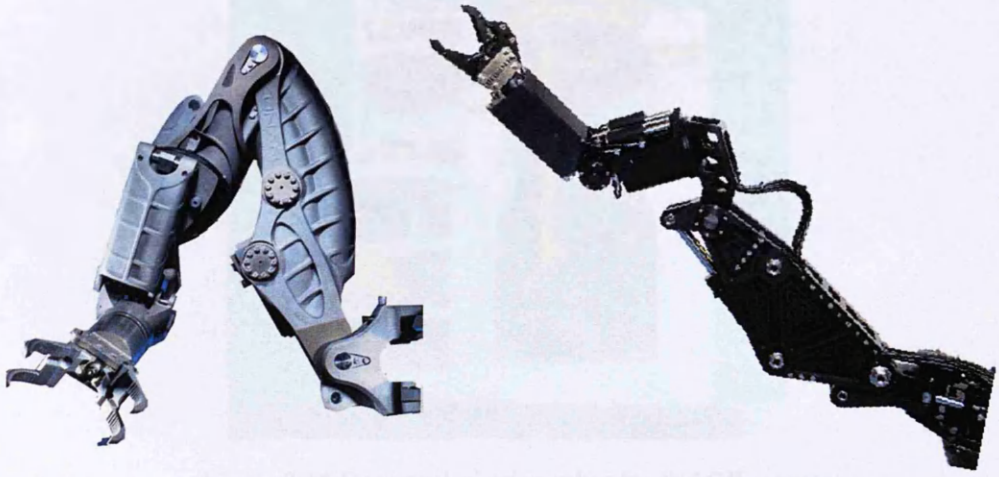
Figure 2.13 MARS-ND layout

2.5 The type of robots developed for decommissioning

Current automated systems employ virtually no autonomy or programmed motion; invariably there is a human in the control loop and this is expected to continue for many years to come. Consequently all systems employ remote control, tele-operation or master-slave manipulation. Current systems tend to fall into one of the four following categories:

1. Relatively expensive customised solutions to specific problems
2. General purpose plant as shown in Figure 2.14
3. Systems fabricated from off-the-shelf components as shown in Figure 2.15
4. Automated process plant for packaging and waste processing

Figure 2.14 BROKK general purpose plants



Schilling Hydro-Lek

Figure 2.15 Off-the-shelf manipulators

At Windscale Advanced Gas Cooled Reactor (WAGR), a customised solution was used for the (DECON) demonstrator project (Benest and Wise 2002). DECON is one of the recognised decommissioning strategies. Within DECON everything is decontaminated to a level that allows removal of regulatory control shortly after shutdown of operations. Residual waste is treated, packaged and removed for disposal; no benefit is derived from waiting for additional decay of radioactivity. Figure 2.16 shows the customised solution system used for DECON. It consisted of an extendable mast with a 6-DOF manipulator at the end. Waste material was transported out of the reactor containment vessel by overhead gantry crane and finally

lowered through the floor into concrete storage vessels for disposal. The floor over the reactor was filled with lead shot to protect the workers above. Dose rate was kept to a total of 17 mSv per worker over the six year of the project. It produced 22 tonnes of Low Level Waste and 10 tonnes of Intermediate Level Waste. The total project cost £80m, with the automated handling system alone costing about £8m.

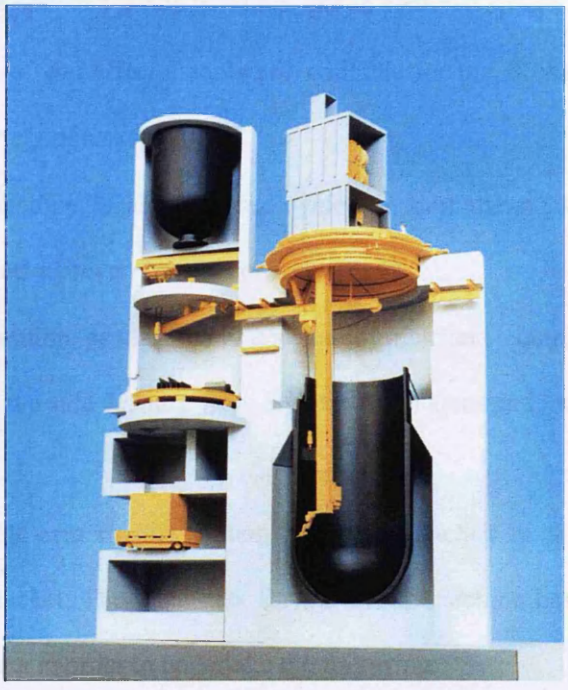


Figure 2.16 Decommissioning robot for WAGR reactor

There are advantages and disadvantages to DECON. The advantages are that it quickly frees the site and allows some of the workforce to be retrained for decommissioning. The disadvantages are that more waste is produced and workers are exposed to a greater radiation hazard. Automation and robotics therefore have a significant role to play within this process. If the reactor vessel at WAGR had been demolished immediately, dose exposure would have been 1 Sv/hr, a worker would therefore have reached their annual dose rate in 20 minutes.

2.6 Current Brokk machines used for decommissioning applications

Brokk machines are primarily designed for demolition tasks and are already used within the nuclear decommissioning industry for demolition and other tasks. Examples of BROKK

general purpose machines are shown in Figure 2.14. The Brokk machine is suitable for adaptation for specific nuclear decommissioning tasks such as pipe cutting and the removal of identified parts for the following reasons:

- The Brokk machine is both tough and small allowing it to be used for demolition tasks and to be operated inside buildings and in confined spaces
- A wide range of end-effector tools are available for the Brokk machine in order to carry out a variety of tasks
- The Brokk machine is already used widely within the nuclear decommissioning industry so it has a proven track record
- A remote operation pendant containing a joystick and control buttons allows the operator to be at a safe distance from high radiation areas and hazardous debris

In this research a multi-arm robotic system has been attached to the Brokk manipulator through a universal bracket. The main task for the Brokk machine has been modified from demolition to D&D tasks in order to undertake the following:

1. To carry and transport the multi-arm robot system when the multi-arm robot system needs to be used for decommissioning tasks
2. To position the multi-arm robot system by lifting or lowering it to the right position needed for the multi-arm robot system to undertake the required task
3. The Brokk hydraulic system is used to power hydraulically the multi-arm robot system

Figure 2.17 shows a selection of modified Brokk machines used for decommissioning applications in the UK with a single arm robot system. Because of the classified nature of this work it was not possible to obtain further information regarding their specification.

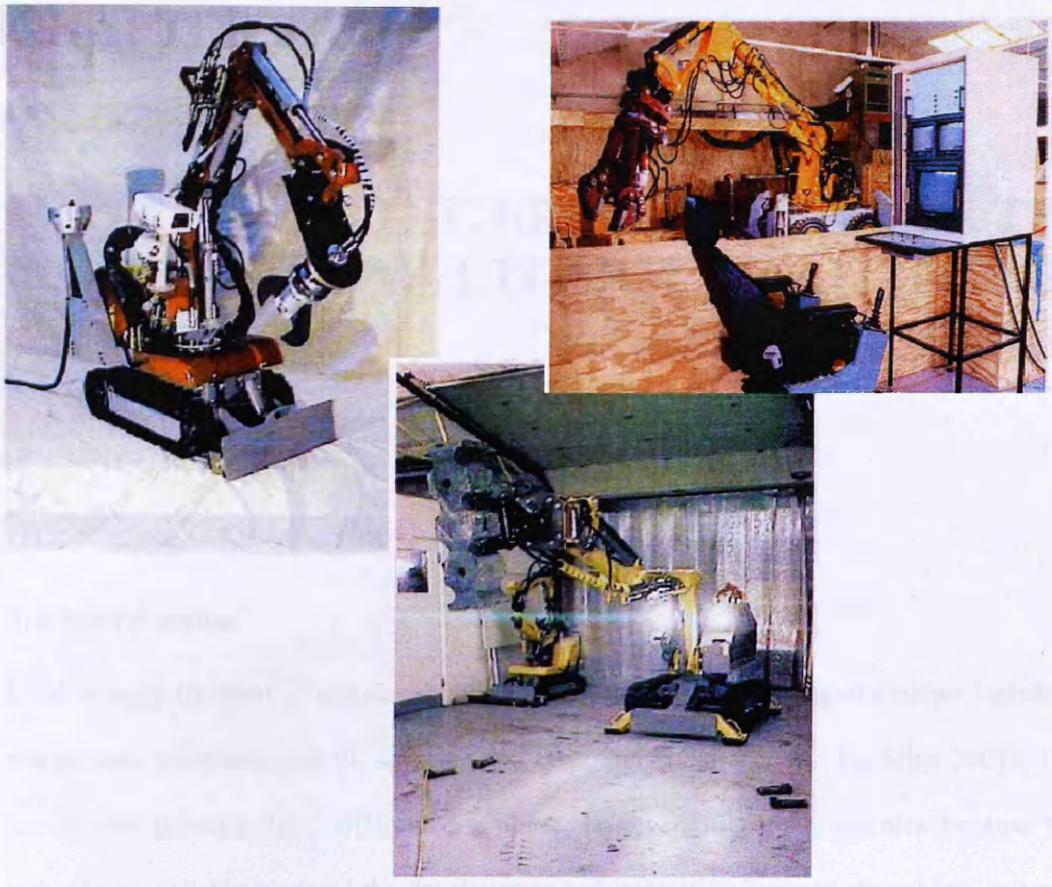


Figure 2.17 Modified Brokk Machines for decommissioning applications (JFN 2006)

2.7 Conclusion

This chapter has discussed the design requirements for a multi-arm robot system for use in D&D tasks. It explains the development of the multi-arm robot system using modern tools and processes, and demonstrates the systematic approach in the development of the robot system using system engineering principles. Finally this chapter has given examples of the types of robotic systems currently used for decommissioning tasks in the nuclear industry, including some of the Brokk machines presently used in the UK for decommissioning applications. Chapter 3 discusses in detail the development of MARS-ND and compares the traditional and modern paths used for the development of this robotic system.

CHAPTER 3

THE DEVELOPMENT OF THE MULTI-ARM ROBOT SYSTEM

3.1 Introduction

Until recently the field of robotics drew on a multitude of engineering disciplines including mechanical, electrical, control, software and computing (Clarence W. De Silva 2007). The interactions between these different disciplines, however, was quite complex because the technology available hindered the development and interaction between the robotic systems, and the control of these systems by a single operator. The requirements of a robotic system as given below illustrate their interdisciplinary nature and the need for easy communication among the various engineers involved in their design:

- The geometry of the manipulator must allow its tool to be positioned along the path
- The required positions for the servos that drive each joint of the manipulator must be generated in real time usually by computer
- The servo system must be capable of responding to the required positions in a smooth and timely manner
- The joint actuators must be sized to provide the torques needed as the arm moves
- The feedback transducers must have appropriate resolution to allow the servos to control the joint positions within a defined error

- The mechanical system itself must meet specific predefined degrees of stiffness, accuracy and repeatability
- There must be a suitable user interface as well as an interface to production control computers

3.2 Traditional development path

In the traditional path for developing a robot system each engineer has specific and defined roles.

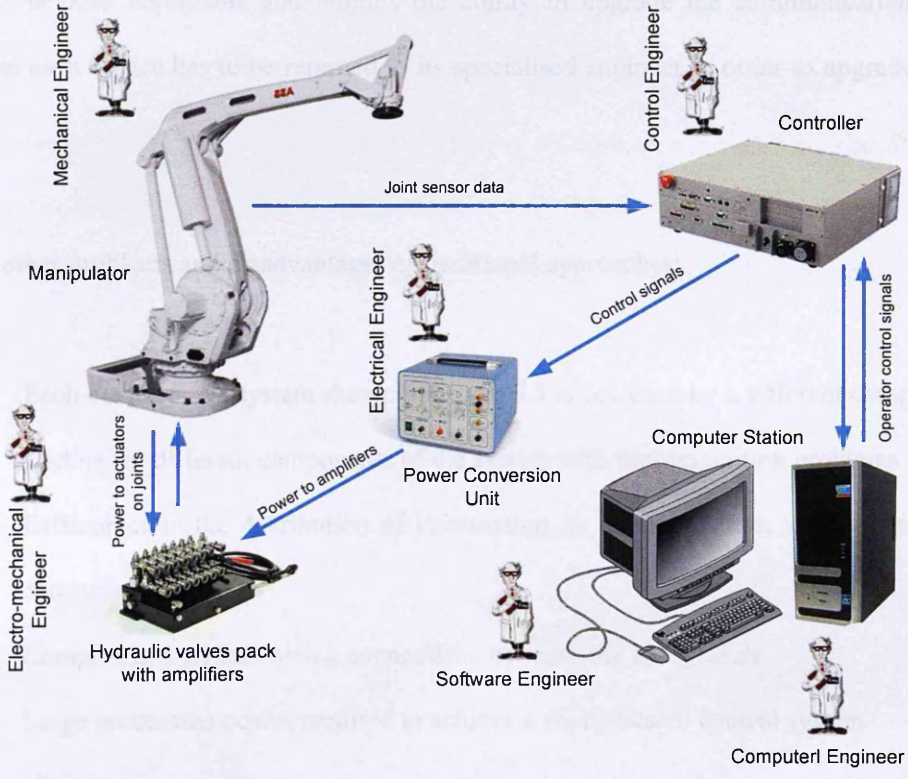


Figure 3.1 Traditional paths for developing a robotic system (Bakari 2008)

For example, mechanical engineers design the robot structure and its joint mechanisms, bearing and heat transfer characteristics; electrical engineers design the robot's control electronics, power amplifiers and signal conditioning; electro-mechanical engineers may work on the robot's sensors; computer engineers design the robot's computing hardware; control engineers design the motion algorithms for every motion of the robot joints and the open or

closed loop control required for the robot system; and software engineers design the software or the programming code needed to produce the signal outputs for the robot's joints, and read the signal inputs from the robot's sensors as well as the user interface. Figure 3.1 demonstrates these roles.

The main computer used to communicate with the traditional robotic devices shown in Figure 3.1 utilises point-to-point connections, as explained in chapter two. A computer based on point-to-point connections (Salvatore *et al* 1995) makes the interchange ability among the various devices impossible and inhibits the ability to upgrade the communication system because each device has to be returned to its specialised engineer in order to upgrade and fix faults.

Some other problems and disadvantages of traditional approaches:

1. Each device or subsystem shown in Figure 3.1 is designed by a different company
2. Coding for different components of the system with time-execution problems
3. Difficulties in the distribution of information for example, from a sensor to several consumer processes
4. Complex and critical wiring connections between the subsystems
5. Large processing power required to achieve a sophisticated control system
6. High costs
7. The presence of critical fault points may jeopardise the functioning of large control blocks if they malfunction
8. The closed nature of the system makes interchange ability among the various elements of the control system very difficult
9. Difficult to upgrade the communication system to meet new control requirements
10. Difficult to modify the controller when intelligence is added to the system in order to suit the nature of that task

Robotics and automation challenge traditional engineering disciplines because of the need for an integrated approach toward the selection of devices in order to meet the intended function. This process intrinsically requires team activity and the crossing of boundaries between conventional engineering disciplines. Robotics and automation are still emerging areas of engineering and require the integration of the essential elements of mechanical engineering, electrical engineering, and computer science.

3.3 Modern development path

One of the key issues in the development of modern robotic systems is the Mechatronic approach (A. Ollero *et al* 2006). The modern development path of robotics requires engineers to be able to move between different engineering disciplines and to be able to apply and integrate both theory and practice. The emergence of Mechatronics facilitates this process. Mechatronics is a science which combines mechanics, electronics, control and computer science, as illustrated in Figure 3.2.

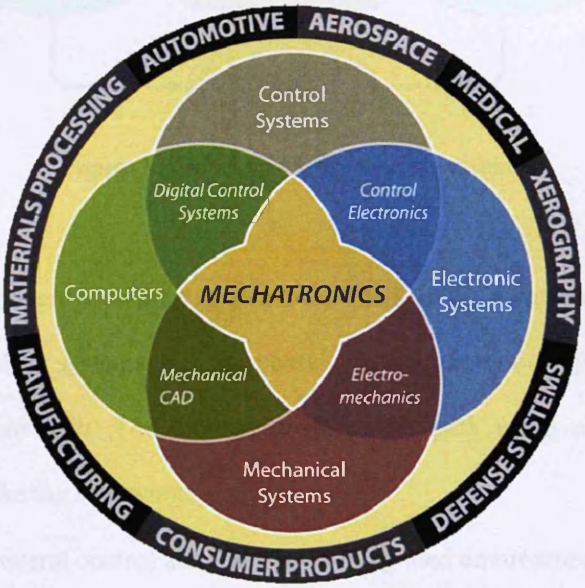


Figure 3.2 Mechatronics (Aerial Venn diagram 2007)

Mechatronics applies the latest, cost effective technology in the areas of computers, electronics, controls, and physical systems to the design process to create products that are

more functional and adaptable. Mechatronic design deals with electronically controlled mechanical devices in such a way that the distribution of functions between mechanical and electronic hardware, and software components within the devices takes into account the overall performance of the system rather than the separate elements (Rolf Isermann 1996). It also considers the possibility of future modifications. The structure of such a system should be an open architecture system. Mechatronics has transcended traditional engineering disciplinary boundaries and makes the development of robotic systems easier than the traditional development path. Figure 3.3 shows a modern Mechatronic system where real-time software is at the heart of the system.

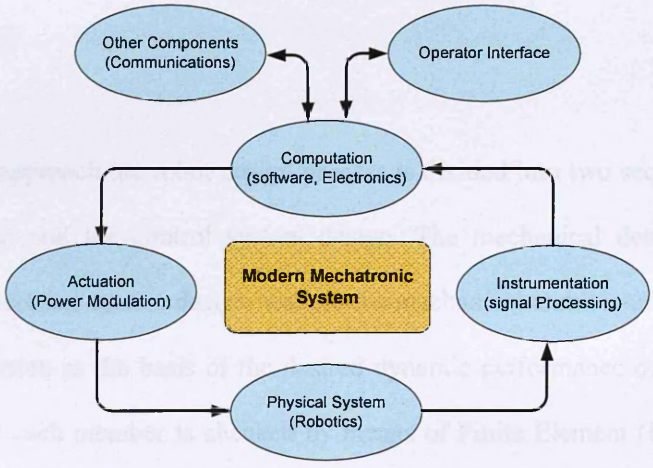


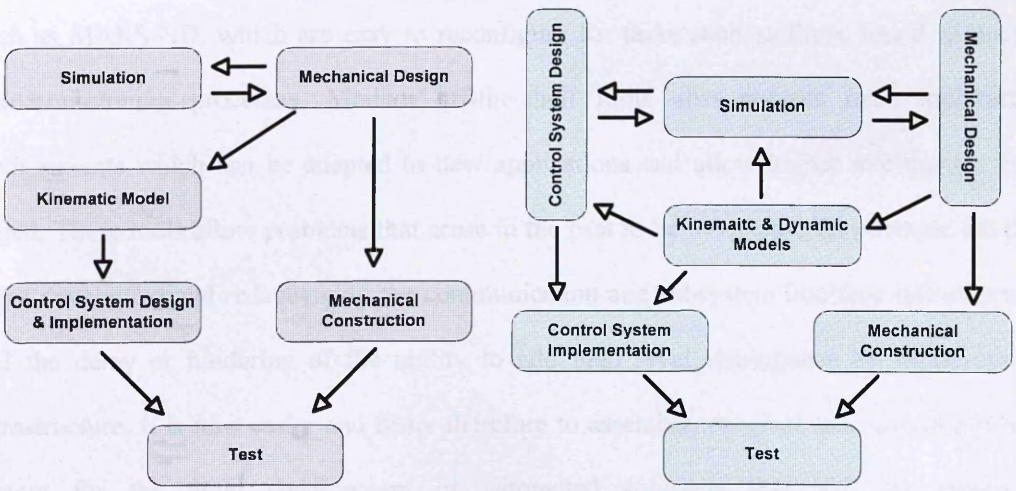
Figure 3.3 Modern Mechatronic System

Today the use of modern commercially off-the-shelf tools combined with the modern development path enables systems such as robotic systems to be developed without using the traditional development path. The modern development path using modern tools, allows researchers to undertake the following:

- To combine several control activities in an integrated environment using standardised bus communication between all tools. This is the basis of the development of fully distributed control systems. In the architecture of a fully distributed control system all the tools needed to control the robot system are connected to a single communication channel

- To build a real-time graphical simulation interface and a single advanced real-time graphical user interface that provide the operator with various options of operation modes to control robot tools and allow tasks to be achieved in a user friendly manner
- To use universal operating software, such as LabVIEW software, which is capable of connecting with a broad range of devices, protocols, and interfaces required for measurement applications
- To use an open architecture controller which can be changed or modified by the user, the hardware and software structure for this type of controller is such that integrating sensors, operator interfaces and new control laws for servos and actuators can be done more easily

In the traditional approach the robot design process is divided into two sequential phases: the mechanical design and the control system design. The mechanical design is not usually influenced by the control system design because the mechanical architecture is designed using multi-body simulation as the basis of the desired dynamic performance of the machine. The detailed design of each member is checked by means of Finite Element (FE) analysis which considers the required robot motion and applied forces.



(a) (b)

Figure 3.4 (a) Traditional approach of robot manufacturer (b) Integrated mechatronic approach

The control system is realised using kinematic equations which regulate the position of each actuator separately in order to impose the desired end-effector position. With the Mechatronic approach mechanics and control are studied simultaneously. This allows both the kinematic model and the dynamic model to be obtained. Figure 3.4 shows a comparison between the traditional industrial approach of robot manufacturers and the alternative modern integrated Mechatorinc approach (Luca *et al* 2003).

The research undertaken for this PhD represents a case study of the application of a Mechtronics approach combined with the use of modern tools to develop a robotic system such as MARS-ND without the use of a traditional development path. In the past this task would have required a team of engineers comprising mechanical, electrical, control, software and computer engineers to work together. The interaction among these and other disciplines would have proven to be quite complex. In this research the development of the robotic system has been undertaken primarily by a sole researcher.

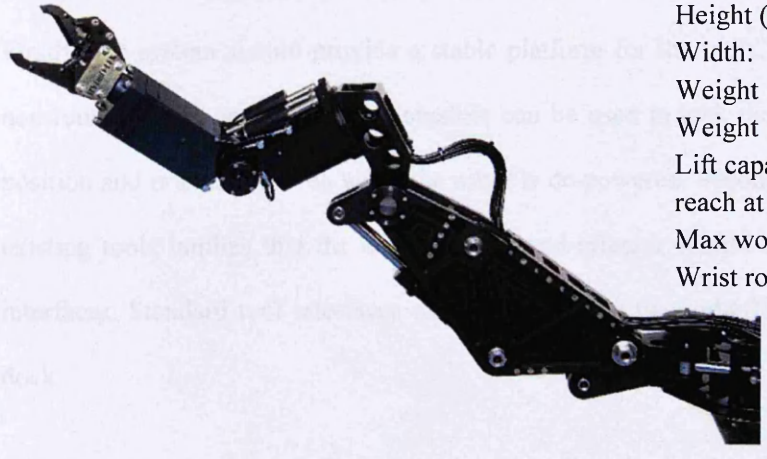
3.4 Modern off-the-shelf tools

Modern commercial off-the-shelf tools permit engineers to build sophisticated robotic systems, such as MARS-ND, which are easy to reconfigure for tasks such as those found in nuclear decommissioning processes. Modern off-the-shelf tools also possess open architecture environments which can be adapted to new applications and allow higher intelligence to be added. These tools allow problems that arose in the past to be overcome, for example the time spent developing and redeveloping the communication and subsystem interface infrastructure; and the delay or hindering of the ability to add high level intelligence by initial system infrastructure. It is now easier and faster therefore to assemble, program and control a robotic system for the rapid development of automated solutions that suit the nature of decommissioning activities. This rapid development is the result of the integration of new technology with commercially available off-the-shelf tools. An example of modern off-the-

shelf tools increasingly used within the robotics field are a new class of devices such as controllers and field devices, commonly referred to as Networked Embedded Devices (NED). Their increasingly low cost and small size makes them suited for the development of sophisticated robotic systems and large scale sensing applications.

The robotic system developed for this thesis, MARS-ND, was built entirely from commercial off-the-shelf components. MARS-ND is a hydraulically/electrically linked system with special features such as programmability, resolved motion and collision avoidance. The development of such a system enables subtle and precise demolition tasks to be performed in radioactive environments which are often too hazardous for humans. The main off-the-shelf components of MARS-ND consist of the smallest Brokk demolition machine, the Brokk 40 (Brokk Technical paper 2004), as a mobile platform; and two Hydro-lek robot manipulators (Hydro-Lek Ltd 2004). Suitability, cost and availability were all important factors in the selection of these particular off-the-shelf-components for the development of MARS-ND. These components and other off-the-shelf tools used within the MARS-ND system and some of the problems discovered are discussed in more detail below.

3.4.1 Hydro-Lek manipulator



Dimensions:

Length of arm:	1500mm
Length of slew plate:	410mm
Height (stowed):	800mm
Width:	180mm
Weight in air:	45kg
Weight in water:	32kg
Lift capacity at full reach at 160 bars :	150kg
Max working pressure:	210 bars
Wrist rotation:	360° continuous

Figure 3.5 Off-the-shelf 6 DOF Hydro-Lek manipulator

The Hydro-lek HLK-7W as shown in Figure 3.5 is a 6 DOF manipulator with a continuous jaw rotation mechanism and dual function gripper fitted with a pressure sensor and designed to grip an object and also cut ropes up to 19mm diameter. The azimuth yaw, shoulder pitch, elbow pitch, forearm roll and wrist pitch joints are fitted with potentiometer feedback sensors.

From the experience of previous working system (Esposito, C., *et al* 1993) the desirable characteristics of a manipulator for working in hazardous environment can be identified as having the following criteria:

1. Rugged construction
2. Waterproof
3. Modular design
4. Easy decontamination
5. Rugged electronics
6. Remote data communication system
7. Payload carrying capacity over 200kg
8. On-board power supply

In addition, the selection of a manipulator of 6-DOF is essential to perform full motion in 3D space. The specific user requirements before the purchase of an off-the-shelf manipulator introduce two important constraints which contribute to the technology selection process. Firstly, the system should provide a stable platform for the tools. Locking facilities such as non-return valves or self-locking actuators can be used to lock the manipulator joints at any position and orientation even when the robot is de-powered. Secondly, the requirement to use existing tools implies that the design of the end-effector should consider the tool manifold interfaces. Standard tool interfaces can be installed in the end-effector or in the tool change dock.

Figure 3.6 shows the Hydro-Lek manipulator purchased for this research. This manipulator was chosen because of its low cost and availability however, it did not meet all of the user requirements as explained in Table 3.1 below. It has the facility for self-locking actuators or non-return valves when the robot is powered but there is no locking mechanism that can hold the manipulator at any position and orientation when the robot is de-powered. This could be rectified by the addition of appropriate check valves in the hydraulic system. The robot manipulator came with a specific dual-function end-effector, in order to attach different types of end effectors modification will need to be carried out on the tool interfaces.

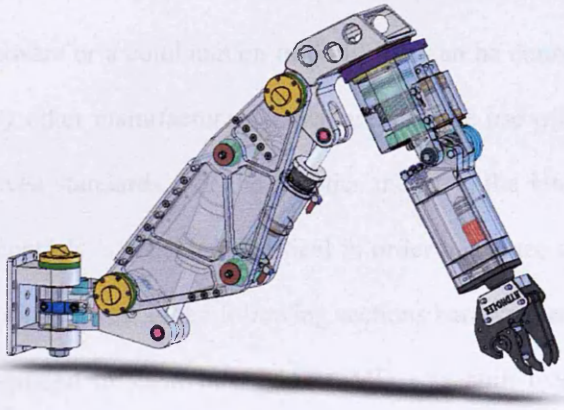


Figure 3.6 Hydro-Lek with built in actuators (Bakari 2006)

3.4.2 Brokk 40 machine

The Brokk 40 robot consists of a moving vehicle with a single 5-DOF manipulator with five linear actuators; a hydraulic tank; and a controller and remote control device designed to operate the vehicle and its manipulator.

Figure 3.7 Brokk 40 Machines, one with tool attached and one without tool

The Brokk 40, as shown in Figure 3.7, is an off-the-shelf machine which was designed for heavy demolition tasks and to pass through very narrow spaces such as 650mm wide doorways. Its low weight enables it to be used in most buildings and small dimensions (Width: 600mm; Height: 900mm) allow it to operate in confined spaces. It is electrically powered which also facilitates internal use. It is the smallest robot in the Brokk family.

3.5 Integration and interfaces

Integration and interfaces in an open architecture system refers to an information technology system (software, hardware or a combination of both) that can be connected easily to devices and programs made by other manufacturers. Open architecture use off-the-shelf components and conform to approved standards. For the robotics industry, the interoperability permitted by open architecture controls is considered critical in order to reduce the price of integration between different robotic systems. In the following sections hardware and software integration and interfaces are discussed to show how MARS-MD was built using tested off-the-shelf commercially available components.

3.5.1 Universal bracket attachment

For this research, the Brokk 40 machine was modified to carry and transport two 6 DOF manipulators. A universal bracket was manufactured and fitted at the end of the Brokk manipulator where the multi-arm system is attached, as shown in Figures 3.8 and 3.9.

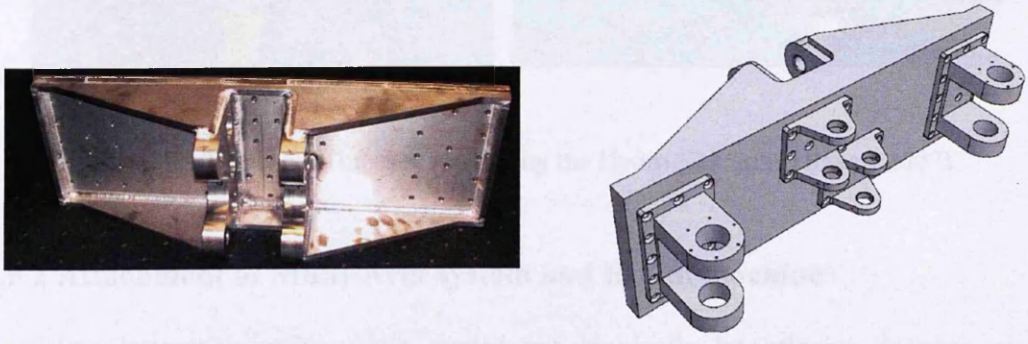


Figure 3.8 Universal bracket (Bakari 2007)

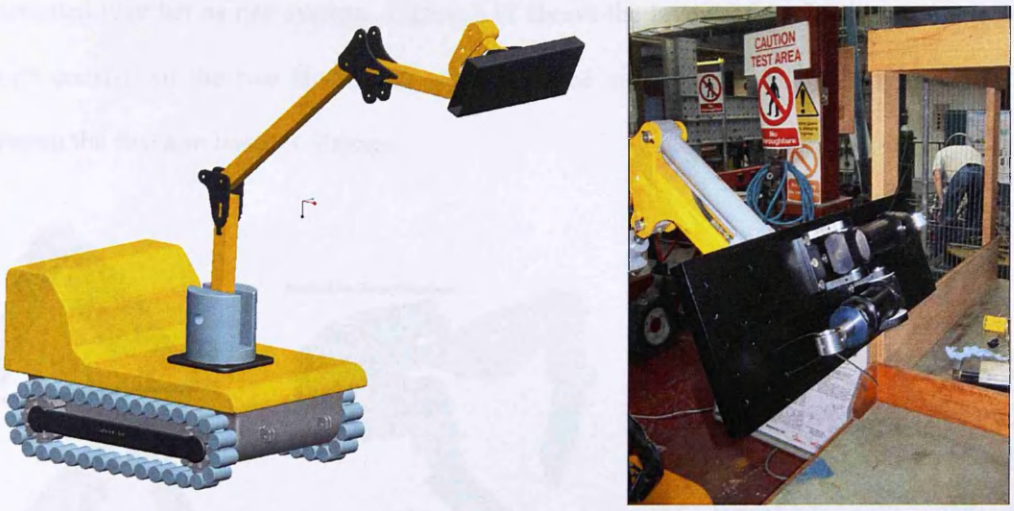


Figure 3.9 Universal bracket attachment with Brokk Machine (Bakari 2007)

The mounting bracket was designed to both hold the weight of the two arms with full payloads and act as a stable platform for the two arms. The other side of the bracket is designed to be fitted to the end of Brokk machine manipulator where it can be rotated forward and backward. Figure 3.10 shows the front of the mounting bracket with the two Hydro-Lek arms attached. Figure 3.10 also shows the back view of the mounting bracket with the end of the Brokk manipulator attached to the bracket and the hydraulic hoses for operating the Hydro-Lek arms.

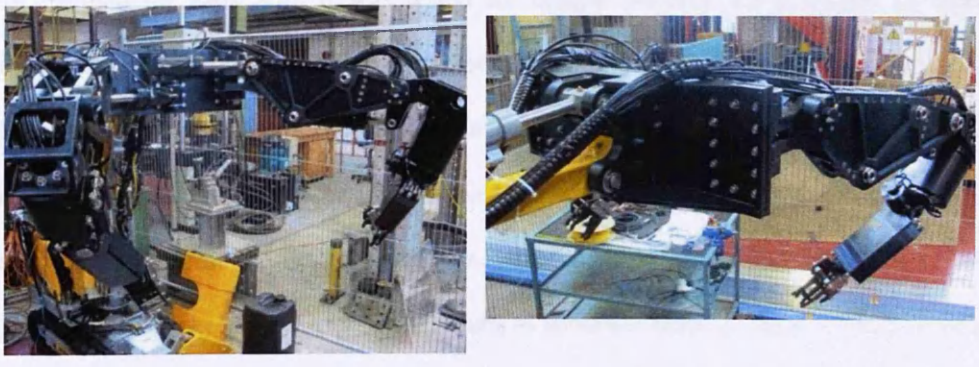


Figure 3.10 Universal bracket supporting the Hydro-Lek arms (Bakari 2007)

3.5.2 Attachment of Multi-Arm system and Brokk machine

A system integration process was carried out physically by bringing together many subsystems and their components into one system and ensuring that the subsystems fitted and

functioned together as one system. Figure 3.11 shows the layout of the hardware integration which consists of the two Hydro-Lek arms mounted on a mounting bracket. The distance between the two arm bases is 500mm.

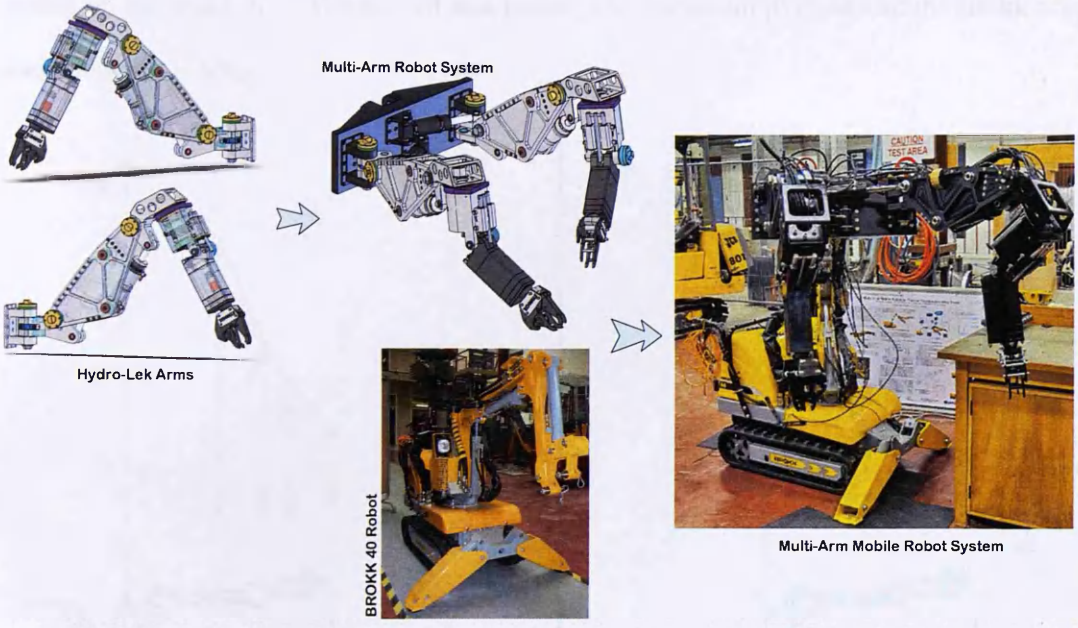


Figure 3.11 Hardware integration (Bakari 2007)

The mobile platform of the Brokk machine was used to carry the multi-arm system with a full 100kg payload for each arm. It was tested for critical position for a variety of different configurations of both arms. An important problem that became evident concerned stability for the system as a whole and specifically for the Brokk machine, due to the weight of both Hydro-Lek arms and high payloads. It became evident that the design of the Hydro-Lek arms did not meet our original design specifications, which included lower weight and shorter arms. It was necessary to explore a variety of solutions to solve the stability issues. The best solution found was to remove the last link of the Brokk machine manipulator in order to counter-balance the payloads without using the Brokk front stabilisers. For this research however, the last link of the Brokk manipulator was not removed because the Brokk remained predominantly still for research purposes.

3.5.3 Brokk load and stability

Figures 3.12 and 3.13 below show the load and stability of the Brokk machine with and without the use of a front stabiliser. These figures illustrate a variety of payloads at different points on the Brokk arm, without tool attachment. The maximum payload that the Brokk arm can withstand is 60kg.

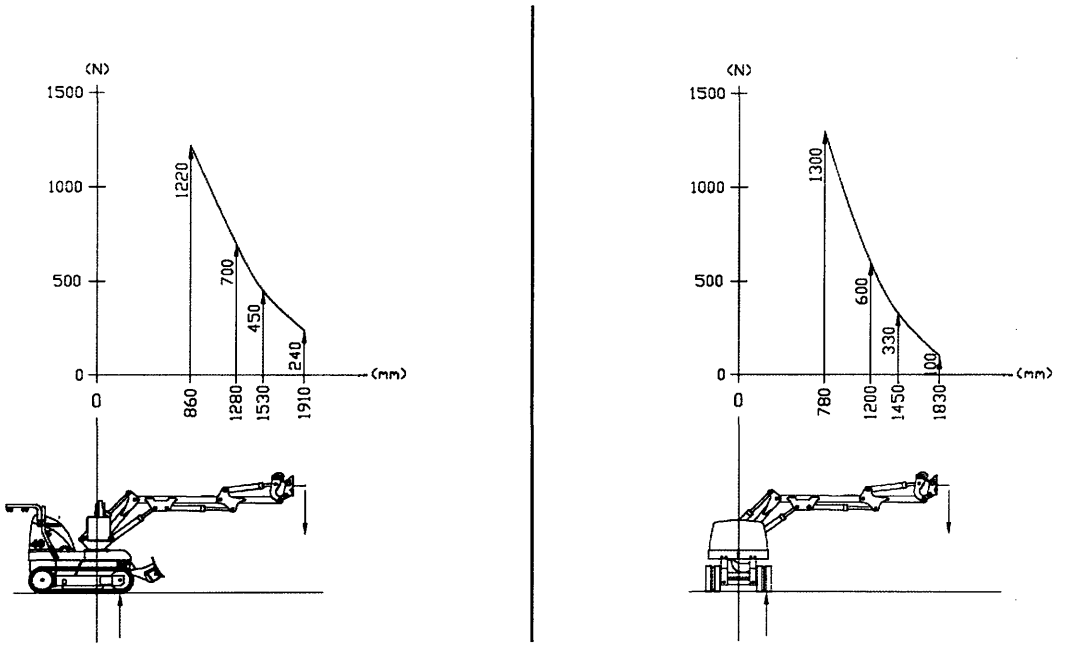


Figure 3.12 Brokk load and stability without the use of the front stabiliser (Brokk Technical Paper 2009)

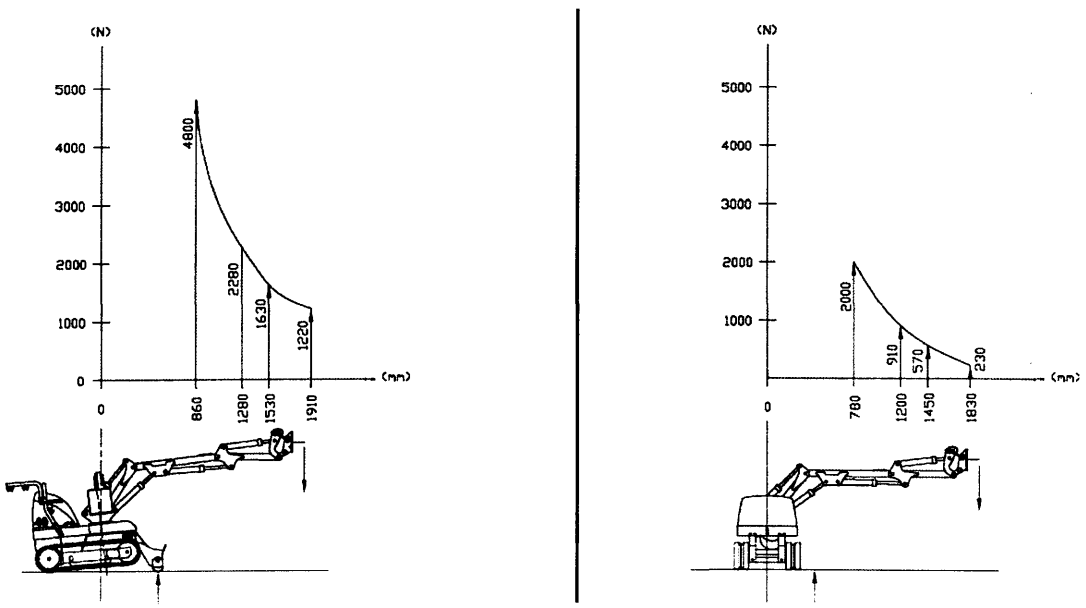


Figure 3.13 Brokk load and stability with the use of the front stabiliser (Brokk Technical Paper 2009)

The design of the Hydro-Lek arms received for this research did not meet our original design specifications, therefore the weight of the attached Hydro-Lek multi-arm system, illustrated in Figure 3.14 below, exceeded the maximum payload of the Brokk arm. This made it essential to apply the front stabiliser of the Brokk machine to counter balance the excessive loads.

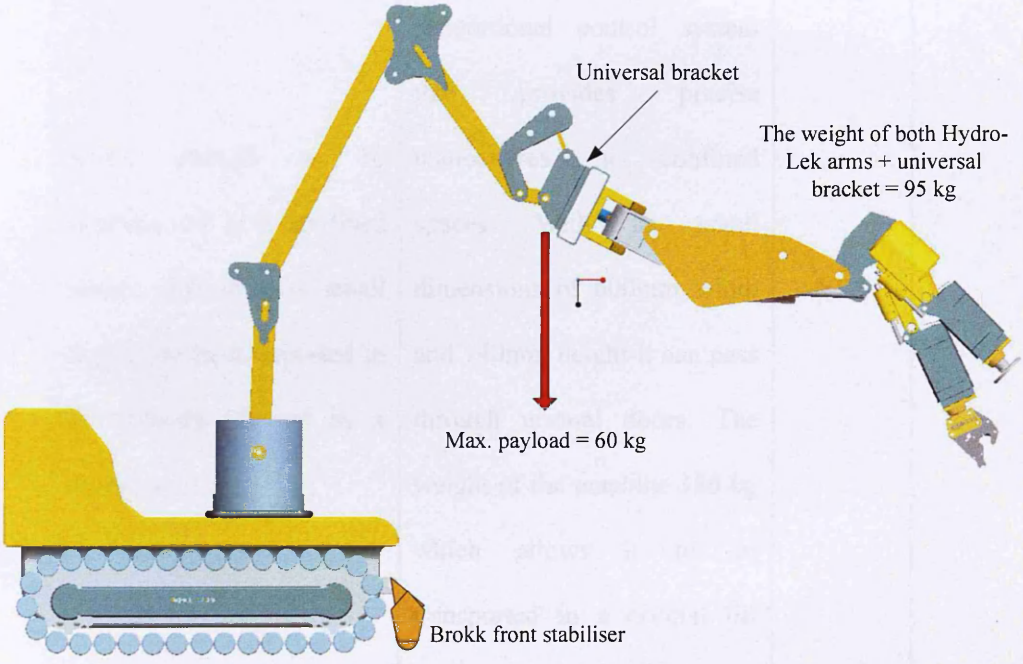


Figure 3.14 Brokk load stability with the attachment of Hydro-Lek arms (Bakari 2006)

3.5.4 MARS-ND final design specification

Table 3.1 below gives a summary of the original design specification required for this research, and the final design specification following the delivery of the components of MARS-ND. The purpose of this table is to analyse, evaluate and identify the limitations of the original robot design specifications of MARS-ND identified in order to meet the objectives of this research; and the final design specifications. For the sake of this research and its time frame, the final design specifications were accepted to allow the processes of its integration, interface, control and observation to be undertaken during the period of this research.

No.	Original design specification	Final design specification	Met our design objectives	
	Brokk 40 mobile platform			
1	Small enough to be manoeuvred in a confined space, able to pass small doors and be transported in an ordinary lift and by a small van	Brokk 40 machine has a proportional control system that provides precise manoeuvres in confined spaces. With its small dimensions of 600mm width and 740mm height it can pass through normal doors. The weight of the machine 380 kg which allows it to be transported in a normal lift and small van	YES	
2	To have enough hydraulic power to operate both Hydro-Lek arms	Brokk's hydraulic system pressure 17,5 MPa and Pump flow rate of 13,5 l/min were more than was needed to operate the Hydro-Lek arms. A pressure relief valve was used to lower the oil pressure required by Hydro-Lek actuators	YES	
3	To have enough hydraulic power to lift the Hydro-Lek	The Brokk system had enough hydraulic power to lift	YES	

	arm system to any position without struggle	the Hydro-Lek arms upward, downward, left and right		
4	The weight of the Hydro-Lek arms + the universal bracket for attachment should not exceed the Brokk maximum payload of 60 kg	The weight of the Hydro-Lek arms was 95 kg, which exceeded the maximum payload of the Brokk machine. This caused problems for the stability of the Brokk machine		NO
	Hydro-Lek 6 DOF manipulator			
6	To have 6 revolute joints (6 DOF)	Each Hydro-Lek arm has 6 revolute joints and a gripper	YES	
7	To possess human arm configuration and flexibility	The Hydro-Lek arms possess human arm configuration with azimuth yam, shoulder pitch, elbow pitch, forearm roll, wrist pitch and wrist rotate, but they lack flexibility in terms of joint configuration		NO
8	To be similar or close to human arm size of 1000 to 1200mm length. To have a width of 150 mm	Hydro-Lek was delivered with 1500 mm length and 180 mm width		NO
	The weight of each arm to be 30 kg - 32 kg so that	The Hydro-Lek arms delivered were 45 kg in air		

9	both arms + the universal bracket used for attachment would be under the maximum payload of the Brokk arm	and 32 kg in water. Unfortunately both arms were planned to be used in air and not in water		NO
10	Each joint of the arm fitted with a feedback sensor such as a potentiometer sensor	All of the joints of the Hydro-Lek arm were fitted with potentiometer sensors except the wrist joint which only rotates 360° continuously. This caused problems for the formation of the kinematic equation needed to build a PID control system for the wrist joint		NO
11	Maximum hydraulic working pressure not to exceed the Brokk working pressure	The Hydro-Lek arm maximum working pressure was 16 MPa. It only needed a small pressure relief valve to be attached to the main hydraulic line to reduce the Brokk working pressure of 17,5 MPa to around 12 MPa	YES	
12	To have a locking system in order to lock the joint actuators when the system is de-powered	No locking facility		NO

13	The work envelop of both arms to be similar to the work envelop for human arms. The work envelop is determined from the sum of the degrees of freedom (DOF) plus the length of the robot arm	The distance between both arms was 500 mm. The length of both arms were longer than required but it covered the work envelop when undertaking a given pipe cutting or parts dismantling task	YES	

Table 3.1. The design objectives of MARS-ND and its limitation

3.5.5 cFP controller and valve pack integration

Figure 3.15 shows the integration of the cFP controller and the valve pack through cables. These cables were originally connected when the proportional amplifier was delivered.

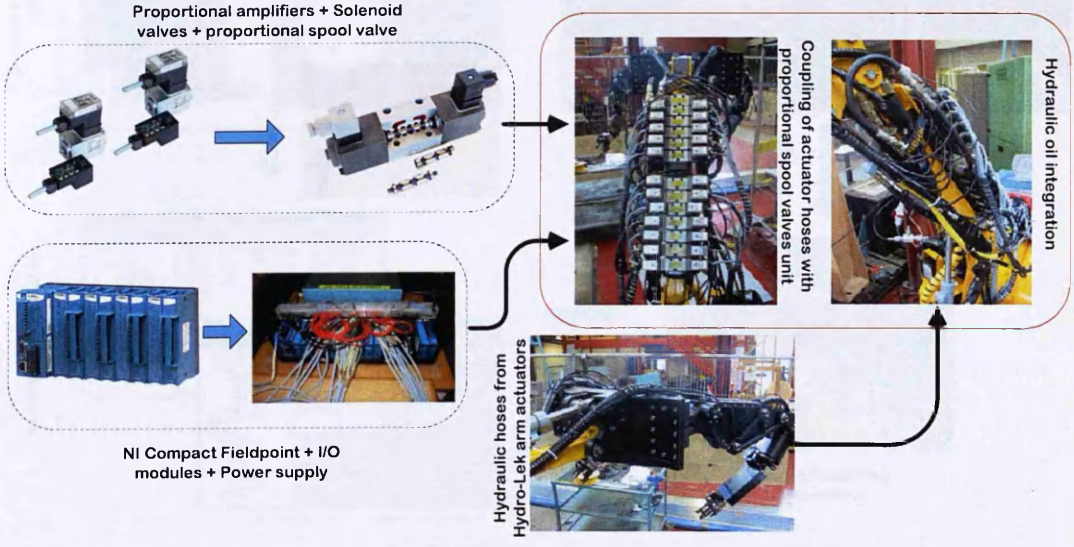


Figure 3.15 Electronic and hydraulic integration (Bakari 2007)

3.5.6 Using NI Measurement and Automation Explorer (MAX)

The development software used to control robotic systems is expensive due to the inherent complexity of these systems. There is therefore a need to develop tools that permit a reduction in the programming effort and aim for the generation of modular and robust applications.

LabVIEW software fulfils many of these requirements because it is an object orientated framework for programming robotic systems and it provides a full-featured graphical programming tool to develop measurement, automation and control applications and its development system is reusable software which has a clear interface. Furthermore LabVIEW is an open system that allows the researcher to carry out modifications, extensibility and integration with other systems. Figure 3.16 shows the data flow layout throughout the system from the LabVIEW interfaces on the host PC (user interface) to the robot arm joints.

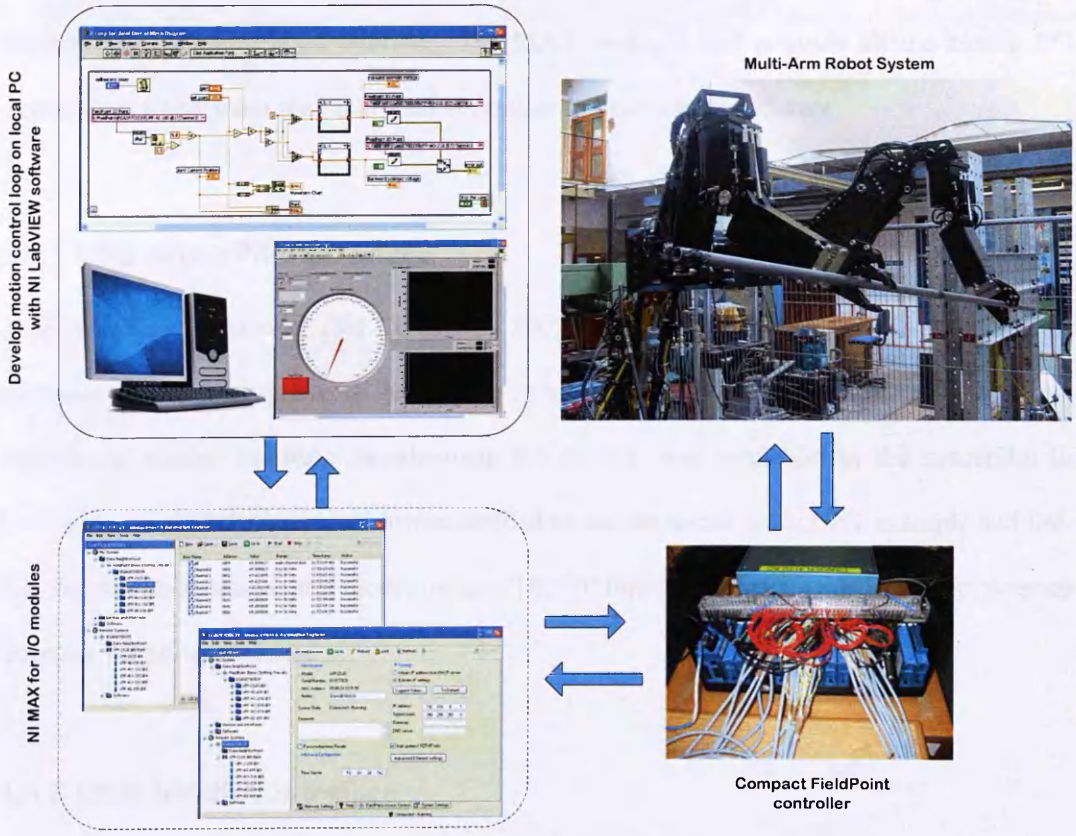


Figure 3.16 Software integration layout (Bakari 2007)

LabVIEW operational software was used to create and test the motion control loop before it was downloaded to the FieldPoint controller in the building of MARS-ND.

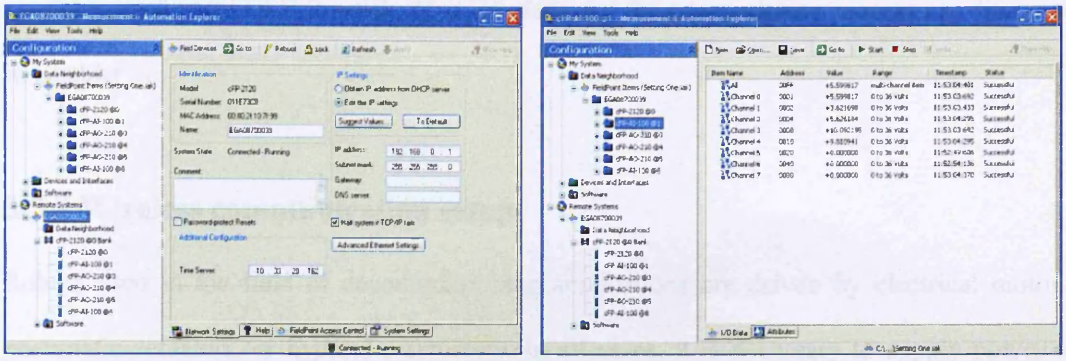


Figure 3.17 NI Measurements and Automation Explorer (MAX)

The NI Measurement and Automation Explorer (MAX) (National Instruments 2006) is the software downloaded to the host PC, as shown in Figure 3.17. It is integrated with the software driver for FieldPoint hardware and used to configure the FP hardware. It also acts as the network communication interface. The MAX manages and controls all the analog I/O module signals between the FieldPoint controller and LabVIEW software.

3.5.7 USB SpacePilot interface

A 3D spacepilot controller (3D Connexion 2005) was used to control the MARS-ND arms therefore it was appropriate to bridge the MARS-ND with the spacepilot using the java behavioural model. Software Development Kit (SDKs) was available for the spacepilot in C++, visual basic.NET and java. It was decided to use the visual basic.NET example and link this into the Java behavioural model using a TCP/IP interface. This was the intended external interface for the application.

3.5.8 USB Joystick interface

The use of a commercially available USB joystick called Predator GM-2500 (Trust Company Products 2007) made it possible to overcome the limitations and difficulties faced when using

the SpacePilot. A joystick simulation VI model was available in the LabVIEW built -in library. This model was designed to recognise and communicate with the commercially available third party USB joystick. This interface and the programming process are discussed in Chapter 5.

3.5.9 Wireless communication setup

Robots used in the field of decommissioning applications are driven by electrical motors, pneumatic actuators, or hydraulic actuators. In all cases, it is necessary to supply power to each actuator and take the information from the sensors in the joints and end-effectors of the robot. This requires many cables and wires to provide the communication between the operator on the host PC and the robot while in a decommissioning environment that is too hazardous for the presence of human operators. These cables and wires however, need to be long enough for the robot to travel to its designated target. This immediately creates a problem as these cables limit the movement and manoeuvrability of the whole robot. In this research the communication between the human operator in the host PC and the robot is wireless apart from one single cable which provides the electrical power needed to operate the Field Point controller and the hydraulic oil pump which operate the robot actuators. Figure 3.18 shows the real-time operation of the tele-operated multi-arm mobile robot over a wireless link using off-the-shelf commercially available wireless network systems.

A standard IEEE 802.11 (WLAN Standards 2005) wireless local area network system was setup for the control of the MARS-ND system. A wireless access point (AP) was used for this wireless networking to allow the communication between the Compact Field Point controller and the host PC through an Ethernet converter and NETGEAR wireless USB (NETGEAR 2005). The Compact FieldPoint 2120 automation controller sent all the control signals that drove the actuators and received the feedback signals from the joint sensors. The Field Point network communication interfaced then automatically published all the measurements through the wireless network to MAX on the host PC.

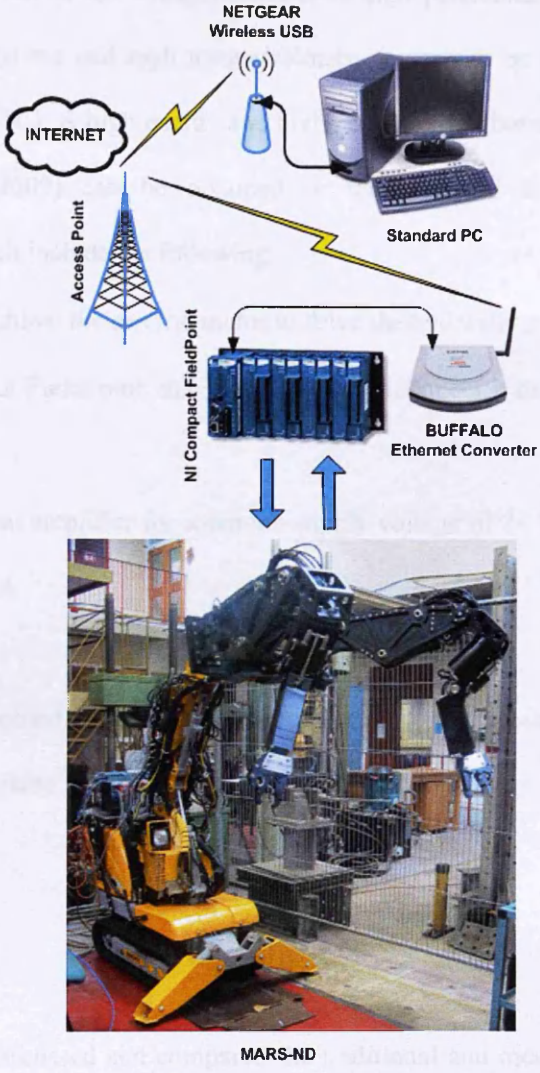


Figure 3.18 Wireless communication system for MARS-ND (Bakari 2006)

The MARS-ND system is completely wireless system except for a single cable attached to the mobile platform which provides the electrical power needed to operate the mobility of the Brokk machine; the Field Point controller; and the proportional amplifiers needed to operate the solenoid valves which operate the robot actuators. For the next stage of this research it will be necessary for MARS-ND to be completely wireless so that it can be used to execute a given task in an environment that is too hazardous for human presence. It will therefore be essential to consider the use of a powerful on-board rechargeable battery to provide an equivalent electrical power to operate MARS-ND. This battery could be attached, for

example, to the back of the Brokk machine. A high performance Li-ion battery pack that possesses good runtime and high torque/velocity power can be used as an on-board power supply for MARS-ND. A high energy and high power Li-ion battery manufactured by (ABSL Power Solution 2009) can be obtained or manufactured to meet MARS-ND power specifications which include the following:

- Brokk machine: the electric motor to drive the hydraulic pump requires 4000 Watts
- NI compact FieldPoint: maximum power to connect it to the I/O modules requires 9 Watts
- Proportional amplifier for solenoid: supply voltage of 24 VDC and maximum current of 1200 mA

The use of an on-board battery was not considered for this research because the aim of the research was to create a prototype to be used in a laboratory setting solely for research purposes.

3.6 Conclusion

This chapter has discussed and compared the traditional and modern paths for developing a robotic system. It has shown the disadvantages of the traditional path and the importance of building a sophisticated robotic system using a modern development path. The modern development path includes the use of a mechatronic design process with available modern commercial off-the-shelf tools that possess open architecture environments. These tools can be adapted to new applications and allow higher intelligence to be added. This chapter has also discussed the selection of commercially available off-the-shelf hardware and their integration within a mobile delivery platform. The chapter then describes the selection of commercially available off-the-shelf hardware for the development of MARS-ND.

The specific tools chosen for MARS-ND were selected because of the ease with which they communicated with one another, and could be integrated and controlled from one operating system. One of the premises of this research was to develop MARS-ND using the modern development path and the application of mechatronics because of the ease of communication, integration and interaction. The application of mechatronics concepts within this research allowed one researcher to develop MARS-ND without the need for external expertise in terms of software and hardware integration, programming and motion control of the robot. These choices proved to be an effective approach to the development of MARS-ND both in terms of the functionality of the robot with respect to the off-the-shelf tools selected; and the ability for one researcher to integrate the software and hardware systems and to build the motion control profile. The off-the-shelf tools that were used to build the foundation of MARS-ND are outlined in this chapter, while the rest of the items are illustrated in Appendix A.

The use of LabVIEW operating software, as discussed in this chapter, facilitated the creation of motion control for the robot arm because of the ability of LabVIEW to integrate and communicate with all of the off-the-shelf tools within a single user interface. This is a different approach to previous research projects which have used specific pieces of software for each tool, creating several user interfaces to facilitate control of the robot arm. The application of LabVIEW within this research simplified this process because of the use of one user interface.

Furthermore this chapter has explained the wireless communication system setup for MARS-ND. This involved the application of a Compact FieldPoint controller using a standard IEEE 802.11 wireless local area network system, an Ethernet converter and USB NETGEAR for MARS-ND. The Compact FieldPoint controller is an example of a new class of device commonly referred to as Networked Embedded Devices (NED). These NED are designed to permit wireless networking to be setup easily, using wireless local area network systems that allow communication between the motion controller and the user interface on the host PC in

order to control the motion of the robot arms. The use of NED as a wireless communication system for MARS-ND made the wireless communication system a relatively simple and cheap system that could be easily set-up and did not need software expertise.

The purpose of hardware integration and the development of software interfaces are to establish the low and high levels controllers. In order to establish the low level controller it is necessary to have available the kinematics model of the robot arms. Kinematics modeling of the robot arms is important in order to understand the behavior of the arms. The low level and high level controllers used for MARS-ND were modeled using LabVIEW operating software system. Their function was implemented through the application of a Compact FieldPoint controller which was used to program and implement selected tasks. This use of LabVIEW operating software and Compact FieldPoint controller allowed the control of the robot arms while undertaking a given task. The Kinematics modeling and the low level controller used for MARS-ND are discussed in detail in Chapter 4.

CHAPTER 4

HYDRO-LEK ROBOT KINEMATICS AND CONTROL

4.1 Introduction

To achieve smooth and fast motion of robot manipulators it is necessary to use model-based motion control schemes. In order to achieve high-performance model-based motion control a control engineer must take the following steps:

I. Model the kinematics and dynamics of the robot

II. Obtain model parameters via direct measurements and, or identification

III. Establish the accuracy of the models and validate the estimated parameters

IV. Deduce to what extent the rigid-body model covers the real robot dynamics

Model-based motion control schemes also employ models of robot kinematics and dynamics.

Robot kinematics can be classified as forward kinematics (FK) and inverse kinematics (IK). FK is a computation of the position and orientation of the robot's end-effector as a function of its joint angles. IK is a computation of the joint angles of the robot arm from the end-effector coordinates by means of its position and orientation. The solution of the robot manipulator IK can be split into two categories:

1. Closed form solutions: The IK can be rewritten in a manner that leads to a set of highly structured non-linear equations that may be solved explicitly for the joint variables. The closed form methods commonly used are: geometrical and algebraic. The

geometrical approach exploits all geometric relations of the manipulator under study (Hemami, A. 1987 and Lee & Ziegler 1982). The algebraic method (Paul, R.P. *et al* 1981 and Paul *et al* 1984) is based on the manipulation of the homogeneous transformation matrices in order to isolate the joint variables. This method is the most common approach used to obtain the inverse kinematic solution of a robot manipulator.

2. Numerical solutions: A numerical algorithm is applied that explicitly generates all solutions in a computationally feasible manner. These solutions are iterative in nature and have been the subject of extensive research (Lumelsky, V. J. 1984, Milenkovic, V., and Huang, B., 1983, Uicker, J. J. *et al* 1964, Goldenberg *et al* 1985 and Gupta, K.C. and Kazerounian, K 1985).

Software packages such as the Symbolic Robot Arm Solution Tool (SRAST) developed by Luis G. *et al* (1988) can also be used to calculate FK and IK. They allow the engineer to avoid the need to use algebraic computations to verify results. SRAST is a symbolic computation of robot manipulator kinematics that symbolically solves the FK and IK of an n degree-of-freedom manipulator with the use of Artificial Intelligence Techniques. As an input it expects corresponding Denavit-Hartenberg (D-H) parameters; as an output it generates a closed form of the FK and IK solution.

The D-H convention is a notation system developed to assign orthonormal coordinate frames to a pair of adjacent links in an open kinematic chain (Denavit, J., & Hartenberg, R., 1955). The procedure involves finding the link coordinates and using them to find 4×4 homogeneous transformation matrices composed of four separate submatricies to perform transformations from one coordinate frame to its adjacent coordinate frame. D-H is also a form of kinematic calibration. In order to compute the FK and IK, kinematic calibration is needed.

Kinematic calibration is a process of determining the actual kinematic parameters by using the nominal kinematic parameters and some measurements made on the robot. Various methods

have been used for calibrating robots. These methods have been reviewed by Roth, Z. *et al* (1987) and Sheth and Uicker (1971). In general calibration involves the following three stages:

1. Modelling which involves developing a model to represent the kinematic parameters as a function of a measurable parameter (Paul 1982, Kirchner *et al* 1987, Judd and Knasinski 1987)
2. Measurement which involves developing a strategy to measure the measurable parameters (Whitney *et al* 1986, Bennett and Hollerbach 1988 and Driels 1993)
3. Identification which involves determining the actual kinematic parameters by using the model and measurements (Skiar 1987, Driels 1990 and Ananth 1992)

In the discussion of kinematics and dynamic modelling within robotics literature, a number of modelling methods are presented to meet a variety of different requirements (Sciavicco and Siciliano 1996, Vukobratovic and Potkonjak 1982, Fu *et al* 1987 and Kozlowski 1988). For robot kinematics, the model suggested is a mapping process between the task space and the joint space. The task space is the space of the robot-tip coordinates, these include:

- The end-effector's Cartesian coordinates
- The angles that define the orientation of the end-effector

The joint space is the space of joint coordinates, these include:

- The angles for revolute joints
- The linear displacements for prismatic joints

The robot configuration is defined in the joint space. The mapping from the joint to the task space is the FK or direct kinematics. The opposite mapping is the IK which reveals singular configurations that must be avoided during manipulator motions. Both FK and IK can be represented as recursive or closed-form algebraic models. The algebraic closed-form representation facilitates the manipulation of a model enabling its straight forward mathematical analysis. In the context of robotics it is important to achieve high accuracy of

computation as quickly as possible; this can be achieved faster with the closed-form models than a numerical iteration solution. The closed form models are therefore preferable to a numerical iteration solution for real-time control. It is usually not an easy task however to derive closed-from models in compact form in particular IK models, even when software for symbolic computation is used. The derivation demands a series of operations, with permanent combining of intermediate results to enhance compactness of the final model. To simplify derivation, a control engineer sometimes approximates robot kinematics for example, by neglecting link offsets according to D-H notation (Sciavicco and Siciliano 1996; Fu *et al* 1987).

When a model is derived it is useful to establish its accuracy. To compare it with a recursive representation of the same kinematics is a straightforward procedure with available software packages. Examples of software routines specialised for robotic problems include *Workspace* and others presented in Nethery and Spong (1994) and Corke (1996). Once a model has been prepared the next step is to estimate the model parameters. Kinematic parameters can be obtained with sufficient accuracy, as they are found through direct measurements. The estimation itself is a process that requires identification experiments performed directly on the robot. To establish experimental conditions allowing for the simplest and the most time-efficient least-squares (LS) estimation of the parameters, joint motions, speeds, and accelerations are reconstructed via an observer (Belanger 1992 and Belanger *et al* 1998). After the estimation is finished experimental validation of the model has to be carried out. The objectives of experimental validation are to test how accurate the model represents actual robot dynamics. If the accuracy is satisfactory the application of the model for model-based control purposes can be established. For validation purposes a manipulator needs to execute motions similar to those it is intended to perform in practice. In the task space, these are the sequences of straight-line and curved movements. A dynamic model can be further used in model-based control algorithms. The model contributes to the performance of the applied control algorithms to the extent that it matches the real robot dynamics.

4.2 Hydro-Lek robot modelling for motion control

4.2.1 Kinematic modelling

Robot manipulators can be considered as a set of bodies or links connected in a kinematic chain by joints (Lewis *et al* 2004). The HLK-7W Hydro-Lek arm is a 7-function robot manipulator (six rotary joints and one prismatic joint for the gripper).

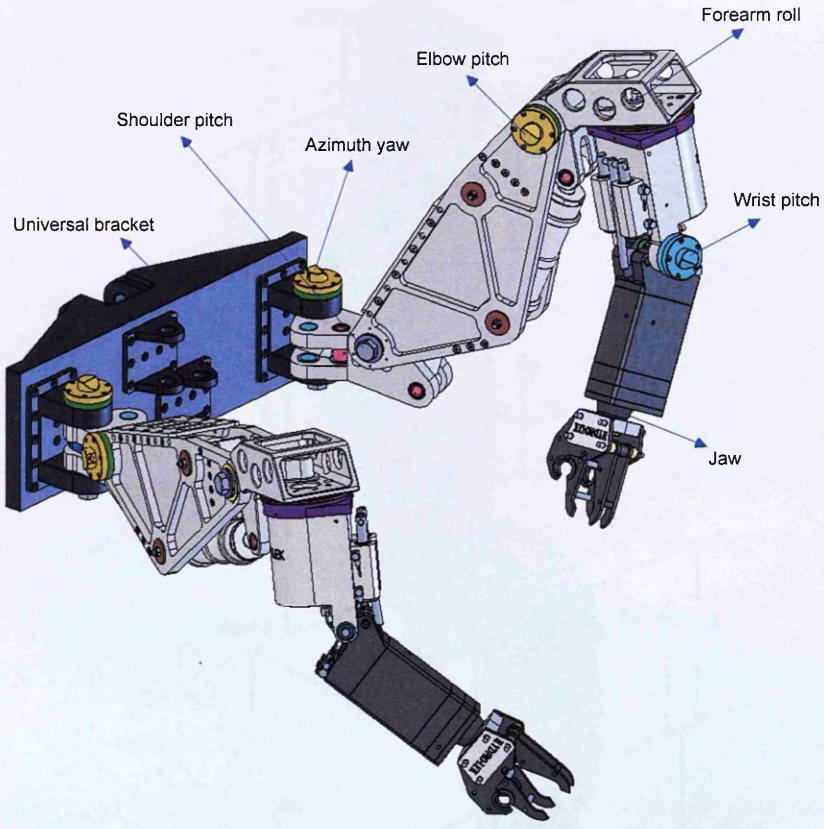


Figure 4.1 Hydro-Lek multi-arm system

Figure 4.1 shows the 7-function or 7 degree of freedom (DOF) HLK-7W manipulator. Joint one (azimuth yaw) rotates with an axis perpendicular to plane XY. Joint two (shoulder pitch) rotates perpendicular to joint one. Joint three (elbow pitch) rotates parallel to joint two and is offset by link indicates as A. Joint four (forearm roll) is perpendicular to joint three and is offset by the link indicated as B. Joint five (wrist pitch) is perpendicular to joint four, parallel to joint three and is offset by the link indicated as C. Joint six (jaw) is perpendicular to joint five and is offset by the link indicated as D. The Hydro-Lek HLK-7W manipulator structure is

kinematically defined by giving each link four parameters which are d_i, a_i, θ_i and α_i. The four given parameters describe how to get from one joint to another. Neighbouring links have a common joint axis between them. The distance along the common axis from one link to the next link is offset d_i. The amount of rotation about the common axis between one link and its neighbour is joint angle θ_i.

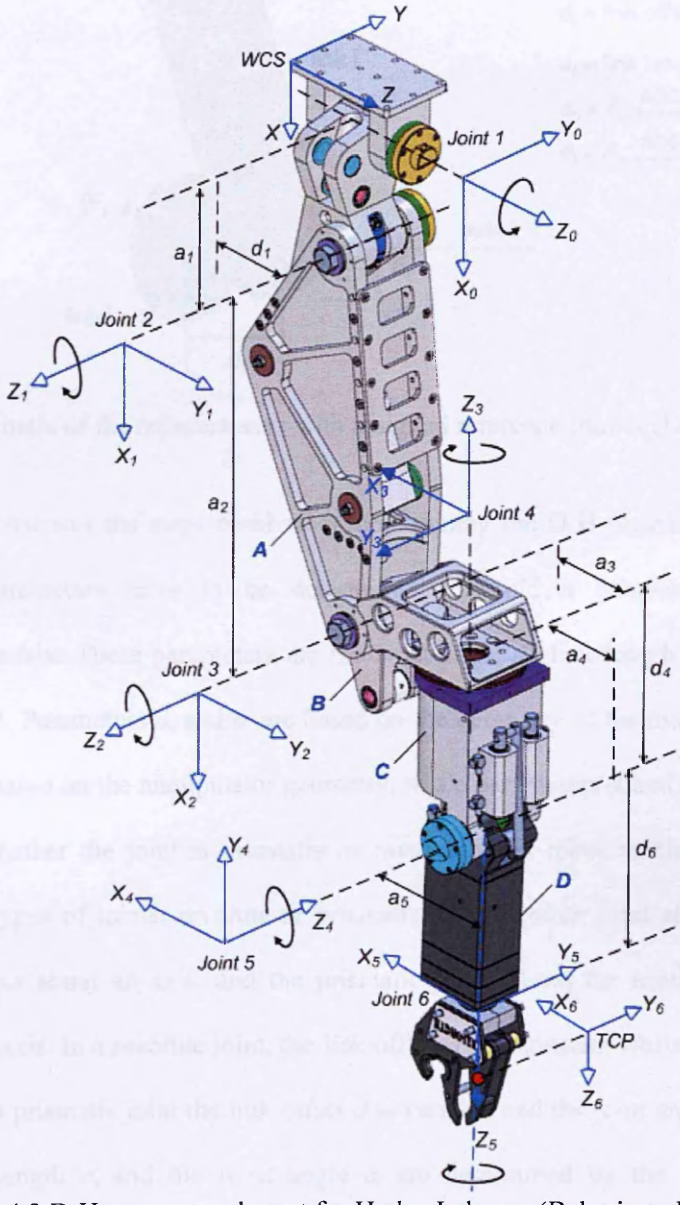


Figure 4.2 D-H parameters layout for Hydro-Lek arm (Bakari *et al* 2007)

The definition of mechanisms by means of these four parameters is the D-H convention (Cox, D., 2004; Denavit, J., & Hartenberg, R., 1955; Hemami, A., 1986). The location and orientation of each joint frame is shown in Figure 4.2.

Figure 4.3 shows the following connections: link i connected to link i-1 to link i+1 through a_i.

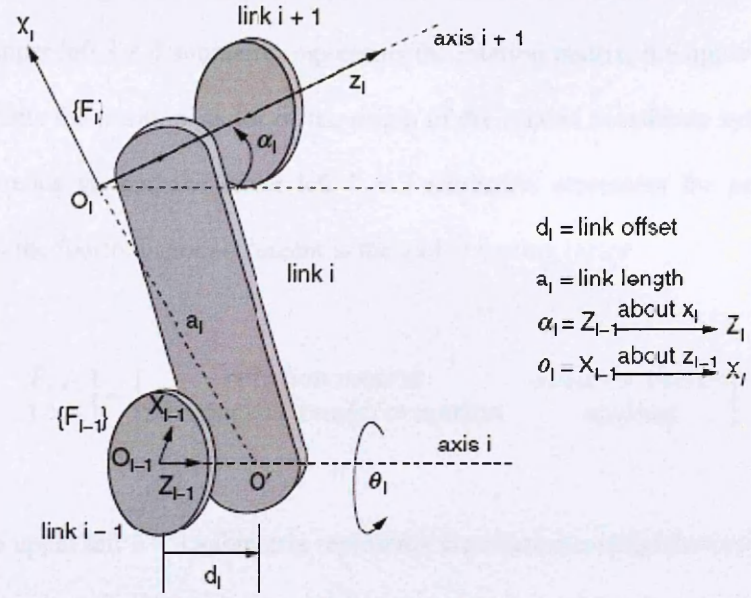


Figure 4.3 Schematic of the adjacent axes with assigned reference frames (Desai, J. P., 2005)

This figure demonstrates the steps involved in determining the D-H parameters. In the D-H method four parameters have to be determined to build a complete homogeneous transformation matrix. These parameters are the twist angle α_i, link length a_i, link offset d_i, and joint angle θ_i. Parameters α_i and a_i are based on the geometry of the manipulator and are constant values based on the manipulator geometry, while parameters d_i and θ_i can be variable depending on whether the joint is prismatic or revolute. In a robot manipulator, there are commonly two types of joints: revolute or prismatic. The revolute joint allows for rotation between two links about an axis, and the prismatic joint allows for translation or sliding motion along an axis. In a revolute joint, the link offset d is a constant while the joint angle θ is a variable; in a prismatic joint the link offset d is variable and the joint angle θ is normally zero. The link length α_i and the twist angle a_i are determined by the geometry of the manipulator and are therefore constant values.

Having determined all the D-H parameters, the transformation matrix $\boldsymbol{T}$ can now be computed. The homogeneous transformation matrix is a 4 × 4 matrix which maps a position

vector expressed in homogeneous coordinates from one coordinate system to another coordinate system. A homogeneous transformation matrix can be considered to consist of four submatrices. The upper left 3 × 3 submatrix represents the rotation matrix; the upper right 3 × 1 submatrix represents the position vector of the origin of the rotated coordinate system with respect to the reference system; the lower left 1 × 3 submatrix represents the perspective transformation; and the fourth diagonal element is the global scaling factor.

$$T = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ f_{1\times3} & 1\times1 \end{bmatrix} = \begin{bmatrix} rotation\ matrix & position\ vector \\ perspective\ transformation & scaling \end{bmatrix} \quad (4.1)$$

In equation 4.1, the upper left 3 × 3 submatrix represents the rotation matrix; the upper right 3 × 1 submatrix represents the position vector of the origin of the rotated coordinate system with respect to the reference system; the lower left 1 × 3 submatrix represents perspective transformation; and the fourth diagonal element is the global scaling factor. In order to find the homogeneous transformation matrix **T** that represents a rotation of α angel about the ***OX*** axis, followed by a translation of a unit along the ***OX*** axis, followed by a translation of d along the ***OZ*** axis, followed by a rotation of θ angle about the ***OZ*** axis it is necessary to use equation 4.2 as given below:

$$T = T_{z,\theta} \times T_{z,d} \times T_{x,a} \times T_{x,\alpha} \quad (4.2)$$

To form the homogeneous transformation matrix for joint i (i =1,2,....,n), the position and orientation of the i th coordinate frame with respect to the previous one ($i-1$) can be specified in equation 4.3 as given below:

$$T_{i-1}^{i} = \begin{bmatrix} R_{i-1}^{i} & \vdots & d_{i-1}^{i} \\ 0\ \ 0\ \ 0 & & 1 \end{bmatrix} \quad (4.3)$$

The matrix R_{i-1}^{i} represents the orientation of frame O_i shown in Figure 4.3 with respect to frame O_{i-1}, therefore

$$R_{i-1}^{i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \quad (4.4)$$

The vector d_{i-1}^{i} describes the position of the origin of frame O_i with respect to frame O_{i-1}, therefore

$$d_{i-1}^{i} = \begin{bmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ d_i \end{bmatrix} \quad (4.5)$$

Finally the homogeneous transformation matrix for joint *i* takes the following general format

$$T_{i-1}^{i} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

The FK can be computed as a product of homogeneous transformations between the circumjacent coordinate frames as shown in Figure 4.4 below.

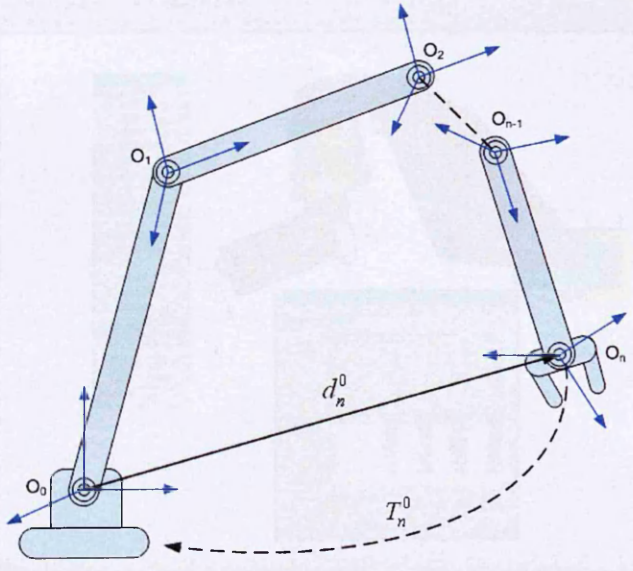


Figure 4.4 Transformation from end-effector (tip) frame to base frame

The following equation, equation 4.7, can be used when computing the homogenous transformation illustrated in Figure 4.4:

$$T_0^n(\theta) = T_1^0(\theta)T_2^1(\theta)T_3^2(\theta) \dots \dots \dots T_n^{n-1} = \begin{bmatrix} R_n^0(\theta) & d_n^0(\theta) \\ 0_{1\times 3} & 1 \end{bmatrix} \quad (4.7)$$

The orientation and position of the tip coordinate frame O_n with respect to the base frame O_0 is determined by R_n^0 and d_n^0, respectively. In general, both R_n^0 and d_n^0 nonlinearity depend on the generalised coordinates, it is thus not always possible to explicitly express θ in terms of the tip position and orientation coordinates. Consequently there is no general closed-form representation of the IK and numerical techniques are often used to solve IK (Sciavicco and Siciliano 1996).

4.2.2 Determining the D-H parameters for Hydro-Lek arm

The Hydro-Lek arm has six joints all of which are revolute joints. Each of these joints has a θ value of θ^{variable} with i being the joint number. Starting from the base, the joint coordinate frames are assigned. Having established the coordinate frames, the next step is the determination of the D-H parameters.

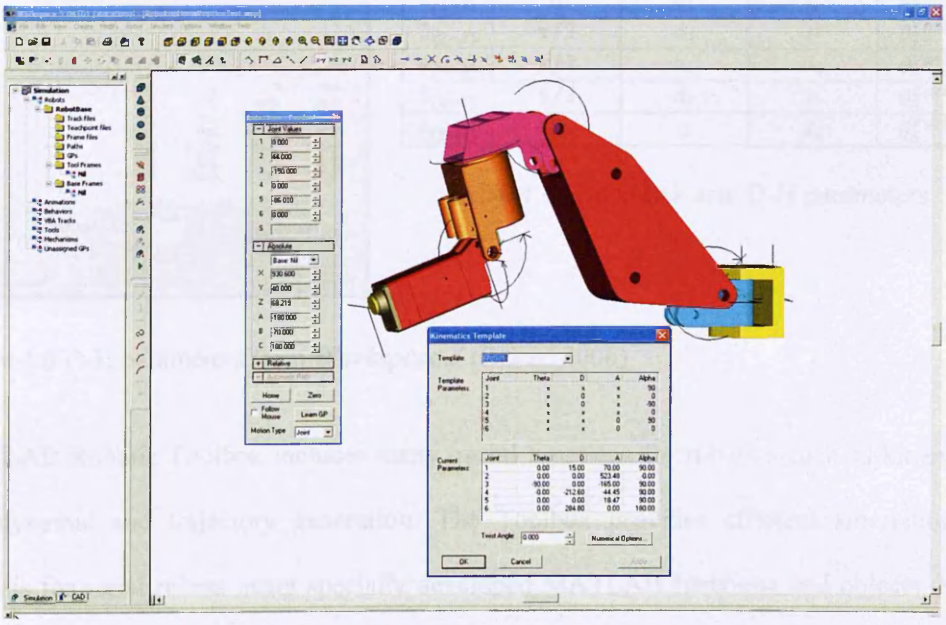
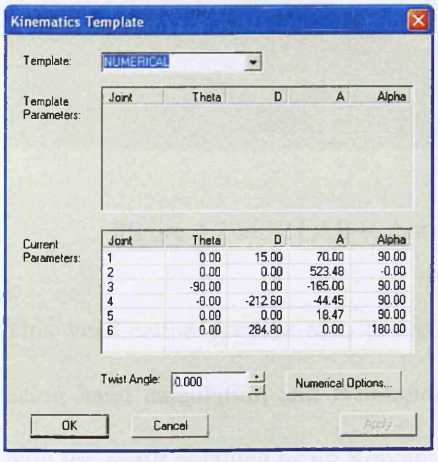


Figure 4.5 ***Worspace 5*** simulation software (Bakari 2005)

Robotic simulation software ***Workspace 5*** was used for this research to obtain the D-H parameters for the Hydro-Lek robot arms. Figure 4.5 shows the Hydro-Lek robot arm without a gripper in ***Workspace 5*** window with the D-H table obtained.

4.2.3 Validating the D-H parameters with other robotic software

The D-H parameters were validated using two robotic simulation software ***Workspace 5*** and MATLAB Robotic Toolbox (Corke 1996). After the Hydro-Lek model was derived it was necessary to establish its accuracy with the D-H parameters obtained using hand calculations that followed the D-H convention rules. Figure 4.6 and Table 4.1 below shows the results of the D-H parameters created for the Hydro-Lek robot arm and its comparison with the use of ***Workspace 5*** robotic simulation software. When these results were compared they were found to match, ***Workspace 5*** therefore proved that the kinematic modelling for the Hydro-Lek arm was correct and valid.



No.	Twist Angle α_i	Link Length a_i	Joint offset d_i	Joint Angle θ_i
$1_{(0-1)}$	$\pi/2$	a_1	d_1	$\theta_1^{Variable}$
$2_{(1-2)}$	0	a_2	0	$\theta_2^{Variable}$
$3_{(2-3)}$	$\pi/2$	a_3	0	$\theta_3^{Variable}$
$4_{(3-4)}$	$\pi/2$	a_4	d_4	$\theta_4^{Variable}$
$5_{(4-5)}$	$\pi/2$	a_5	0	$\theta_5^{Variable}$
$6_{(5-6)}$	π	0	d_6	$\theta_6^{Variable}$

Table 4.1 Hydro-Lek arm D-H parameters

Figure 4.6 D-H parameters from ***Workspace 5*** (Bakari 2006)

MATLAB Robotic Toolbox includes many useful functions for robotics such as kinematics, and dynamic and trajectory generation. The Toolbox provides efficient simulation and analysis for serial robots using specially developed MATLAB functions and objects. In this research MATLAB Robotic Toolbox was used to verify the correctness of the kinematic

modelling and the D-H parameters obtained for the Hydro-Lek robot arm. The functions used in the verification were the FK, IK, and the *drivebot* function which is used to drive a graphical simulation of Hydro-Lek created with the Robotic Toolbox. Figure 4.7 below shows a screen-shot of the HydroLek arm model at zero-position; the drivebot; and the D-H table.

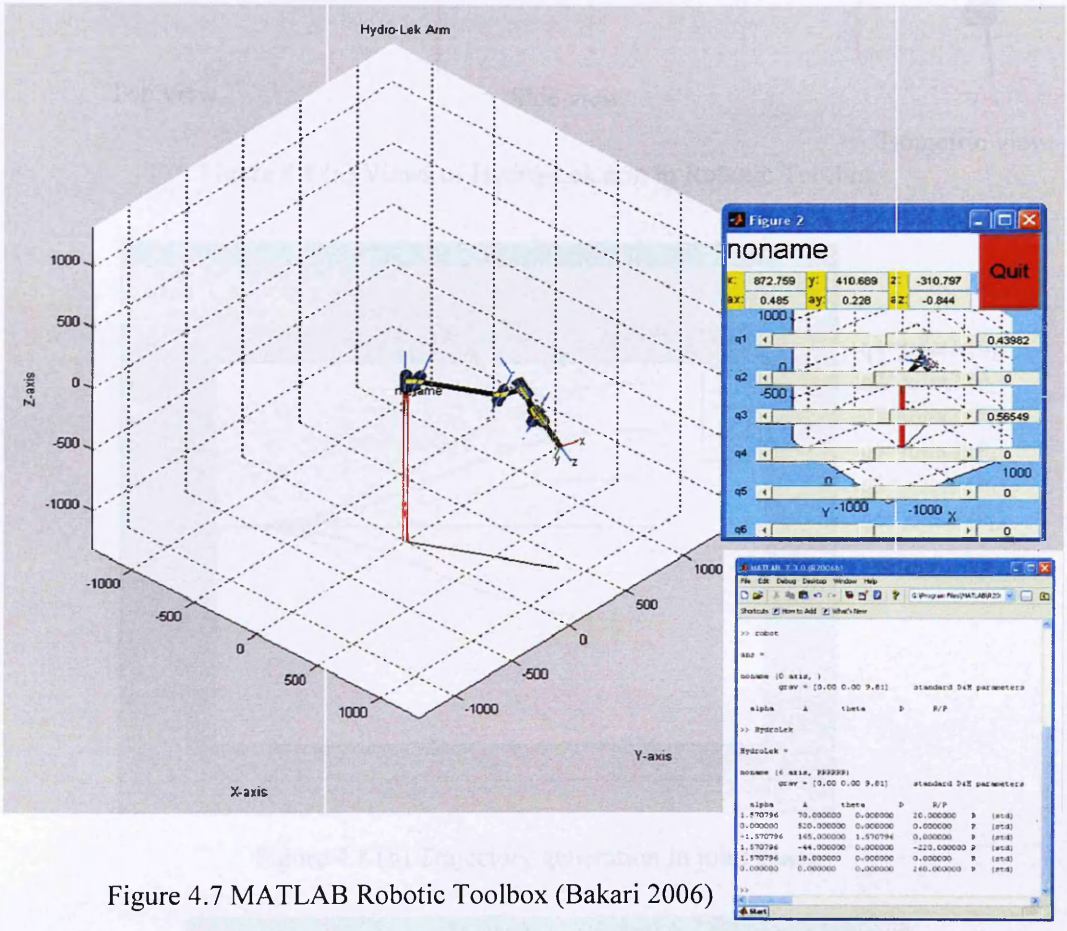


Figure 4.7 MATLAB Robotic Toolbox (Bakari 2006)

This verification process also proved that the kinematic modelling of the HydroLek arms using hand calculation and ***Workspace 5*** simulation software were correct when compared with the result obtained using Robotic Toolbox.

Robotic Toolbox can also be used for trajectory generation in Cartesian and joint spaces. The trajectory generation algorithm used in the toolbox is based on the work of Paul R. (1981). Figures 4.8a, 4.8b and 4.8c show the trajectories in joint and Cartesian spaces from point to point and in a straight line along the x-axis, respectively.

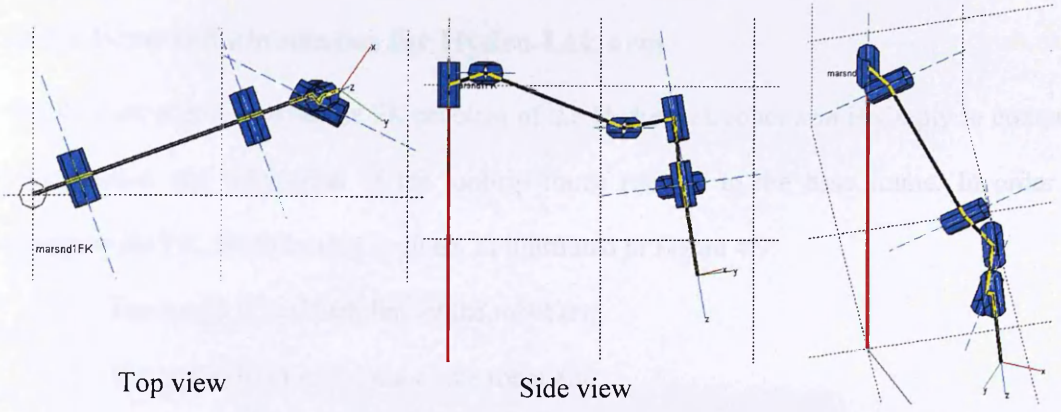
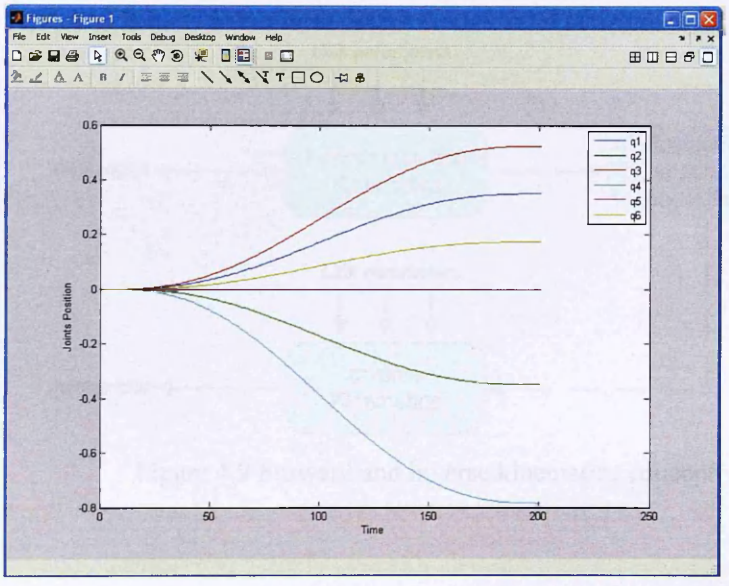


Figure 4.8 (a) Views of Hydro-Lek arm in Robotic Toolbox



q1=joint1 (θ_1)
q2=joint2 (θ_2)
q3=joint3 (θ_3)
q4=joint4 (θ_4)
q5=joint5 (θ_5)
q6=joint6 (θ_6)

Figure 4.8 (b) Trajectory generation in joint space

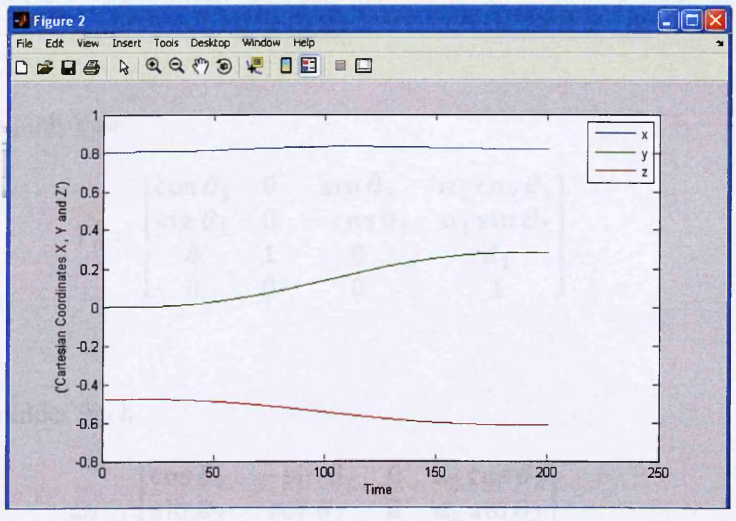


Figure 4.8 (c) Trajectory generation in Cartesian space

4.2.4 Forward kinematics for Hydro-Lek arm

Given a set of joint angles the FK problem of the Hydro-Lek robot arm is simply to compute the position and orientation of the tool-rip frame relative to the base frame. In order to compute the FK, the following is given, as illustrated in Figure 4.9:

1. The length (L) of each link of the robot arm
2. The angle (θ) of each joint of the robot arm

It is then necessary to find the position of any point (x,y,z) coordinate of the end-effector (tool).

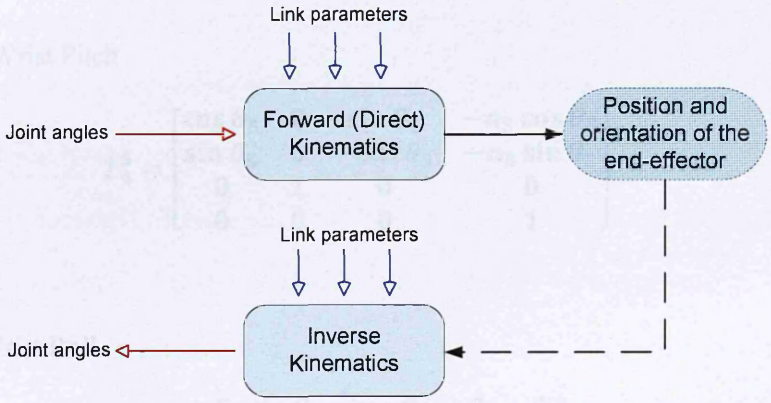


Figure 4.9 Forward and inverse kinematics concept

Link transformations can now be formed using the homogenous transformation matrix in equation 4.6 and the D-H parameters in Table 4.1.

Joint One – Azimuth Yaw

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & a_1\sin\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Joint Two – Shoulder Pitch

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

Joint Three – Elbow Pitch

$$T_2^3 = \begin{bmatrix} \sin\theta_3 & 0 & -\cos\theta_3 & -a_3\sin\theta_3 \\ -\cos\theta_3 & 0 & -\sin\theta_3 & a_3\cos\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

Joint Four – Forearm

$$T_3^4 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & a_4\cos\theta_4 \\ \sin\theta_4 & 0 & -\cos\theta_4 & a_4\sin\theta_4 \\ 0 & 1 & 0 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

Joint Five – Wrist Pitch

$$T_4^5 = \begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 & -a_5\cos\theta_5 \\ \sin\theta_5 & 0 & -\cos\theta_5 & -a_5\sin\theta_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

Joint Six – Wrist Roll

$$T_5^6 = \begin{bmatrix} -\sin\theta_6 & \cos\theta_6 & 0 & 0 \\ \cos\theta_6 & \sin\theta_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

Having determined all six transformation matrices for the Hydro-Lek robot arm, the product of the six-link transform matrices leads to:

$$T_0^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

For which $r_{ij} \forall 1 \leq i,j \leq 3$ represents the elements of the orientation matrix R_0^6 and the vector $\boldsymbol{P} = [p_x \quad p_y \quad p_z]^T$ represents the position of the end-effector with respect to the base

coordinate frame. Further details concerning the FK calculation are discussed in Bakari *et al* (2007).

4.2.5 Inverse kinematics for the Hydro-Lek arm

Given the position of the end-effector, the role of the IK is to compute the angles of each joint of the robot arm. Although solving the IK problem of a robot arm using a closed-form analytic solution is preferable and has many advantages over a numerical solution including faster calculation than numerical iterative procedures, it was not possible to develop a closed-form analytic solution for the Hydro-Lek arm for this research. After a considerable amount of research it did become possible to achieve some sets of analytical equations for some joints of the robot arm but it was not a complete solution because of the existence of coupled position and orientation for some of the joints of the robot arm. This design problem is discussed in more detail in section 4.2.6 below. A detailed description of the research on the development of a closed-form inverse solution is outlined in Appendix D. A numerical solution was implemented, however, through the application of the Jacobian pseudo-inverse method (Kapoor, C., & Tesear, D., 1999; Lewis, F. L., et al 2004; Meredith, M., & Maddock, S., 2004).

4.2.5.1 The Jacobian solution

The Jacobian matrix solution for the HydroLek arm relates the joint velocities in joint space to the end-effector velocity in Cartesian space. The Jacobian matrix is a matrix of differentials in which any differential changes in the end-effector location are caused by differential changes in the joints variables. The Jacobian matrix $\boldsymbol{J(q)}$ is the transformation from end-effector velocity vector $[\dot{\boldsymbol{P}}^{\boldsymbol{T}} \quad \boldsymbol{\omega}^{\boldsymbol{T}}]^{\boldsymbol{T}}$ to joints velocity vector $\dot{\boldsymbol{q}}$. Since the generalised Cartesian velocity vector of the end-effector is composed of two sub-vectors $\dot{\boldsymbol{P}}$ and $\boldsymbol{\omega}$, the Jacobian matrix may be partitioned into linear and orientation parts by writing,

$$\begin{bmatrix} \dot{P} \\ \omega \end{bmatrix} = J(q)\dot{q} = \begin{bmatrix} J_p(q) \\ J_\omega(q) \end{bmatrix} \dot{q} \tag{4.15}$$

For which $\dot{P} = [\dot{p}_x \quad \dot{p}_y \quad \dot{p}_z]^T$ represents the resolved linear velocity of the tool-tip and $\omega = [\omega_x \quad \omega_y \quad \omega_z]^T$ represents the angular velocity. In this matrix the linear position Jacobian $J_p(q)$ represents the first three rows of $J(q)$ and the angular Jacobian $J_\omega(q)$ its last three rows. Thus the arm Jacobian $J(q)$ is $6 \times n$ matrix, with n the number of joints in the manipulator. In this case $n = 6$, therefore the Jacobian of the Hydro-Lek manipulator is square. The computation of the linear position Jacobian can be obtained using the following relation:

$$J_p(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} & \frac{\partial P}{\partial q_2} & \cdots & \frac{\partial P}{\partial q_n} \end{bmatrix} \tag{4.16}$$

The following chaining operation can be used to evaluate the orientation of the arm Jacobian:

$$T_0^i = T_0^1 T_1^2 \ldots\ldots T_{i-1}^i = \begin{bmatrix} R_0^i & P_0^i \\ 0\,0\,0 & 1 \end{bmatrix} \tag{4.17}$$

In this operation R_0^i denotes the rotation matrix of frame i with respect to the base frame, for which $R_0^i = [x_i \quad y_i \quad z_i]$.

The vectors x_i, y_i and z_i represent the x, y and z-axis of frame i in the base coordinate. Since all angular velocities are represented in the same coordinate frame, as represented by the joint rotation $q_i = \theta_i$ which occurs about joint axis z_{i-1}, the angular velocity for joint variable i is given by $z_{i-1}\dot{q}$. The orientation part of the Jacobian matrix therefore, can be constructed taking into account the fact that the prismatic joints do not contribute to the angular velocity of the end-effector. Hence, the orientation part of the Jacobian matrix may be written as:

$$J_\omega(q) = [k_1 z_0 \quad k_2 z_1 \ldots\ldots k_n z_{n-1}] \tag{4.18}$$

Where $\boldsymbol{k_i = 0}$ if $\boldsymbol{q_i}$ is prismatic and $\boldsymbol{k_i = 1}$ if $\boldsymbol{q_i}$ is revolute. Also the vector $\boldsymbol{z_0 = [0 \quad 0 \quad 1]^T}$.

The implementation of IK in this research is based on the Jacobian technique. The objective of this technique is to incrementally change the joint orientations from a stable starting position towards a configuration state that will result in the required end-effector being located at the desired position in work space. The amount of incremental change on each iteration is defined by the relationship between the partial derivatives of the joint angles $\boldsymbol{\theta = [\theta_1 \quad \theta_2 \ldots \ldots \theta_6]^T}$ and the difference between the current location of the end-effector $\boldsymbol{X = [P^T \quad \omega^T]^T}$ and the desired position, $\boldsymbol{X_d = \left[P_d^T \quad \omega_d^T\right]^T}$. The link between these two sets of parameters leads to the following Jacobian system:

$$\boldsymbol{dX = J(\theta)d\theta} \tag{4.19}$$

By rearranging equation 4.19 this is represented as:

$$\boldsymbol{d\theta = J(\theta)^{-1}dX} \tag{4.20}$$

This form transforms the nonlinear system of equation into a linear one that can be solved using iterative steps. This results in a new problem however in that equation 4.20 now requires the inversion of the Jacobian matrix which can be a difficult process. It is therefore preferable to use the right-hand generalised pseudo-inverse (Meredith, M., & Maddock, S., 2004). Generating the pseudo-inverse of the Jacobian can lead however, to inaccuracies in the resulting inverse that need to be reduced. Any inaccuracies of the inverse Jacobian can be detected by multiplying it with the original Jacobian then subtracting the result from the identity matrix. A magnitude error can be determined by taking the second norm of the resulting matrix multiplied by $\boldsymbol{dX}$, see equation 4.23 below. If the error becomes too big, then $\boldsymbol{dX}$ can be decreased until the error falls within acceptable limits.

An overview of the algorithm using iterative IK is as follows:

1. Calculate the difference between the goal position and the actual position of the end-effector:

$$\boldsymbol{dX = X_d - X} \tag{4.21}$$

2. Calculate the Jacobian matrix using the current joint angles
3. Calculate the pseudo-inverse of the Jacobian

$$\boldsymbol{J^+ = J^T(JJ^T)^{-1}} \tag{4.22}$$

4. Determine the error of the pseudo-inverse

$$\boldsymbol{error = \|(I - JJ^{-1})dX\|} \tag{4.23}$$

5. If $\boldsymbol{error} > e$ then

$$\boldsymbol{dX = {dX}/{2}}\text{, go back to step 4} \tag{4.24}$$

6. Calculate the updated value of the joints angle vector θ and use this as the new current value,

$$\boldsymbol{\theta = \theta + J^+ dX} \tag{4.25}$$

7. Use the FK to determine whether the new joints angle vector locate the end-effector close enough to the desired location and orientation. If the solution is adequate then terminate the algorithm, otherwise go back to step 1.

The computational demand of the algorithm is relatively high over a number of iterations.

4.2.6 Issues related to the structure of Hydro-Lek arm

In this research a great deal of time was spent attempting to obtain the closed-form IK solution of the Hydro-Lek arm in order to achieve high accuracy fast computation for real-time control of the Hydro-Lek arm. The problems discussed in section 4.2.5 with respect to the mathematical complexity of the closed-form IK solution were mainly due to the bad design configuration of the Hydro-Lek HLK-7W manipulator. Figure 4.10 (a) shows the Hydro-Lek

HLK-7W arm that was used for this research. Figure 4.10 (c) shows a new recently designed Hydro-Lek arm based on the HLK-7W arm. In this research, the problems discovered in the original Hydro-Lek HLK-7W with regards to the complexity of the closed-form IK, appeared through the application of the robotic simulation software for kinematic modelling. These findings were verified at a later stage when Hydro-Lek provided the design for a modified version of the Hyrdo-Lek HLK-7W arm. The new arm has 7-functions and is similar to the HLK-7W arm, the difference between the two arms is that the new arm has design improvements in terms of flexibility of use; reach; and improved configuration for better control of the arm and ease of kinematic modelling, especially the closed form IK solution. In addition, the hydraulic actuator needed to rotate joint five is modified; and the power and ease of rotation for the new arm is significantly improved. The new arm also eliminates some parameters such as the dimension a_4 as shown in Figure 4.10 (b). This elimination, as shown in Figure 4.10 (d), may prove that the closed form IK of the new Hydro-Lek arm can be achieved.

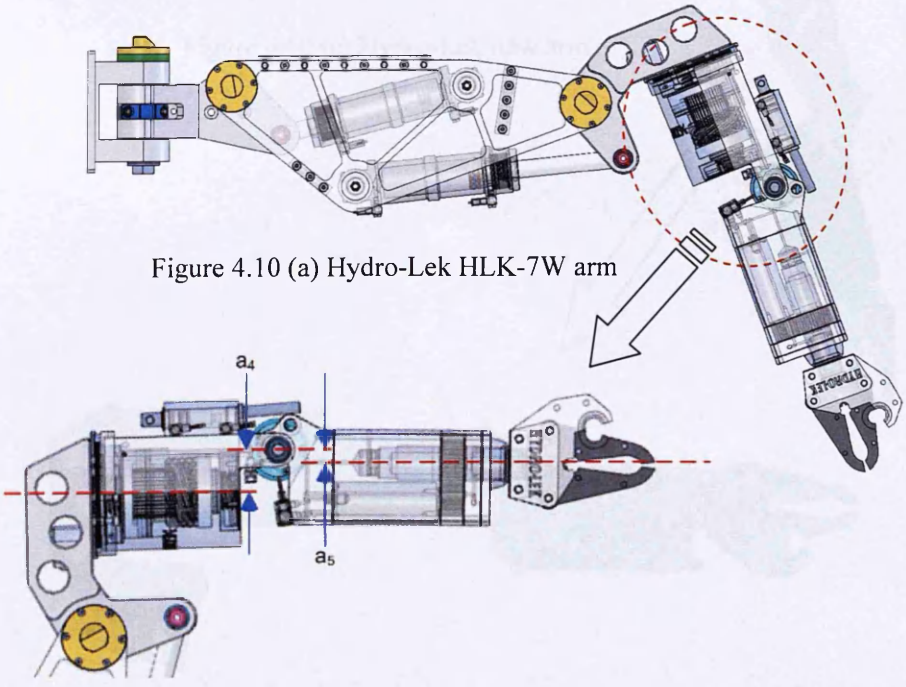


Figure 4.10 (a) Hydro-Lek HLK-7W arm

Figure 4.10 (b) Hydro-Lek HLK-7W arm joints 4 and 5

Through comparison of these two arms it became clear that the Hydro-Lek HLK-7W arm was originally designed without carrying out any kinematic or dynamic modelling using simulation software in order to understand the behaviour of the arm before it was manufactured. This is an example of how important it is to understand the configuration of robot manipulators before manufacturing commences. In addition it is important to also understand their kinematic complexity, especially the IK problems; their control; the power of the actuators; and the types of task that can be carried out by the end-effector. This verification process helps to reduce obstacles, cost and problems after the robot arm is manufactured.

Once all the kinematics and dynamics for a specific robot arm have been developed, it is then possible to design and build the low level controller.

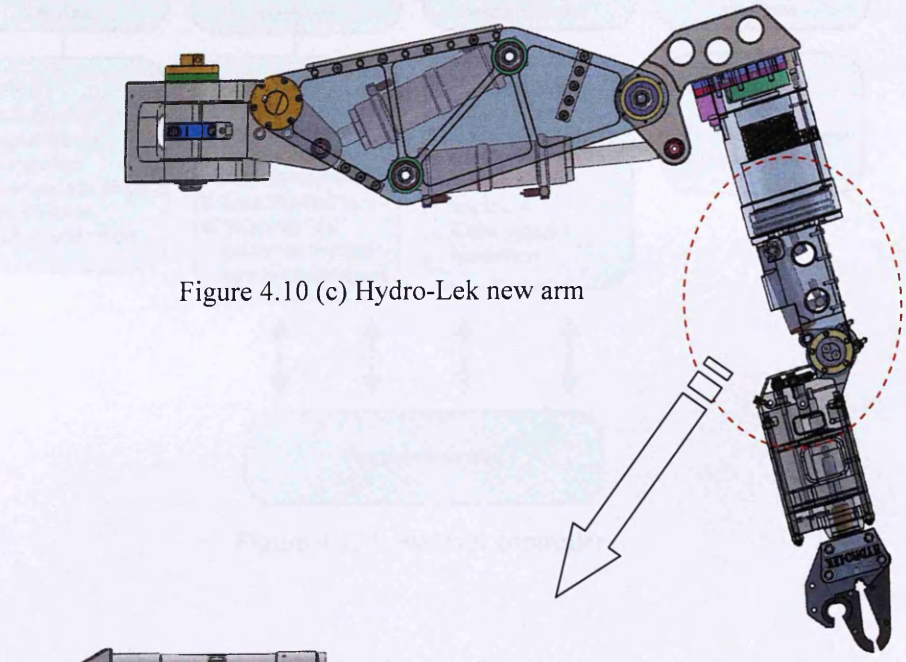


Figure 4.10 (c) Hydro-Lek new arm

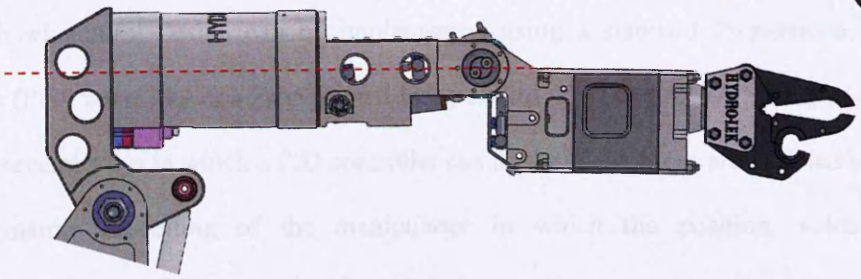


Figure 4.10 (d) Hydro-Lek new arm joints 4 and 5

4.3 Low level controller

The low-level controller receives the required joint angles from the high-level controller. It then sends signals to the robot valves in order to move the robot to the desired position in a well-controlled motion. The high level controller undertakes all the mathematical operations to assign the joint angles needed to move the robot end-effector to a desired position defined by an angular or Cartesian position. The Low-Level controller functional decomposition is illustrated in Figure 4.11. The high level controller functional decomposition is illustrated in Figure 4.12.

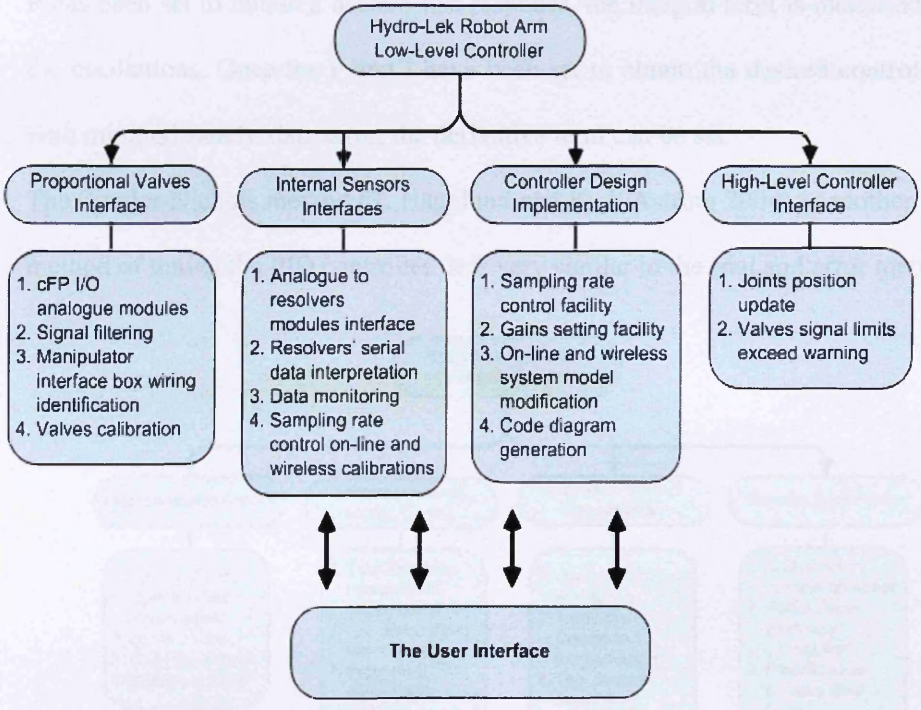


Figure 4.11 Low-level controller

The low-level control design can be implemented using a standard Proportional Integral Derivative (PID) controller or a Proportional Integral Plus (PIP) controller (Young *et al* 1987). There are several ways in which a PID controller can be designed, these are outlined below:

1. Dynamic modelling of the manipulator in which the position, velocity, and acceleration of the joints are mapped into forces. The torque exerted to the structure is based on the Lagrange formulation, which ensures the appropriate structure of the

dynamic model commonly used in control algorithms. Once the dynamic modelling is achieved and all the necessary parameters are defined, then the system identification toolbox of MATLAB software can be used in conjunction with the CAPTAIN toolbox developed by Young *et al* (2003) to obtain the PID tuning needed to form the PID control algorithm.

2. The "guess and check" method is a trial and error method that can be used to obtain the PID gains. In this method, I (integral) and D (derivative) terms are set to zero first and P (the proportional gain) is increased until the output of the loop oscillates. Once P has been set to obtain a desired fast response, the integral term is increased to stop the oscillations. Once the P and I have been set to obtain the desired control system with minimal steady state error, the derivative term can be set.
3. The Ziegler-Nichols method (T. Hägglund and K. J. Astrõm 2004) is another popular method of tuning the PID controller. It is very similar to the trial and error method.

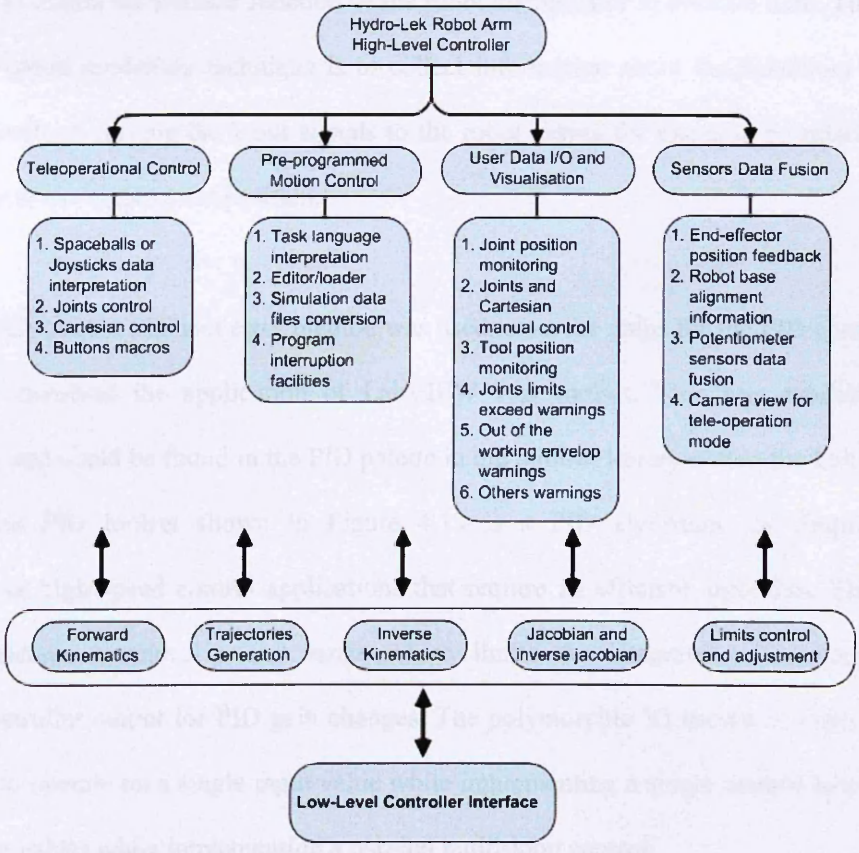


Figure 4.12 High-level controller

A PIP controller has a different process of design to PID. The transfer function of a robot, which represents the system behaviour, can be obtained using a data-based modelling technique. One of the successful techniques is that developed by Young P. C. (1996) which has found a wide range of applications including heavy machinery (Dixon *et al* 1997), intelligent excavator (Gu *et al* 2004) and environmental related applications (Lee *et al* 1998; Taylor *et al* 1998). The PIP control system provides a solution for controlling a system with time delays. It works in the same way as the Smith Predictor (SP) works in the traditional PI/PID digital control systems but in a much more flexible and robust way in changing the design terms. The PIP controller can be considered as an extension of the conventional PI controller in which the PI action is enhanced by higher order forward path and feedback compensators, Taylor *et al* (1998).

The system identification toolbox of MATLAB in conjunction with the CAPTAIN toolbox can be used to obtain the transfer function of the robot manipulator in discrete time. The base for the data-based modelling technique is to collect information about the behaviour of the robot as a result of varying the input signals to the robot valves for example by relating the input voltage to the output joint position.

In this research work a trial and error method was used to set the gains for the PID controller. The process involved the application of LabVIEW PID toolset. This was available for modification and could be found in the PID palette in the built-in library within the LabVIEW software. The PID toolset shown in Figure 4.13 is a PID algorithm for simple PID applications or high speed control applications that require an efficient algorithm. The PID algorithm features a control output range which limits the integrator anti-windup and bumpless controller output for PID gain changes. The polymorphic VI shown in Figure 4.13 can be used to operate on a single input value while implementing a single control loop or an array of input values while implementing a parallel multi-loop control.

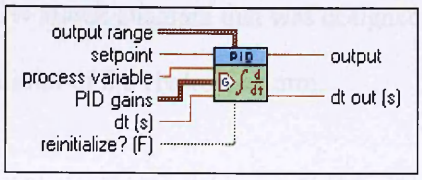


Figure 4.13 PID toolset in LabVIEW

The parameters in the VI are:

Output range: Control output is mapped to the range specified for output high and output low. Default range is -100 to 100.

Setpoint: The setpoint value of the process variable that is being controlled. This is the desired value for the process variable.

Process variable: The measured value of the process variable that is being controlled. This is the feedback value of the feedback control loop.

PID gains: Cluster of proportional gain, integral time, and derivative time parameters.

Dt (s): Interval in seconds at which the VI is called.

Reinitialize? (F): The set of TRUE reinitialise internal parameters, such as integrated error, to default values of 0.

Output: The control output of the PID algorithm that is applied to the controller process.

Dt out (s): Actual time interval in seconds.

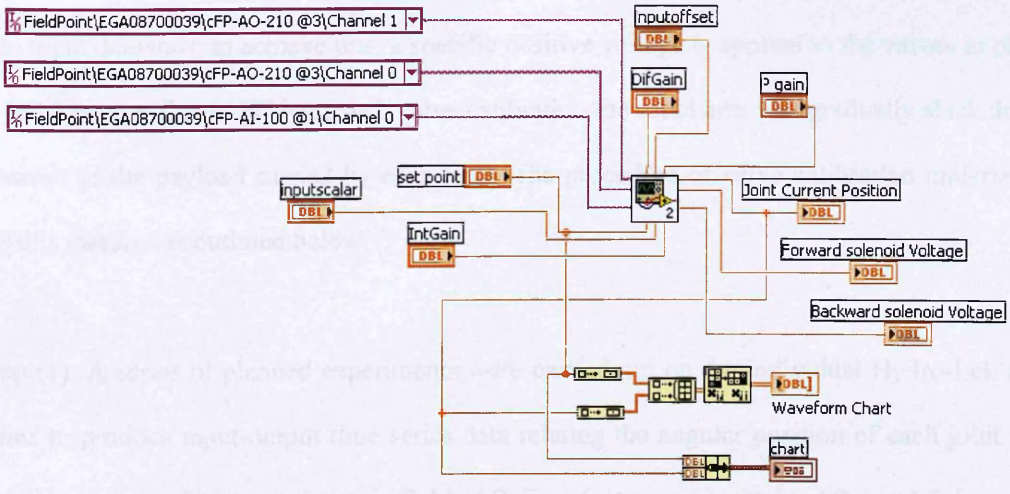


Figure 4.14 LabVIEW Block Diagram to represent the formation of the control loop

Figure 4.14 shows a LabVIEW Block Diagram that was designed containing the modified PID VI for the control of a single joint of the Hydro-Lek arm.

Figure 4.15 shows the LabVIEW Front Panel that was built for the PID gains when the trial and error process was carried out for the individual joint of the Hydro-Lek arm.

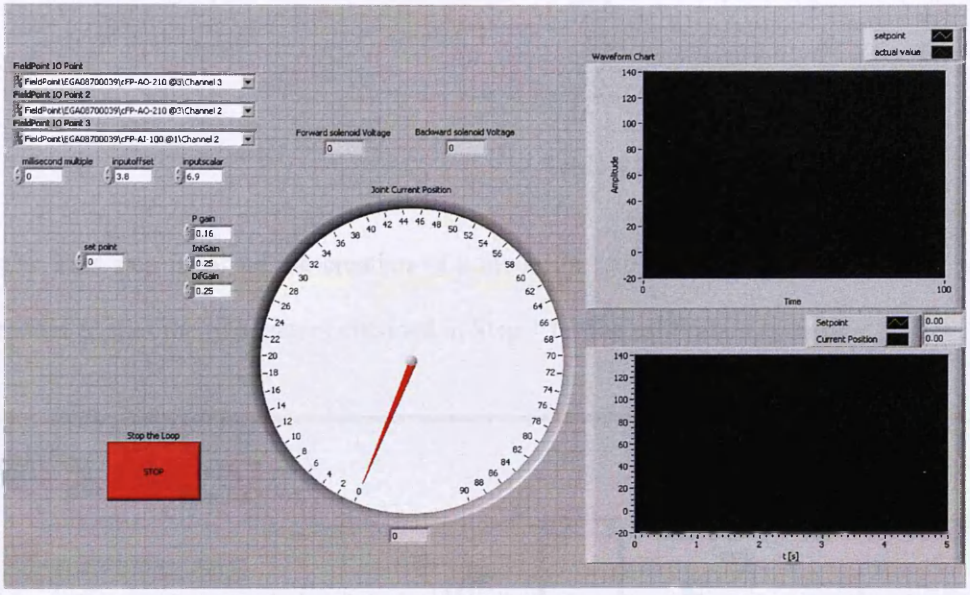


Figure 4.15 LabVIEW Front Panel for the control of PID gains

4.3.1 Valve calibration

In order to help provide the arm joints with meaningful input values it was essential to carry out valve calibration. Valve calibration is based on normalising the input voltage of each joint into input demands; to achieve this, a specific positive voltage is applied to the valves in order to hold the arm fixed. Without such valve calibration the robot arm will gradually slack down because of the payload carried by each joint. The procedure of valve calibration undertaken for this research is outlined below:

Step (1): A series of planned experiments were carried out on the individual Hydro-Lek arm joints to produce input-output time series data relating the angular position of each joint and the valve input voltages, as shown in Table 4.2. Two factors were obtained for each joint.

$$\text{Factor} = \frac{86^\circ - 0^\circ}{16.1 - 3.45} = \frac{86^\circ}{12.65} = 6.8$$

Sensor feedback (Volts)	Angle meter (degree)
3.45	-6
3.90	-4
4.4	0
5.9	11
7.32	21
8.9	30
10.86	45
12.9	58
14.9	70
16	78
16.1	80

Table 4.2 Calibration results

Step (2): This step involved the creation of a model in LabVIEW as shown in Figure 4.16. This model applies the two factors obtained in Step 1 for the calibration process.

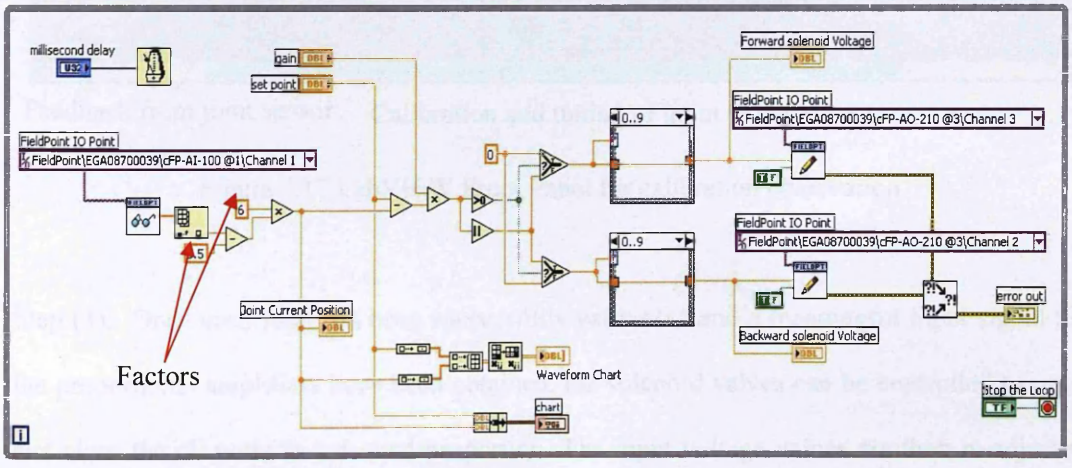


Figure 4.16 LabVIEW Block Diagram to represent the calibration process

A LabVIEW VI model was also created to operate the two proportional amplifiers P02 in order to control the two solenoid valves using set points as shown in Figure 4.17.

Figures 4.16 and 4.17 shows the Block Diagram and Front Panel VIs that were used to calibrate and provide the arm joints with meaningful input values.

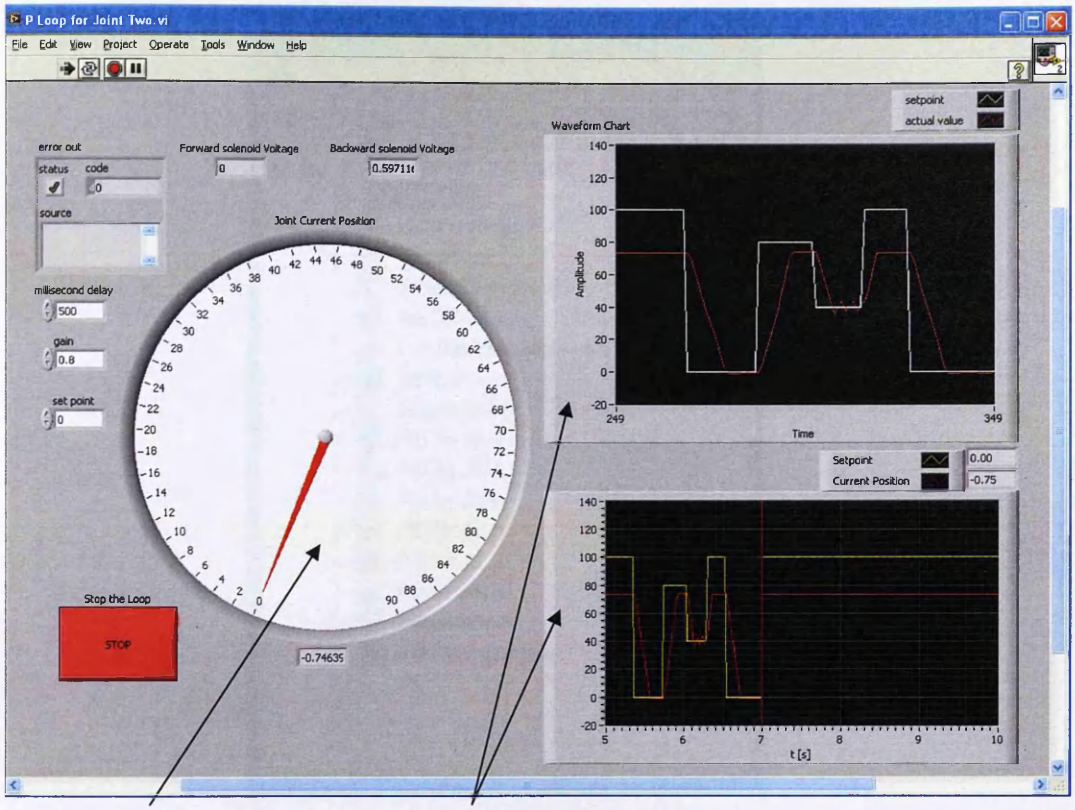


Figure 4.17 LabVIEW Front Panel for calibration observation

Step (3): Once each joint has been successfully calibrated and a meaningful input signal for the proportional amplifiers have been obtained, the solenoid valves can be controlled to open and close the oil ports to a desired proportion. The input voltage values are then re-adjusted again so that small voltage values $\leq$ 1volt become the signal required to operate the joints. This change is achieved by adjusting the dither frequency and amplitude, ramp time, and maximum and minimum current on the amplifier device.

Step (4): This process involves interfacing the following LabVIEW VIs:

- The motion control VI explained in Figure 4.16
- The PID control VI explained in Figure 4.14
- The forward kinematic (FK) VI

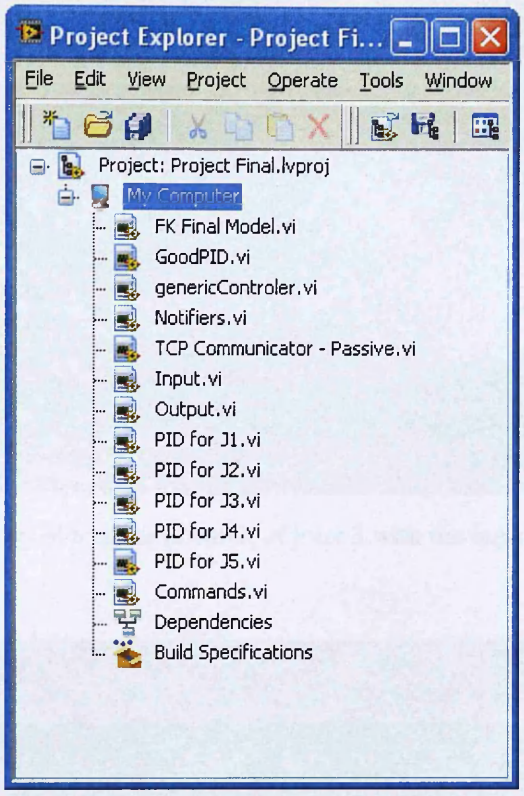


Figure 4.18 LabVIEW project for MARS-ND

The LabVIEW VIs are interfaced into a single project or a single user interface VI where the motion control for the robot joints and the observation of the sensors feedback can be executed. Figure 4.18 shows the LabVIEW project that was built to accommodate the motion, kinematics and third party device communications.

4.3.2 Data collection and analysis

A series of experiments were carried out on the Hydro-Lek arm joints 3 and 4 to produce input-output time series data relating the angular position of joints 3 and 4 and the setpoints. This in turn allowed the researcher to observe the input voltage of the valves in the same LabVIEW GUI. Figures 4.19 and 4.20 show the variation of the angular position of joints 3 and 4 with the setpoints which are translated as an input voltage to the joint valves.

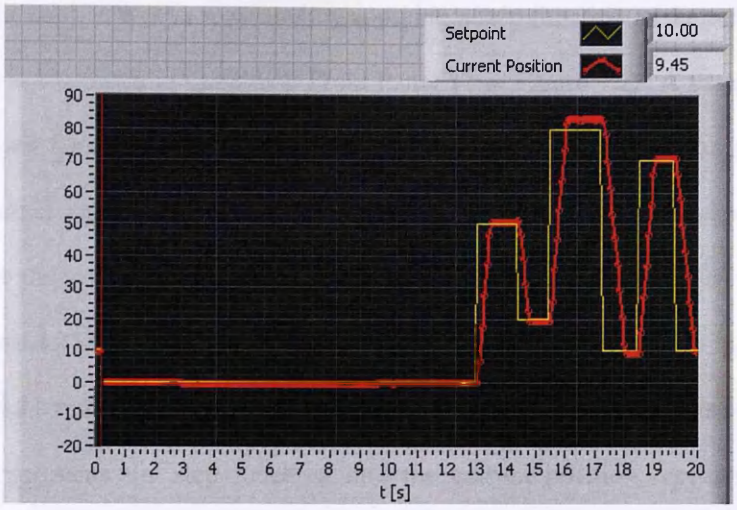


Figure 4.19 Variation of angular position of joint 3 with the input voltage (as setpoints)

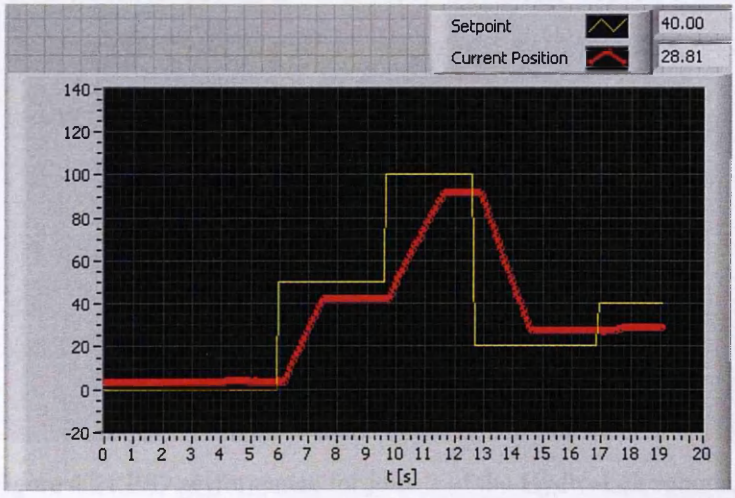


Figure 4.20 Variation of angular position of joint 4 with the input voltage (as setpoints)

4.3.3 PID tuning

The PID control loop in this project was created using the trial and error method as outlined earlier in this thesis. It is a simple PID algorithm which allowed accurate and smooth motion control of the robot joints. The following steps were used to develop the PID controller: firstly to read the sensor feedback signals; secondly to create the desired output signal for the joint actuators from LabVIEW by calculating the proportional, integral, and derivative responses; and thirdly to total those three elements to compute the output signal. The PID control system

performance in LabVIEW was measured by applying a step function as the setpoint command variable. This is illustrated in LabVIEW Front Panel VI in Figure 4.22. A sample time of 1/10 second was used for the PID controller where it samples its input from the analog input NI fieldpoint channel and produces a control output signal on an analog output NI fieldpoint channel. After the setpoint is applied the response of the process variable is measured, as shown in Figure 4.21 below, which in this case is the sensor feedback of the robot manipulator joint. Figure 4.21 represents the PID tuning process for joint 2 of the Hydro-Lek arm where the setpoint represents the step function, and the current position represents the process variable.

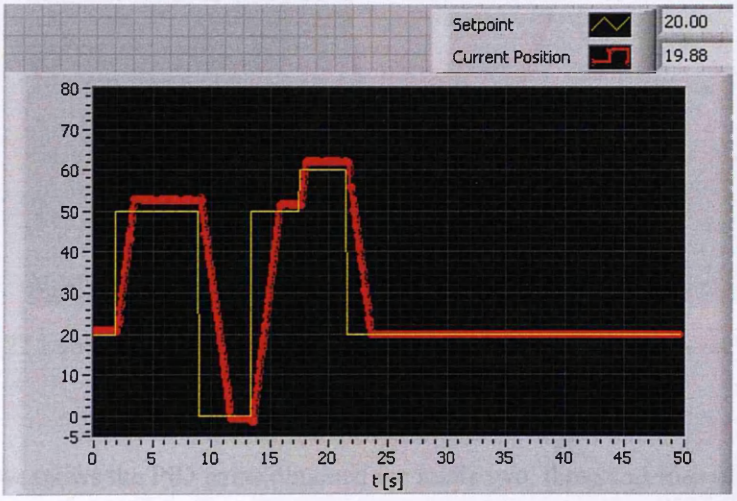


Figure 4.21 PID performance for joint 2 of the Hydro-Lek robot arm

The low level controller design is based on controlling each joint separately or simultaneously, as needed. Each joint of the robot arm is modelled using the trial and error technique. Several tests were carried out for each joint of the arm to evaluate the use of the PID gains and to prove the effectiveness of the low level controller. Figure 4.20 shows the LabVIEW Front Panel model used for the implementation of the PID gains with the robot joints. Figures 4.23 is the graphical representation used to show the plotted data with time delay obtained during the practical experiments for joint three of the Hydro-Lek robot arm. It also show the effectiveness of the PID gains obtained through trial and error for joints two and three. The data collected during the experiments were the setpoint used as the input angles for

the joint actuator, and the feedback signal received from the potentiometer sensor fitted inside the robot joint.

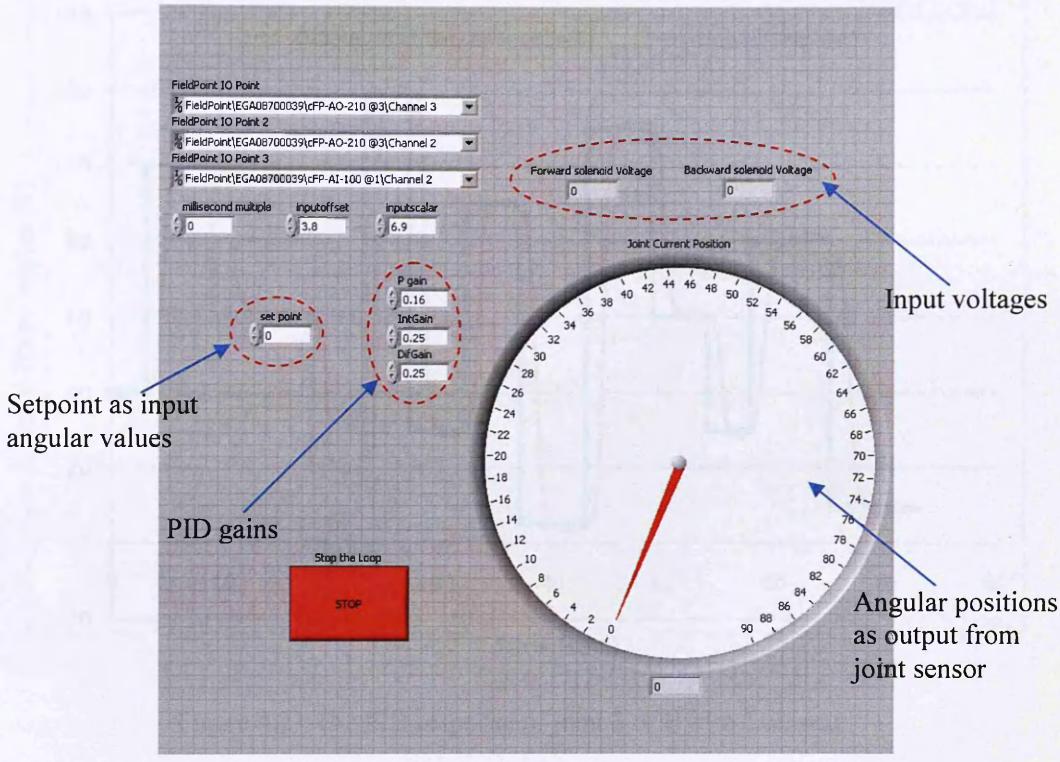


Figure 4.22 LabVIEW Front Panel used for checking the effectiveness of PID gains

The table below shows the PID gains obtained for joints two, three and four of the Hydro-Lek arm. The PID gains were obtained using the trial and error method as discussed above.

Joint name	Proportional gain	Integral gain	Derivative gain
Joint two	0.16	0.25	0.25
Joint three	-0.19	-0.24	0.19
Joint four	-0.75	0.5	0.5

Table 4.3 PID gains for Hydro-Lek joints two, three and four

Figure 4.23 below illustrates the plotted data of the measured response of the process variable of Hydro-Lek arm joint 3, which is quantified by measuring defined waveform characteristics.

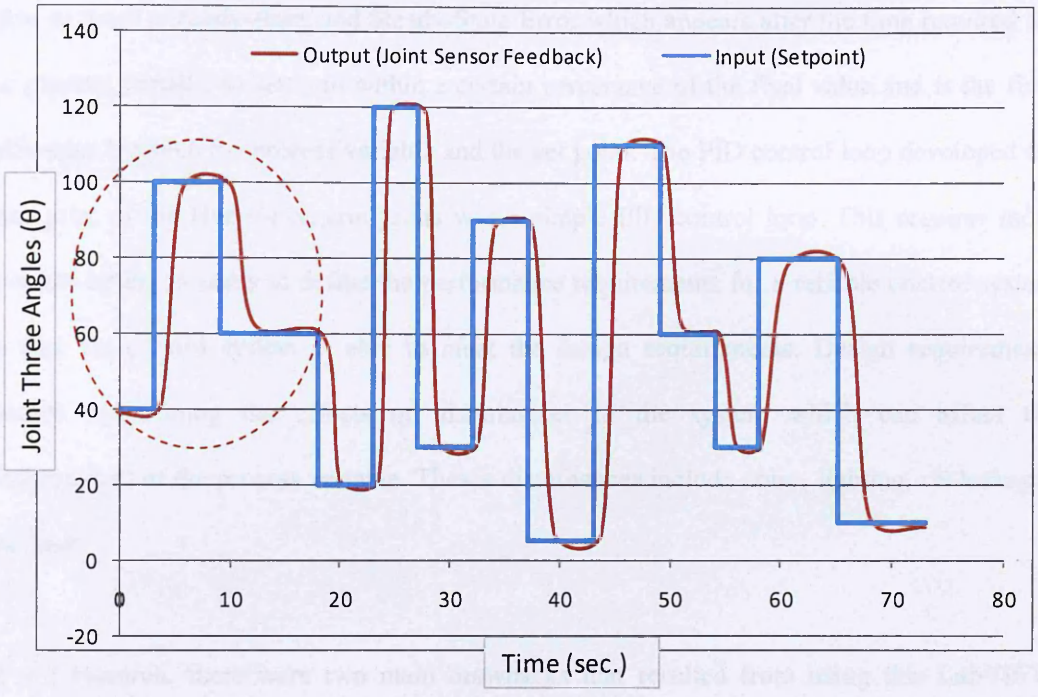


Figure 4.23 The PID response of joint 3 of Hydro-Lek arm

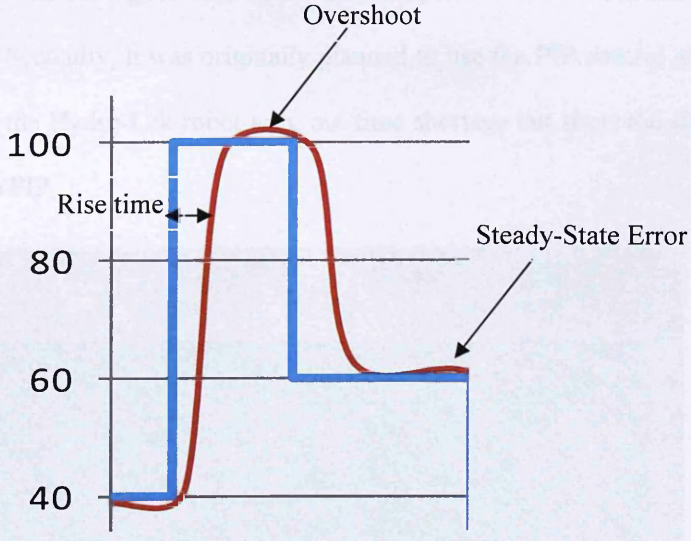


Figure 4.24 The PID response of joint 3 with overshoot, Rise Time and Steady-State Error

Figure 4.24 shows that after the PID gains are tuned, their measured waveform characteristics shows percentage overshoot. This is the amount that the current sensor position (process

variable) overshoots the final value, expressed as a percentage of the final value. Other elements shown in Figure 4.24 include, Rise Time which is the amount of time that the system takes to reach a steady-state; and Steady-State Error which appears after the time required for the process variable to settle to within a certain percentage of the final value and is the final difference between the process variable and the set point. The PID control loop developed for each joint of the Hydro-Lek arm joints was a simple PID control loop. This requires more accurate tuning in order to define the performance requirements for a reliable control system so that the control system is able to meet the design requirements. Design requirements include overcoming the effects of disturbances in the system which can affect the measurement of the process variable. Theses disturbances include noise, lighting, oil leakages and heat.

In this research, there were two main drawbacks that resulted from using this LabVIEW project. Firstly, oil leaked in some of the robot joints; this phenomenon required the PID gains for those joints to be re-tuned. Figure 4.25 shows an example of a result obtained after the oil leakage for one joint. Secondly, it was originally planned to use the PIP control algorithms to control the motion of the Hydro-Lek robot arm, but time shortage cut short the study and the implementation of the PIP.

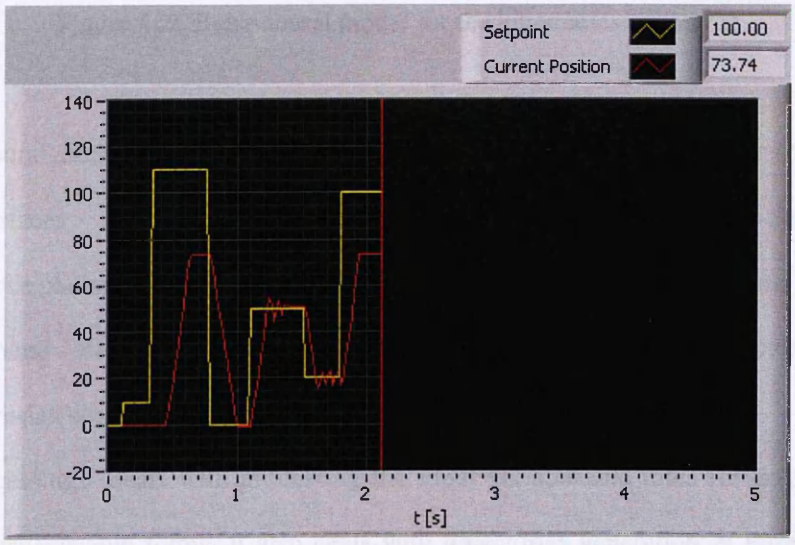


Figure 4.25 Results for disturbed PID gains

4.3.4 Kinematics controller

The kinematics modelling of the robot arm and the design of the low level controller can all be contained within a kinematic controller. The kinematics controller is a software-based module. The behavioural model shown in Figure 4.26 shows the interrelationship between the module components.

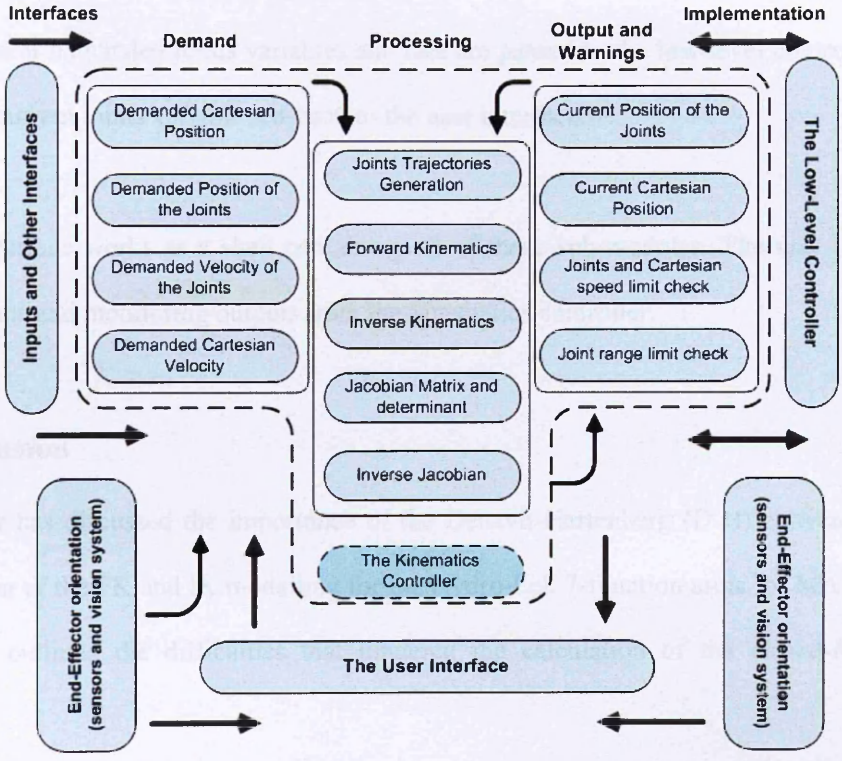


Figure 4.26 Behavioural model for the kinematics controller

The behavioural model allows the division of the kinematics controller into five components:

1. Interfaces - represents interfaces of the input devices such as joysticks, spaceball, mobile phone and other interfaces such as the simulated robot and sensors.
2. Demand - deals with the demanded values of variables and rate in joints space and Cartesian space
3. Processing - deals with processing the input variable and rates for conversion into interchangeable forms. It also deals with trajectories generation in joints space and

Cartesian space. Other functions include singularities detection and the issuing of warnings when limits or ranges are exceeded.

4. Output and warnings – gives values for the joints variables, mainly deals with the outputs to the user. Cartesian variables and rates are monitored and warnings are visualised.
5. Implementation - deals with the low-level controller interface in which data in the form of demanded joints variables and rate are passed to the low-level controller and the current joints variable fed-back to the user interface.

The user interface works as a shell containing all of these sub-modules. The user interface controls inputs and monitoring outputs from the kinematics controller.

4.4 Conclusion

This chapter has discussed the importance of the Denavit-Hartenberg (D-H) convention for the formation of the FK and IK modelling for the Hydro-Lek 7-function arms for MARS-ND. It has also outlined the difficulties that hindered the calculation of the closed-form IK solution.

The Hydro-Lek 7-function arms were purchased with limited technical information. It was therefore necessary to understand the FK and IK of the arms in order to be able to understand their behaviour so that they could be controlled. Through the research process it emerged that the D-H convention is the most popular convention in the field of robotics for modelling the FK and IK of any robot arm. In order to apply this complex convention it was first necessary to understand its theory and application. The D-H parameters were then validated with the use of robotic graphical simulation software which enabled the kinematics of the Hydro-Lek arms to be checked for accuracy.

Following these processes the design of a kinematics controller in LabVIEW provided an interface with external sensors and a simulation package. In this way a specific controller was built for the arms which gave the researcher full control of all aspects of them. This type of controller is called a Low Level Controller, as discussed in this chapter. The development and application of a single motion controller for the Hydro-lek arms with the use of LabVIEW as the operating software is a step forward from previous projects which have used many different controllers and software for the various movements of the robotic arms. Thus the use of LabVIEW and the Low Level Controller simplified the motion control process of the Hydro-lek arms. Valve calibration was undertaken in order to help provide the arm joints with meaningful input values; this process was carried out by normalising the input voltage of each joint into input demands. The low-level controller design is based on the individual joint control strategy. Each joint of the robot arm is modelled using a trial and error method with the PID gains using LabVIEW PID toolkit. The effectiveness of the low level controller for each joint of the robot arm was tested using the motion control loop designed in LabVIEW.

This chapter has also commented on the issues that emerged during the research process in relation to structural problems of the Hydro-Lek HLK-7W arms. These problems became evident because of the intensive research carried out regarding the exact solution of IK. The problems highlight the importance of carrying out FK and IK as part of the design process, before the manufacturing stage; they also demonstrate the importance of rigorous research to the robotics industry.

Chapter 5 will discuss the high level control for the Hydro-Lek arms. It describes the use of SpacePilot to control the motion of each joint or the tool tips of the robot arms; and the problems encountered when using this device. It then outlines the use of an alternative simple third party device, called a USB Joystick, interfaced with the motion control model built with LabVIEW software.

CHAPTER 5

3D GRAPHICAL SIMULATION AND USER INTERFACES FOR HYDRO-LEK ARMS

5.1 Introduction

Remote robotic operations in space, nuclear and undersea environments present challenges that are not usually present in the manufacturing industry where the environment may be controlled. Remote operations in harsh environments require sophisticated and reliable control algorithms capable of adapting in real-time to unexpected events in the workspace (Anthony Lai 2005). Pre-planned, model-based control is insufficient in these environments; instead the manipulator system must be sensor-rich with advanced 3D visualisation for safe, effective and dexterous operations.

Many mobile robots in current use have a degree of autonomy in that they are able to undertake programmed tasks while simultaneously responding to environmental factors. A large proportion of these mobile robots are remotely operated platforms that also have local autonomy (Vajta, L. and Juhasz, T. 2005). A key factor, therefore, in the human-robot interface is realistic visualisation (F. Driewer *et al* 2007). For this reason tele-robotics and robot simulation are usually an interconnected research area. The interactive human control of these tele-robots needs advanced 3D visualisation using novel graphical techniques.

In some cases, robotic motions and behaviour can be pre-recorded so that the operator only needs to satisfy a simple condition to trigger playback of the action. For example, assembly line robots that produce many of the same type of vehicle will repeatedly go through identical motions, making pre-recorded actions a good solution for control. While this works well in static settings, when the environment is dynamic situations may arise where a pre-determined movement may not be appropriate or possible. In such cases human intervention may provide a more desirable or efficient outcome than an autonomous response may suggest. In order for this switch to be as smooth as possible, the operator needs to be able recognise the internal state of robot and direct the robot effectively. Simulation of robotic systems can be used for layout evaluation, feasibility studies, presentations with animation and off-line programming (Sorenti, P. 1997). A significant amount of current research concerns the development of efficient and simple control systems for robotic systems, because robotic systems are complex (Terry L. Huntsberger *et al* 2004).

The objectives of this chapter are to demonstrate how the FK and IK of the Hydro-Lek arms can be obtained using 3D robotic simulation software. The FK and IK can then be used as the basis to build the low-level and high-level controllers for MARS-ND. This chapter also aims to show that the motion control of the Hydro-Lek robot arms can be improved significantly in terms of operator flexibility and control of the robot arms, when third party devices such as SpacePilot and USB joystick are interfaced with the LabVIEW high-level controller of MARS-ND. The advantages and shortfalls of each device are compared and discussed.

This chapter discusses the use of 3D robotic simulation software interfaced with a motion controller and directly controlled via a USB joystick in order to try to control the Hydro-Lek robot system in an efficient and simple manner. As a result of this process the robot arm will immediately respond and take any action that the user requests of it, regardless of the current situation of the robot. While this allows for predictable control of the robot arm, it also allows

the user to inadvertently put the robot arm in undesirable situations, such as a collision with another robot arm for a multi-arm robot system.

5.2 Workspace 5 simulation software

The use of graphical simulation software has many applications in the robotics industry, which can be summarised as follows:

1. Environmental modelling
2. Tools, equipment and robot modelling
3. Motion planning
4. Off-line programming
5. Monitoring and real time control

In this research project, *Workspace 5* (Flow Software Technologies 2005) robot simulation package was used in the simulation of MARS-ND, to obtain the D-H table and for animation purposes. Figure 5.1 shows the 3D model of the Hydro-Lek robot arm in *Workspace*.

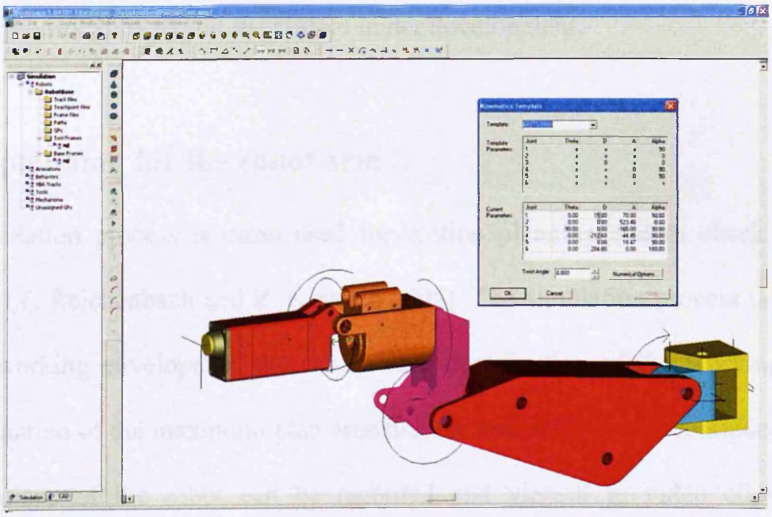


Figure 5.1 Hydro-Lek 3D model in *Workspace 5*

The graphical simulation process of the Hydro-Lek robot arm mainly depends on the D-H convention (Denavit-Hartenberg 1955). The D-H parameters can be derived by constructing a kinematical diagram for the Hydro-Lek arm. As mentioned in Chapter 4 the D-H table

obtained using *Workspace 5* was used for the formation of FK and IK of the arm and to create a high-level motion controller. In the early stages of this research it was decided that the role of the *Workspace 5* simulation process would be to undertake the following processes:

a. Robot design specification
b. Motion planning for the robot arm
c. Environmental modelling for the robot system
d. Task monitoring
e. Off-line programming

5.2.1 Robot design specification

The Hydro-Lek arm is an off-the-shelf arm bought to a fixed specification, it was therefore not necessary to carry out the robot design specification process. In this research, the use of *Workspace 5* helped to understand the design behaviour of the Hydro-Lek arm and its kinematic chains. If the robot arm is designed by the software user, then robot design specification can be a useful step in order to test the adaptability of the robot components and explore possible modification to suit the system under development.

5.2.2 Motion planning for the robot arm

The robotic simulation process is often used for motion planning and to check paths for resolved motion (T. Reichenbach and Z. Kovacic 2005). The simulation process is also used to identify the working envelope of the robot. The identification of the working envelop enables the calculation of the maximum plan area that the end-effector can continuously scan. All planned motions of the robot can be recorded and viewed as video clips creating documentation that can be passed on and used by other professionals. In this research, the *Workspace 5* simulation package was used to record a simple planned motion for the Hydro-Lek arm. Figure 5.2 and 5.3 show the planned motions for the multi-arm system and the moving vehicle Brokk 40.

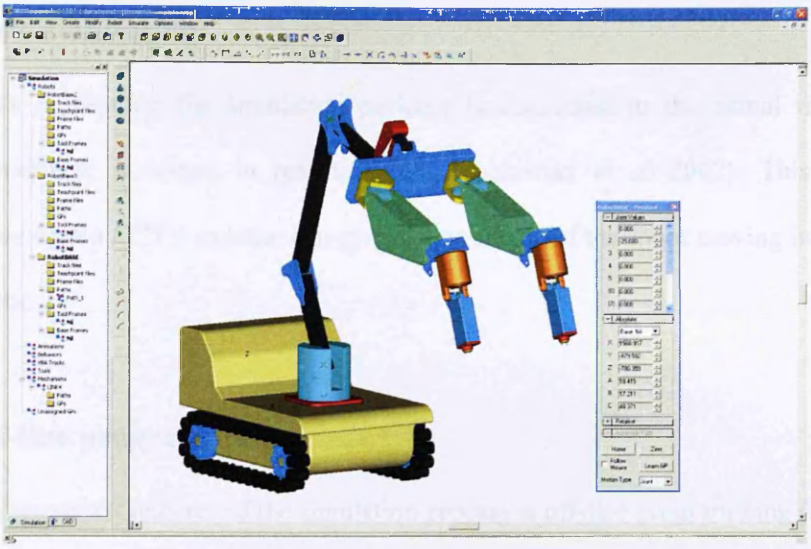


Figure 5.2 MARS-ND 3D model in *Workspace 5*

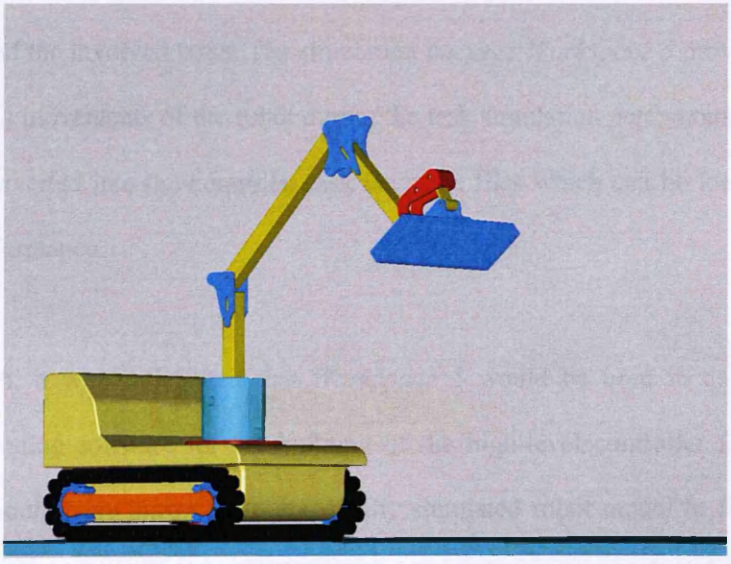


Figure 5.3 Brokk 40 3D model in *Workspace 5*

5.2.3 Environmental modelling of the robot system

Off-line programming for construction tasks requires frequent updating and reprogramming in response to any changes in the working environment. This is time consuming and hard work and for these reasons suggests that it is not feasible to begin every programming task from scratch. Simulation software such as *Workspace 5* allows the conversion of Computer-Aided Design (CAD) drawings to a readable format that the simulation package can implement with the off-line programming process.

5.2.4 Task monitoring

Within task monitoring the simulation package is connected to the actual robot using a suitable interface; it occurs in real time (A. Maslowski *et al* 2002). This process, in combination with a CCTV camera, can give a clear image of the robot moving in the working environment.

5.2.5 Off-line programming

One of the important features of the simulation process is off-line programming (S. F. Chan *et al* 1988). Off-line programming allows instructions to be generated and deployed to the controller through a suitable interface, after satisfactory simulation of the working environment and the involved tasks. The simulation package *Workspace 5* provides track files which record all movements of the robot during the task simulation performance. These track files can be converted into the controller task language files which can be loaded during the actual task performance.

In this research, it was anticipated that *Workspace 5* would be used in conjunction with LabVIEW operating software for the building of the high-level controller for MARS-ND. This process would have allowed the graphically simulated robot model in *Workspace 5* to imitate the actual robot. Previous PhD research at Lancaster University (Zied 2004) successfully implemented the data transfer between LabVIEW and *Workspace 4* simulation package using a specially written Dynamic Link Library. *Workspace 5* was only used in the development of MARS-ND to help obtain the D-H parameters required to find the FK and IK of the Hydro-Lek arms, and to create 3D animations. This was because *Workspace 5* was not user-friendly when it came to 3D modelling of the Hydro-Lek arm unless the 3D model was imported from another 3D package such as SolidWorks. For this reason an alternative 3D simulation package SolidWorks CosmosMotion was explored with the use of a SpacePilot and joystick. These options are discussed in the following sections.

5.3 SpacePilot interface

Java-based tele-operated applications are often used within robotics particularly in the context of educational equipment or simple robots (F. A. Candelas *et al* 2003, S. Dormido 2003 and G. T. McKee 2003). In the field of industrial robot arms there are fewer applications and generally these are designed for specific tasks, for example those used in car manufacturing (F. A. Candelas *et al* 2003 and R. Marin *et al* 2002). Only a few industrial robot applications are based on an open architecture, which offers the flexibility to change the robot being used or to add new robots without modifying either the user-interface or the architecture of the system (A. Aditya & B. Riyanto 2000 and K. Goldberg & R. Siegwart 2002). With regard to simulation, there are not many Java-based applications for industrial robots that offer a realistic virtual environment. The majority represent only wired-models or simplified structures.

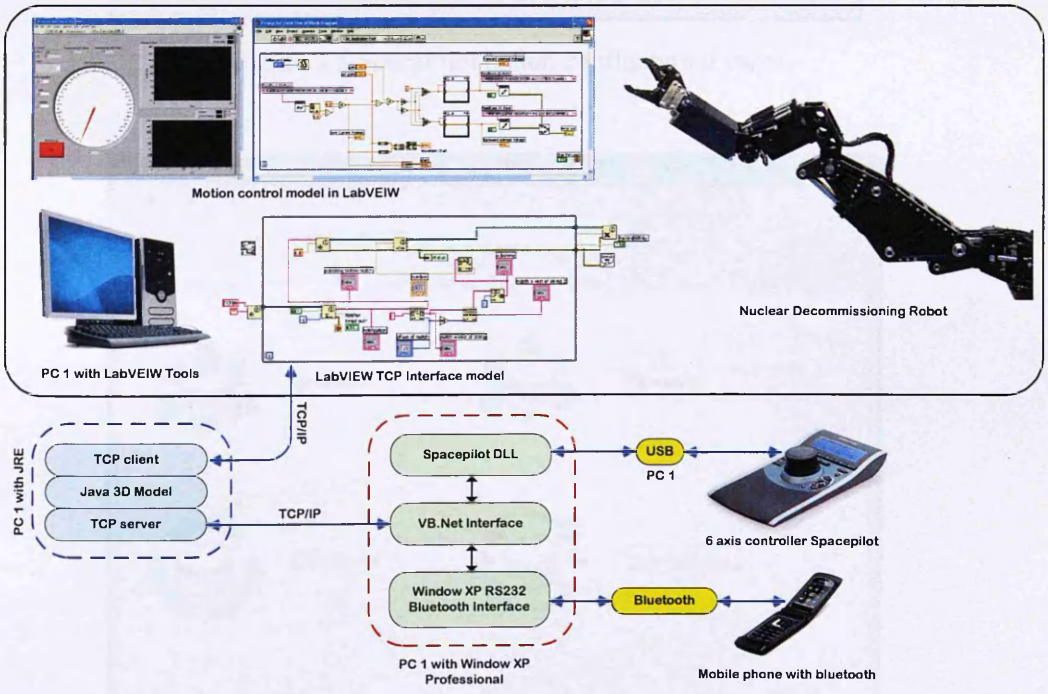


Figure 5.4 High-level controller VI

In this research, an attempt was made to interface a SpacePilot device (3D Connexion 2005) with the high-level controller VI models in LabVIEW (Edward Robertshaw 2008), as shown

in Figure 5.4. Figure 5.5 and Figure 5.6 shows the SpacePilot button configuration and application configuration.

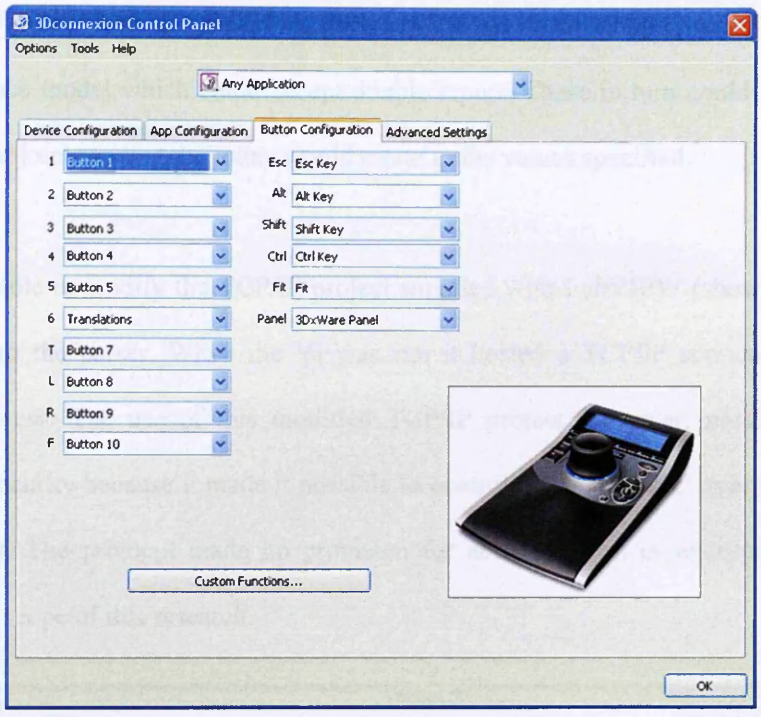


Figure 5.5 SpacePilot button configuration panel

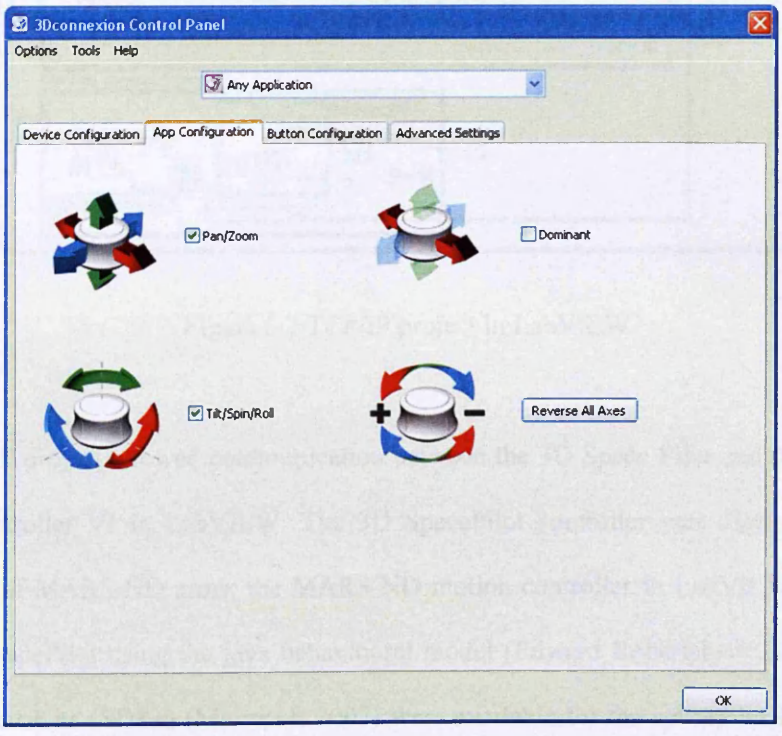


Figure 5.6 SpacePilot application configuration panel

A LabVIEW VI code was created and used to parse a string. It was decided that the MARS-ND system would use named joints, with the name equal to the names used in the LabVIEW behavioural model for each joint. This gave MARS-ND notifier objects within the LabVIEW TCP interface model which could accept double inputs. These in turn could act as set points for the robot joints so that the joints would rotate to the values specified.

It was possible to modify the TCP/IP project supplied with LabVIEW (shown in Figure 5.7) to input into the parser. When the VI was run it hosted a TCP/IP service which allowed external access. The use of this modified TCP/IP project, however, presented a concern related to security because it made it possible to control the MARS-ND over the network and the Internet. The protocol made no provision for authentication or encryption as this was beyond the scope of this research.

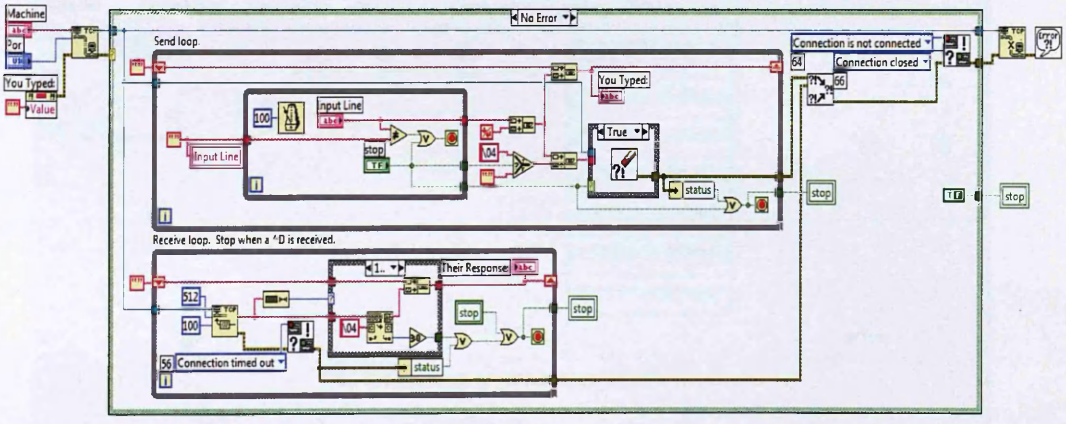


Figure 5.7 TCP/IP project in LabVIEW

This TCP/IP project allowed communication between the 3D Space Pilot and the MARS-ND motion controller VI in LabVIEW. The 3D SpacePilot controller was used to control the movement of MARS-ND arms; the MARS-ND motion controller in LabVIEW was bridged with the SpacePilot using the java behavioural model (Edward Robertshaw 2008). Software Development Kits (SDKs) (Microsoft 2007) were available for the spacepilot in C++, visual basic.NET and java. For this research visual basic.NET was selected and linked to the Java

behavioural model using the TCP/IP interface, which acted as the external interface for the application.

Following the above explorations, it was decided to keep the control of the robot arms as simple as possible. The task of the control became to move the setpoint for the robot arms in 3D space. This implementation proved to be successful but there were limitations of the system at this point that concerned the graphical user interface (GUI) and the rate of response of the MARS-ND arms. Although it was possible to control the MARS-ND arms relative to the inputs of the SpacePilot, there was a lack of feedback from the robot joint sensors. This created logistical problems relating to the fine tuning of the design as it was difficult to relate the physical location of MARS-ND to the GUI diagram, as shown in Figure 5.8.

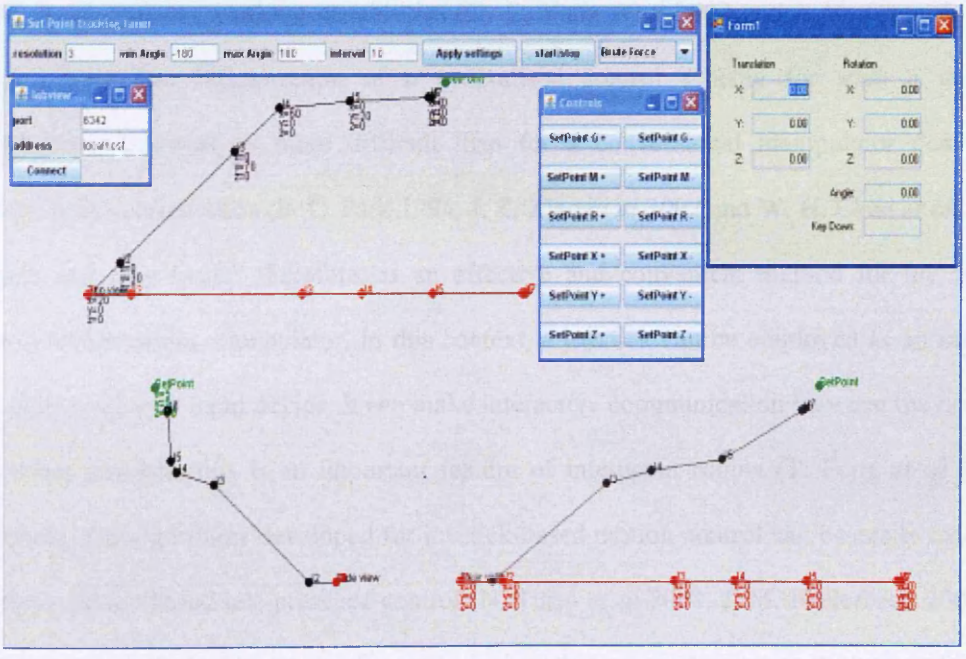


Figure 5.8 Java 3D model GUI

The disadvantages and difficulties encountered in this research in the interfacing of SpacePilot with LabVIEW for motion control are summarised below:

1. LabVIEW did not understand SpacePilot as a third party device, therefore a modified LabVIEW TCP/IP project was needed to communicate with the SpacePilot

2. Non friendly use of Java 3D model GUI
3. Difficulties in tuning the feedback from the MARS-ND arms with the 3D Java model
4. Slow rate of response of the MARS-ND arms when used with the Java model due to the limited computer memory available for the PC used
5. A more powerful PC was needed to simulate the Java 3D model as Java 3D model slowed down the PC

5.4 Joystick interface

A modular reconfigurable robot system consists of a collection of robot modules such as actuators, rigid links, and end-effectors (Yang *et al* 2002, C. J. J. Paredis 1996 and Chen *et al* 1999). These modular components can be assembled rapidly into various robot configurations which have different working capabilities (G. L. Yang *et al* 2002 and I. M. Chen & G. L. Yang 1998). The formalisation of a generalised control scheme for such a modular manipulator, however, is more difficult than for a conventional manipulator due to its flexibility in configuration (F. C. Park 1994, J. Z. Xiao *et al* 2002 and W. H. Chen *et al* 2002). "Teach and play back," therefore, is an effective and convenient method for the motion control of a modular manipulator. In this context, a joystick can be employed as an intuitive position or velocity input device. It can make interactive communication between the operator and robot possible; this is an important feature of intelligent robots (T. Fong *et al* 2003). Moreover, the algorithms developed for joystick-based motion control can be easily extended to haptic device based tele-presence control (N. Turro *et al* 2001, J. M. Hollerbach 2000 and M. Girone *et al* 2001). Haptic interfaces are devices that can communicate the sense of human touch through a multitude of sensors to a robot system, allowing the robot system to simulate the human movements. In tele-presence control, a human operator interfaces with a robot via visual, auditory and force feedback as a form of remote control for the robotic system. It allows control of a robot in difficult situations.

A joystick-based motion control can be realised in either joint space or Cartesian space. Joint-space motion control is relatively easy and straightforward as it does not need kinematics models so that it is independent of manipulator configurations. The major drawback of joint-space motion control however, is that the operator has no feeling for the end-effector motions in Cartesian space. It is therefore not possible to achieve accurate position controls in Cartesian space. Cartesian space motion control is the control of the robot end-effector with the use of the robot kinematics models. The advantage of Cartesian space motion control is that the operator can have a sense of the location of the robot end-effector.

In this research a commercially available USB joystick was employed as a motion input device. Neither the system's design nor the operating software was tied to a particular joystick. The joystick selected for the research has twelve programmable buttons and four controllable axes which are operated through the stick handle and a throttle, as shown in Figure 5.9.

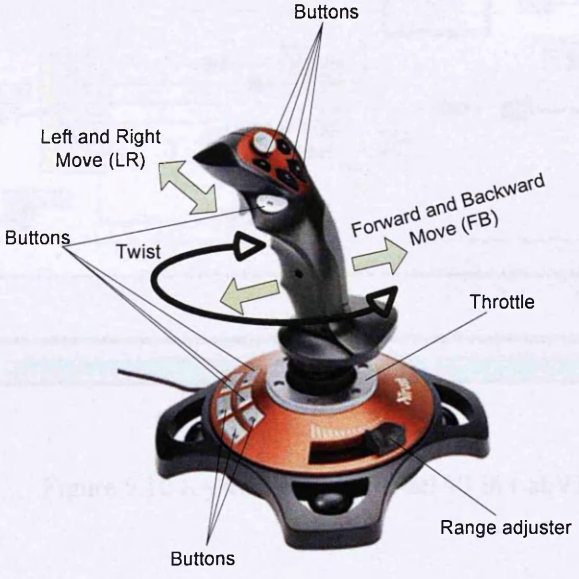


Figure 5.9 USB joystick

The stick handle has three DOFs which include the left-right motion (LR), forward-backward motion (FB), and twist motion (T). For Cartesian space control, LR move, FB move and T can be used to control the manipulator motion (translation and rotation) about X, Y, and Z-axis,

respectively. For joint space control, only the forward-backward motion of the stick handle is used to control the selected joint. The throttle is used to perform fine step control because it can capture small input. For this research, an existing LabVIEW joystick VI model was modified to accommodate the joystick buttons for joint space control. Six of the twelve programmable buttons were used to control the Hydro-Lek joints. Each button was programmed to control the motion of the designated joint. Each time one of the six buttons was pressed then two of the four controllable axes were used to move the selected joint to a desired degree. The joystick control interface created in LabVIEW software for this research is illustrated in Figure 5.10.

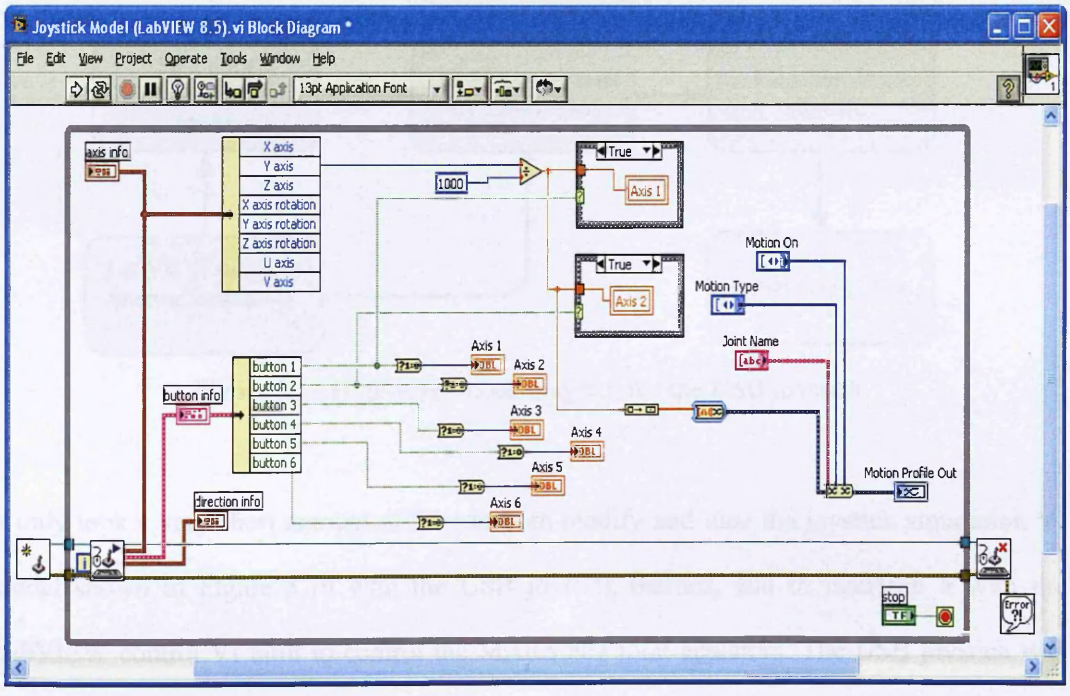


Figure 5.10 Joystick control model VI in LabVIEW

The application of a USB joystick in this research allowed the limitations and difficulties faced using the SpacePilot to be overcome. LabVIEW software recognised the USB joystick as a plug and play device. In addition, there was a simulation VI model available in the LabVIEW built-in library to communicate with devices such as a PC mouse, keyboard, and third party devices such as a joystick. This VI model was adopted and modified for this

research in order to recognise and communicate with the USB joystick. Six of the joystick's twelve buttons were interfaced and tuned with the joystick simulation VI model. Each of these buttons was reprogrammed to rotate each joint of the Hydro-Lek arm with the use of FB motion of the joystick stick handle. Figure 5.11 shows the high-level block diagram for the joint space and Cartesian space controls. Inputs to the robot arm system consist of position and velocity which are delivered via the USB joystick and the graphical user interface within the LabVIEW MARS-ND project. These are represented by the internal commands.

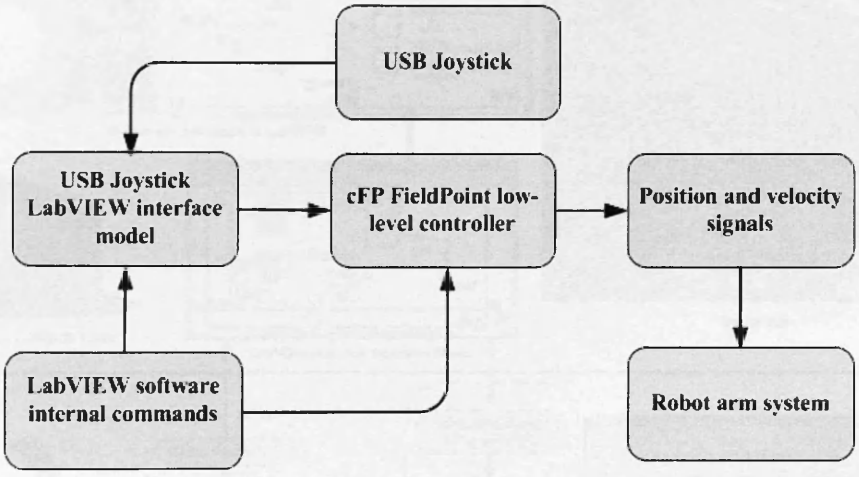


Figure 5.11 High-level block diagram for the USB joystick

It only took a very short amount of time to both modify and tune the joystick simulation VI model shown in Figure 5.10 with the USB joystick buttons, and to interface it with the LabVIEW control VI built to control the MARS-ND joint actuators. The USB joystick was programmed and limited to control only the joint space of the Hydro-Lek arm. It was not possible to implement the end-effector motions in Cartesian space for the following reasons:

1. The wrist of the Hydro-Lek arms had no feedback sensor, it was therefore difficult to know the position and orientation of this wrist joint
2. Incomplete information of kinematic models for the Hydro-Lek arms in the LabVIEW user interface control VI because of step 1
3. The operator did not have a complete understanding of the end-effector motions in Cartesian space because of step 2

In order to overcome the obstacles faced when using the 3D java model of the robot arms, a SolidWorks 3D model of the Hydro-Lek arms was used instead. The SolidWorks 3D model was interfaced with LabVIEW software using a newly designed tool called NI LabVIEW-SolidWorks Mechatronics Toolkit. This toolkit is a user friendly GUI compared with the Java 3D model. Figure 5.12 below shows the layout of the USB joystick interfaced with the LabVIEW control models and SolidWorks 2007.

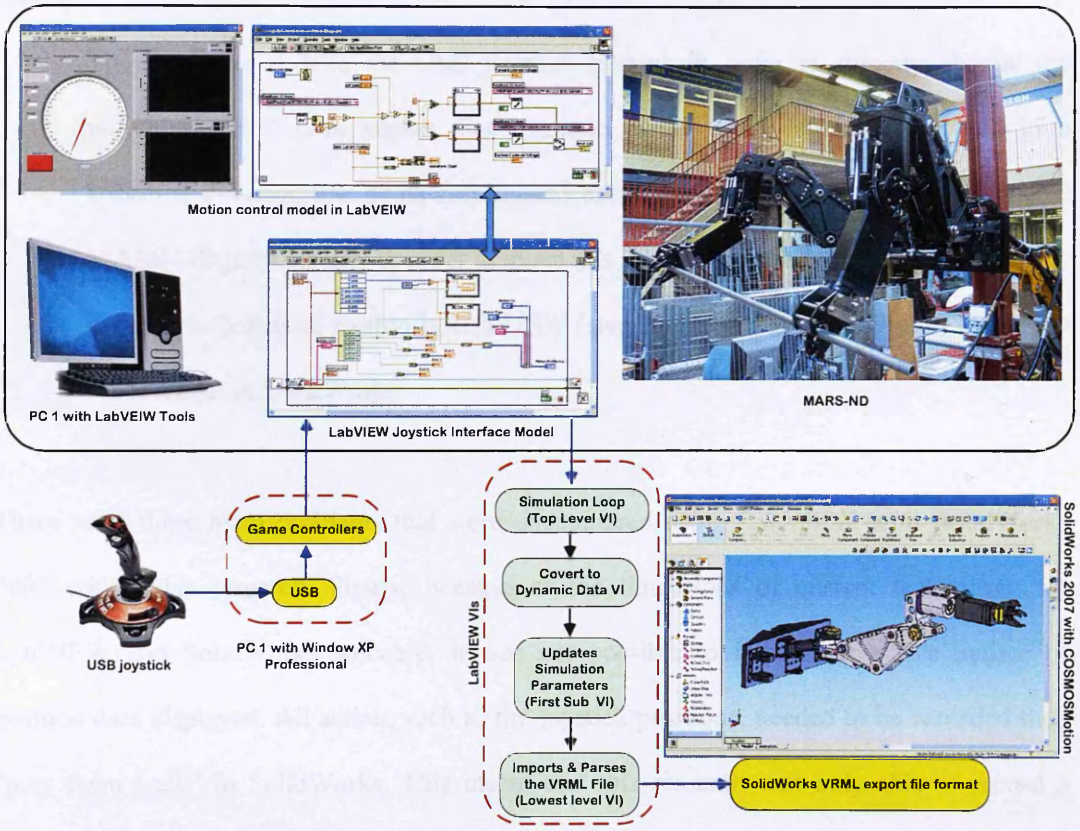


Figure 5.12 Joystick and 3D SolidWorks interface with LabVIEW

In order to create LabVIEW simulations, the SolidWorks robot models are saved as Virtual Reality Modelling Language ($VRML_{97}$) file formats and then exported to LabVIEW. In order to perform a LabVIEW simulation with 3D visualisation provided by SolidWorks $VRML_{97}$ export, it was necessary to create a program that contained three VI's. These included:

1. Top level VI which contained the simulation loop and the state-space matrices information

2. First sub VI that drew and updated the picture in accordance with the simulation parameters
3. Third and lowest level VI which imported and parsed the $VRML_{97}$ file to create objects names and parent/child relationships for each object in the robot system

NI LabVIEW-SolidWorks Mechatronics Toolkit was used in this research for the following reasons:

1. To coordinate it with the USB joystick buttons in order to directly change the individual robot joint angles. For example, when the joystick was moved in a specified direction one of its coordinated buttons fed and controlled the designated MARS-ND joint directly in order to move this joint to a desired angle
2. To map each axis of motion in LabVIEW (step 1) to the constrained joints of the 3D robot model in SolidWorks

There were three main problems that were encountered using LabVIEW with SolidWorks 2007 within this research. Firstly, because of the limitations of current technology for LabVIEW and SolidWorks software, it was not possible to implement a live update of position data displayed. All action, such as the joystick positions, needed to be recorded then "play them back" in SolidWorks. This meant that this research was only able to record a position array using the joystick and then send this motion waveform to SolidWorks. Secondly, SolidWorks 2007 uses arrays or buffers of data rather than a single point at a time. In other words, it was necessary to prerecord all of the motion data first before sending it to SolidWorks, using a "Run COSMOSMotion Simulation.vi". This VI cannot be used in the same loop as the VI that reads the joystick. The joystick data was saved in an array and then sent to SolidWorks 3D model after the joystick loop had finished running. Thirdly, the dynamic data type that was sent had to have the correct time interval (dt) information in it, otherwise the simulation will not play at the correct speed.

5.5 Conclusion

This chapter has discussed the importance of using 3D graphical simulation software such as *Workspace 5* to help develop the FK and IK equations for the Hydro-Lek 7-function arms in order to build the low-level and high-level controllers. It has also outlined the limitations that hindered the use of this software with the real robot. The graphical simulation process in *Workspace 5* depends mainly on the D-H convention. This was used for validation which enabled the kinematics of the Hydro-Lek arms to be checked for accuracy. The Hydro-Lek arms were off-the-shelf components, it was therefore only necessary to understand the FK and IK in order to be able to understand the behaviour of the arms so that they could be controlled.

This research has found that the use of graphical 3D simulation for off-line simulation is very important in the context of a multi-arm robot system, in order to understand this system while undertaking a given task in a specified environment. The control of a multi-arm robot system is more complex than a single arm system. The use of 3D simulation in conjunction with the robot motion controller can significantly enhance the understanding of the nature of the task a multi-arm robot executes. This has particular significance in the context of nuclear decommissioning and the undertaking of tasks within a hazardous environment. 3D simulation can also considerably help the understanding of the robot kinematics and dynamics before the motion controller is established. This aspect is important in the design stage of the robot arms, before they reach the manufacturing stage. This aspect was not employed before the manufacture of the Hyrdro-Lek arms used in this research project, as discovered through this research.

This chapter has also discussed the high level control for the Hydro-Lek arms. It has described the attempted use of SpacePilot to control the motion of each joint or the tool tips of the robot arms and the problems encountered when using this device. It then demonstrated the use of an alternative simple third party device, called a USB Joystick, interfaced with the motion

control model built with LabVIEW software and 3D real-time animation of the robot arm. With regards to the high level control, this research has found that the MARS-ND system can be operated using a USB joystick and 3D simulation software for real-time control and updating of data to the 3D simulation. This type of control, based on sensors, is important in the context of nuclear decommissioning in order to fully understand the movements of the robot arms in real time. It allows cameras and feedback systems to be supplemented by 3D simulation giving a more accurate sense and feeling of the real time movements of the robot within its changing or hazardous environment. Cameras were not used however in this research because of the timeframe and costs.

In this research, the forming of the FK and IK using *Workspace 5* facilitated the building of the low-level and high-level controllers. The interfacing and integration of SpacePilot and a USB joystick in conjunction with LabVIEW operating software establishes a new approach to control issues for multi-arm robot systems in the context of decommissioning tasks. It is important because it allows control of the whole robot arm in any desirable location using a simple mechanism, such as the joystick. This gives the operator flexibility and maximum control of the robot. In addition, the interfacing and integration of the USB joystick in conjunction with LabVIEW operating software allows each arm to be controlled independently by separate joysticks.

In conclusion this research has created a foundation for the control of MARS-ND. This can be developed by other researchers to enable the addition of cameras and a haptic device to further refine the control system in terms of accuracy and reliability of movement.

CHAPTER 6

COLLISION AVOIDANCE ALGORITHM TEST FOR MARS-ND

6.1 Introduction

Most robotic tasks that involve interaction between two parts, such as a multi-arm robotic system, cannot be successfully carried out by relying on purely positional control strategy. For example in the assembly of rigid parts even small path-planning errors and, or control position inaccuracies may induce undesirably large values of contact force. A current issue that is at the forefront of robotics research is the problem of collision-free trajectory planning. Collision avoidance is one of the most important issues of collision-free trajectory planning when operating a system with more than one robot arm. The aim of collision avoidance is to provide control schemes to avoid potential collisions. Several collision avoidance techniques have been developed and applied in a variety of contexts in which robotic systems are employed. These techniques include heuristic algorithms, non-linear programming, configuration space method, artificial potential field algorithms, and kinematics control algorithms.

Collision detection is a subset of collision avoidance. It is concerned with the detection of colliding, or potential colliding, between manipulator links and obstacles; manipulator links themselves; or between two separate manipulators operating close to each other. The aim of collision detection in the context of a multi-arm robotic system is to find the minimum distance between the corresponding objects of each robot arm. Various collision detection

methods are currently used these include, sensor detection or measuring such as laser sensors; ultrasonic; sonar; bumpers; and object visualization using geometrical calculations. There are also various techniques of minimum distance calculation that include, distance between point and point; point and line segment; and between two line segments. In minimum distance calculations the objects and links of the manipulators or obstacles are modelled and represented as points and lines.

The objective of collision detection is to automatically report when a geometric contact is about to occur or has actually occurred. It is typically used in order to simulate the physics of moving objects, or to provide the geometric information which is needed in path planning for robots. Usually the static collision detection problem is studied first and then later extended to a dynamic environment. If the position and orientation of the objects is known in advance the collision detection can be solved as a function of time. A related problem to collision detection is determining the minimum Euclidean distance between two objects. The Euclidean distance between two objects is a natural measurement of proximity for reasoning about their spatial relationship. This chapter discusses the collision avoidance algorithm test for MARS-ND and its limitations. The collision avoidance techniques used by Sugano Laboratory and Kosuge and Hirata Laboratory, and their possible application to MARS-ND were discussed in detail in Chapter 7.

6.2 Previous collision avoidance approaches

The problem of collision avoidance is central to model-based manipulation systems. The simplest collision avoidance algorithms for this type of system fall into generate and test paradigms. A simple path from start to goal, usually a straight line, is hypothesised and then the path is tested for potential collisions. If collisions are detected a new path is proposed, possibly using information about the detected collision to help hypothesise the new path. The three steps in this type of algorithm are:

1. To calculate the volume swept out by the moving object along the proposed path
2. To determine the overlap between the swept volume and the obstacles
3. To propose a new path

An essential component of robot motion planning and collision avoidance is a geometric reasoning system (Lin, M. C., *et al* 1994). This can detect potential contacts and determine the exact collision points between the robot manipulator and the obstacle, or between two manipulators within a specified workspace. Although this system does not provide a complete solution to the path planning and obstacle avoidance problems, it often serves as a good indicator to steer the robot away from its surrounding obstacles before an actual collision occurs. For almost three decades it has been a priority of robotics research to produce paths for robotic devices that avoid collisions by undertaking motion planning (Cameron 1998). Until recently however two central problems restricted the use of motion planning. Firstly, the computation time required to undertake and carry out motion planning limited its usefulness. Secondly, the majority of approaches to motion planning adopted and modified the methods for stationary obstacle avoidance.

In previous approaches to the development of an algorithm for collision avoidance the time-varying obstacle in the configuration time-space (CT-space) is converted to a stationary obstacle. In this approach, therefore, motion planning for time-varying obstacle avoidance is reduced to path planning for stationary obstacle avoidance. Many researchers have examined the context of a single robot with stationary obstacles (Lozano-Perez 1981; Brooks 1983; Luh and Campell 1984; Red and Troung-Cao 1987); there is limited research, however, that is concerned with the problem of a time-varying environment (Lee *et al* 1998) and the issue of collision-free motion planning for a multi-robot system (Yanqiong Fei *et al* 2004). Below I give a brief outline of the development of the key ideas that have led to current thinking in motion planning.

Freund and Hoyer (1985, 1986) formulated the problem of collision avoidance for a multi-robot system as a path-finding problem and suggested an algorithm with simulation results. In 1998 they introduced the concept of a hierarchical coordinator based on the nonlinear control approach, and suggested a real-time collision-free motion planning method for a multi-robot system. Although they provided a practical and systematic approach, they encountered difficulties in the construction of the hierarchical coordinator.

Erdmann and Lozano-Perez (1986) explored the motion planning problem for multiple moving objects using the configuration space-time technique to process time-varying constraints imposed on a moving object. They represented the configuration space-time by using two-dimensional slices and searched for a collision-free path in the space-time.

Lee and Lee (1986, 1987) studied the collision avoidance problem of dual robots. They modelled the robots as spheres and assumed that each robot moved along pre-specified straight-line paths. By constructing a collision map using the path and trajectory information of two robots, they proposed a time scheduling algorithm to modify the velocity profile for the secondary robot. Although this study showed an easy-to-use method for the collision-free motion planning of a multi-robot system, it did not take into account the fact that a collision-free trajectory may not exist on the given path; and did not provide an analytical method to construct a collision map. Chang *et al.* (1994) improved these results by constructing a more accurate geometric model of a robot. They presented a method to obtain heuristically the minimum delay time of one robot. Although they still needed pre-specified paths of two robots to avoid stationary obstacles, they simplified the computational burdens of constructing a complete collision map.

Kant and Zucker (1986) solved the trajectory planning problem in time-varying environments for a point robot. In their approach the problem of planning a collision-free trajectory is decomposed into two sequential sub-problems of path finding with stationary obstacles; and

velocity planning along the chosen path. They first represented potential collisions as forbidden regions in a path-time space by using the path information of a point robot and the trajectory information of moving obstacles. They then presented algorithms to solve the velocity planning problem with different optimality criteria. Kant and Zucker (1986) also developed the path-velocity decomposition technique. In this technique the paths of two robots were independently planned by initially taking only stationary obstacles into consideration. Kant and Zucker then modified the velocity profile of the two robots to avoid collision between them. This approach utilised a stationary obstacle avoidance scheme in the path planning step, but did not consider moving obstacles except for the other robot.

The technique of path-velocity decomposition formed the basis of two further methods. Shin and Bien (1989) developed the concept of coordination space constructed with scalar variables to define the positions along prescribed paths. Bien, Lee and Lee (1992, 1995) proposed an analytical collision-free and time-optimal trajectory planning method for two robot manipulators in the coordination space. The robots were assumed to move along prescribed paths with limited actuator torques and velocities.

The essential concept of the artificial potential field (APF) approach, proposed by O. Khatib (1986) for obstacle avoidance, is to make local decisions at each step based on the distance vectors to the goal and various obstacles. This eventually leads to the goal position. This method treats the robot, represented as a point in configuration space, as particle acting under the influence of a potential field whose local variations are expected to reflect the structure of the free space. The potential function is defined over the free space as the sum of an attractive potential which pulls the robot toward the goal configuration; and a repulsive potential which pushes the robot away from the obstacles to prevent collisions. Virtual forces are defined by negative gradients of potential function. The robot is controlled by the sum of the force moving from high potential configuration to low potential configuration.

Kinematics control algorithms for collision avoidance (Maciejewski and Klein 1985; Zlajpah and Nemec 2002) consider the problem as an inverse kinematics problem. Most of these algorithms were designed for applications which have desired end-effector trajectories throughout the tasks. They solve the inverse kinematics for the angles and angular velocities to satisfy the end-effector constraints and for collision avoidance if the manipulators are redundant. Collision avoidance can therefore be achieved without changing the motion of end-effector, if the manipulators have redundant degrees of freedom.

In 1985 Maciejewski and Klein implemented a kinematics control algorithm for a redundant manipulator to avoid obstacles in dynamically varying environment. Figure 6.1 shows an illustration of the theory they discussed and equation 6.1, gives the equation they used to achieve the goal of collision avoidance.

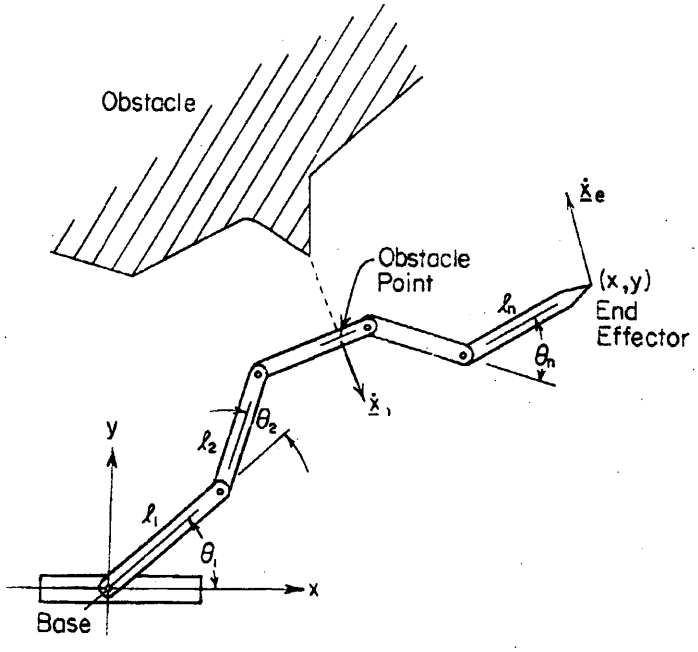


Figure 6.1 Kinematics collision avoidance (Maciejewski and Klein 1985)

$$\dot{\theta} = J_e^{+}\dot{x}_e + [J_0(I - J_e^{+}J_e)]^{+}(\dot{x}_0 - J_0 J_e^{+}\dot{x}_e) \tag{6.1}$$

Where

$\dot{\theta}$: An n-dimensional joint velocities vector, where n is the number of DOF

J: Jacobian matrix (R. Paul 1981)

I: Identity matrix;

J_e: Jacobian matrix of the end-effector

J_0: Jacobian matrix of the collision avoidance point

J_e^+: Pseudoinverse of J_e

$\dot{x}_e$: Velocities vector of the end-effector

$\dot{x}_0$: Velocity vector of the collision avoidance point

Maciejewski and Klein suggest that there are some singular matrices, caused by the singular position of the collision avoidance point, that have to be taken into account when considering the control. This is due to the use of the pseudoinverse in the equation.

Zlajpah and Nemec (2002) discuss kinematics control algorithms as illustrated in Figure 6.2. They categorized the algorithms as velocity strategy and force strategy. They proposed a force strategy control which is a similar algorithm as that presented by Maciejewski and Klein, but also using artificial forces. This strategy avoided the use of the pseudoinverse problem.

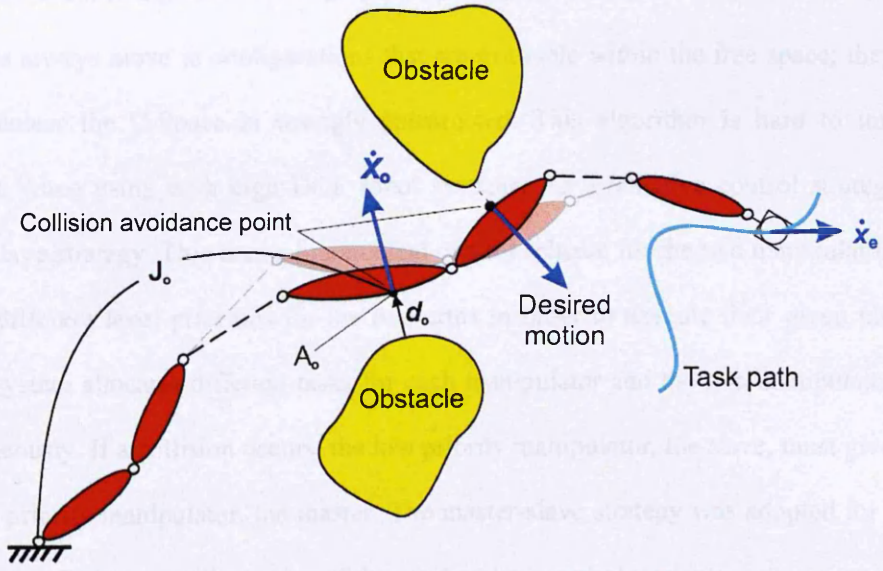


Figure 6.2 Manipulator motion in presence of obstacles (L. Zlajpah and B. Nemec 2002)

The main advantages of kinematics control algorithms are that they are easy to implement however, they are only suitable for applications which possess a redundant manipulator with a specified end-effector trajectory. In this research the kinematics control algorithm was adopted and tested for MARS-ND. In the following sections the control strategy for MARS-ND is illustrated; the implementation of the kinematics control algorithm with MARS-ND is discussed; and an examination is given of the collision avoidance strategy developed in order to find a solution to the problem of collision avoidance between the two Hydro-Lek arms while undertaking a given task. An introduction to collision detection using minimum distance calculation, kinematics control algorithms, collision avoidance strategy and its general solution are illustrated in Appendix B.

6.3 Control strategy for multi-arm robot system

The configuration space algorithm is a good choice to build the control system for a simple dual-arm robot system with two or three DOF for each arm. This is because it can easily convert the dual-arm collision-free control problem to a path-finding problem for a point robot in n-dimensional space, where n is the number of DOF of the dual-arm robot. The control strategy in C-Space algorithm for collision-free control is to control the two arms as a whole. The arms always move in configurations that are available within the free space; they do not collide unless the C-Space is wrongly constructed. This algorithm is hard to implement however, when using with high DOF robot systems. An alternative control strategy is the master-slave strategy. This uses a hierarchical control scheme for the two manipulators which assigns different level priorities for the two arms in order to execute their given tasks. The control system allocates different tasks for each manipulator and the two manipulators move simultaneously. If a collision occurs, the low priority manipulator, the slave, must give way to the high priority manipulator, the master. The master-slave strategy was adopted for MARS-ND. Figure 6.3 shows an illustration of the master-slave control strategy.

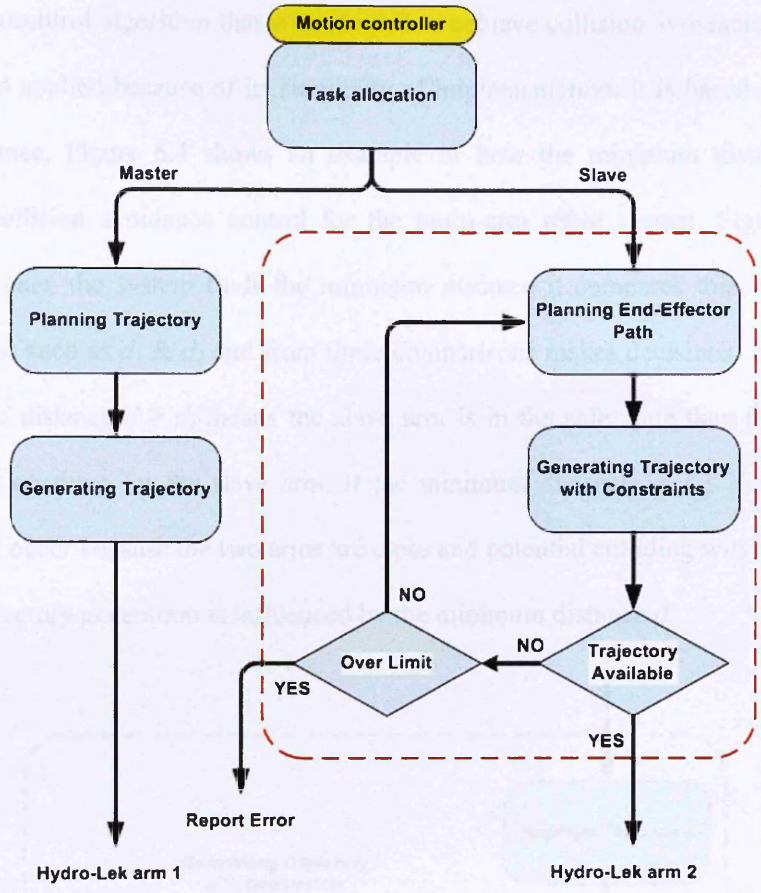


Figure 6.3 Master-slave control strategy

In the master-slave control strategy adopted for MARS-ND the master arm can move to a desired location while the slave arm plans its trajectory without interrupting the master arm. The control system can configure the slave arm's trajectory with constraints that will prevent it colliding with the master arm while in motion. If a desired trajectory is not available for the slave arm the control system will continue to check the trajectory availability, when it finds one then the slave arm is able to continue with its tasks. If no trajectory is found then the control system has to re-plan the end-effector trajectory until it reaches the looping limit. If the system reports an error, it means the slave arm failed to achieve its given task. The constraints on slave arm trajectory generation can be used for many purposes including a joint's angular velocity control and a joint's acceleration control. For the development of MARS-ND it was used for collision avoidance based on the control strategy described above in Figure 6.3.

The kinematics control algorithm that was adopted to achieve collision avoidance control for MARS-ND was applied because of its simplicity of implementation. It is based on finding a minimum distance. Figure 6.4 shows an example of how the minimum distance has an influence on collision avoidance control for the multi-arm robot system. Figure 6.4 also illustrates that once the system finds the minimum distance it compares this with specific critical distances such as d_1 & d_2 and from these comparisons makes decisions. For example, if the minimum distance $d \geq d_1$ means the slave arm is in the safe zone then the trajectory generation will continue for the slave arm. If the minimum distance is $d \leq d_2$ then system suspension will occur because the two arms are close and potential colliding will happen. If $d_1 > d \geq d_2$ the trajectory generation is influenced by the minimum distance d.

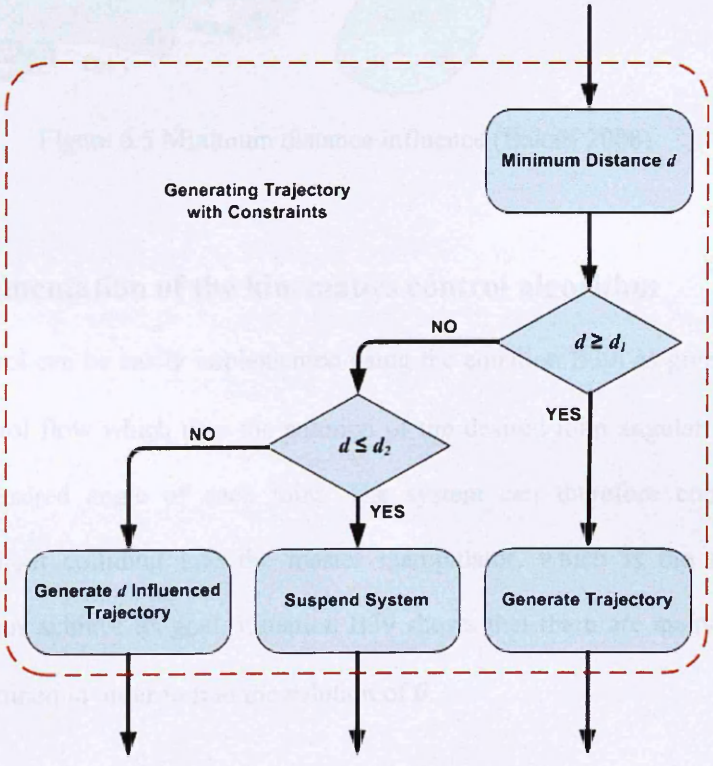


Figure 6.4 Minimum distance influence

The strategy of collision avoidance is to identify the point on the manipulator which is closest to an obstacle, denoted as collision avoidance point (A_0), and then assign to it a motion component that moves the point directly away from the obstacle as shown in Figure 6.5. In the case of a multi-arm system, the minimum distance (d_0) between each link of the two arms can

be calculated using virtual models which have the coordination details for all the links of the two arms. The direction of the collision avoidance point can also be obtained in the same way.

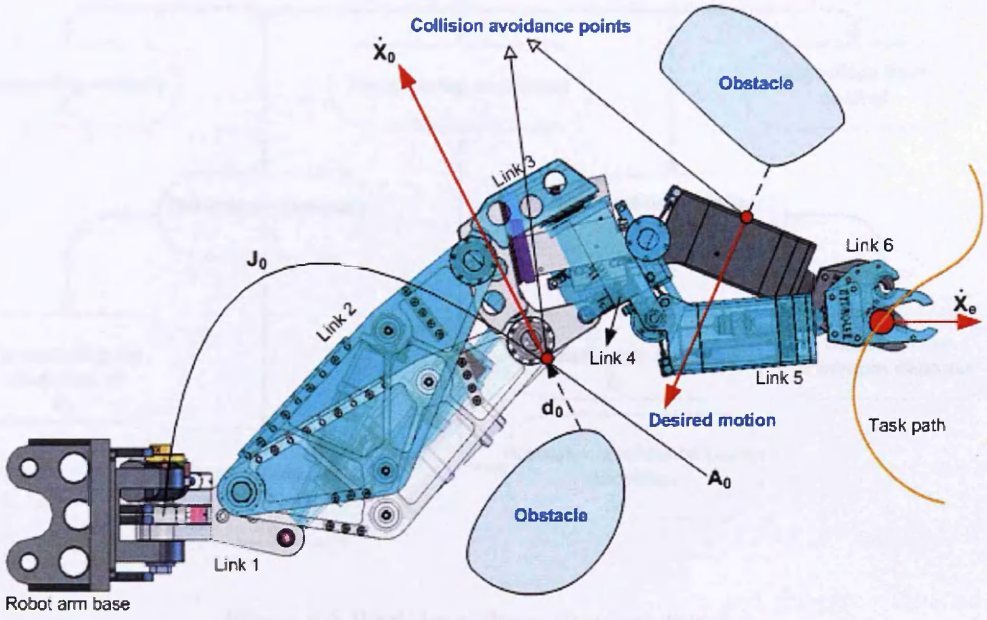


Figure 6.5 Minimum distance influence (Bakari 2008)

6.4 The implementation of the kinematics control algorithm

Kinematics control can be easily implemented using the equation B39, as given in Appendix B, within a control flow which uses the solution of the desired joint angular velocities $\dot{\theta}$ to determine the desired angle of each joint. The system can therefore control the slave manipulator without colliding into the master manipulator, which is the set of moving obstacles, and thus achieve its goal. Equation B39 shows that there are many variables that need to be determined in order to find the solution of $\dot{\theta}$.

The implementation of kinematics control can be divided into several subtasks, as illustrated in Figure 6.6, in this way the algorithm can be implemented step by step. The variables demonstrated in Figure 6.6 are the determination of end-effector velocity; avoidance velocity; Jacobians; virtual manipulators modelling; and algorithm flow control. These are explained in detail in Appendix B.

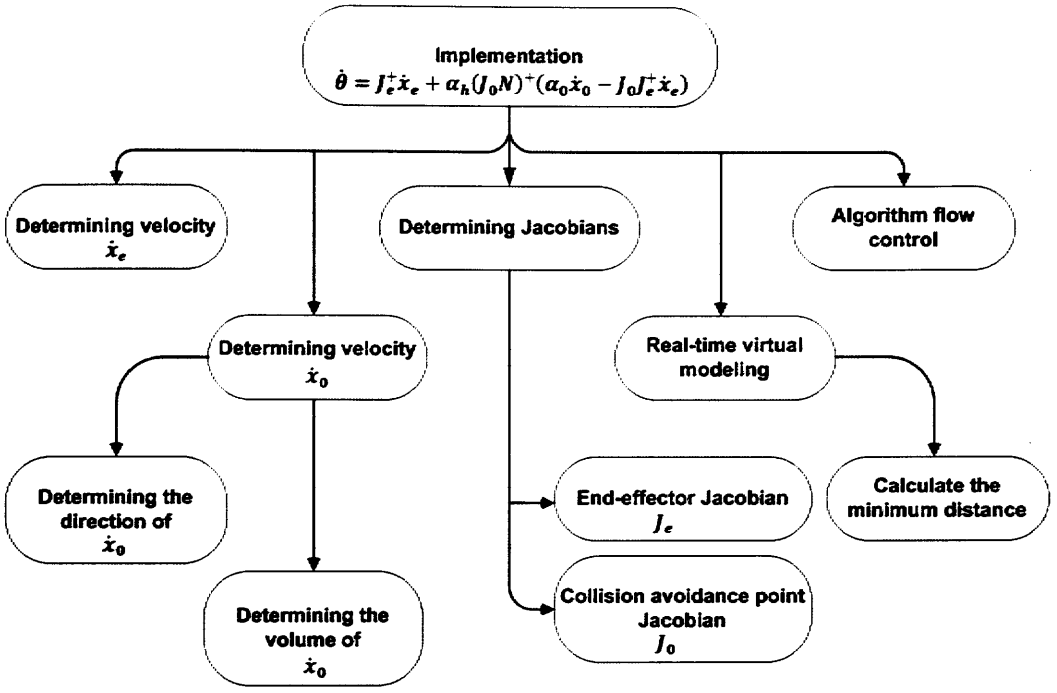


Figure 6.6 Work breakdown structure model

When a master-slave control system is used, the collision avoidance control problem for the dual-arm manipulators is transformed into the problem of collision avoidance control for a single-armed manipulator with moving obstacles. This is because the master arm is considered as a set of moving obstacles. When the condition of the end-effector trajectory is specified it is possible to implement the kinematics control algorithm to achieve collision avoidance control for the slave arm. After a detailed analysis of the kinematics control algorithm, the control algorithm can be implemented step by step as outlined in Figure 6.6.

6.5 Test and evaluation

When the master-slave control strategy was applied to MARS-ND it was found that the kinematics control algorithm would be a good choice for collision avoidance control if the condition of the slave robot manipulator end-effector trajectory is provided. The following sections explain the kinematics control algorithm for collision avoidance as applied for MARS-ND. Firstly a description of the test environment is outlined and a selection of tests

concerning the computation of parameters is provided; then the kinematics control algorithm for collision avoidance is presented, and an evaluation of implementation for MARS-ND is discussed.

6.5.1 Test environment

All the tests were carried out using a robot motion simulation package in MATLAB developed by the author and another researcher at Lancaster University (Bakari and Hu Yang 2007). Some of the functions were extended from the robotic toolbox for MATLAB as developed by Peter I. Corke (1996). The simulation package created the robot according the D-H parameters specified and displayed it in a 3D space. It simulated the 3D robot using a real-time animation function. Figure 6.7 shows an illustration of the robot motion simulation package. The robot manipulators models used for the tests were the dual-arm Hydro-Lek manipulators as shown in Figure 6.8 and a 6-DOF planar robot as shown in Figure 6.9. Since the dual-arm Hydro-Lek manipulators model is complicated and difficult to handle for testing the algorithm, some tests used the planar robot which made it easy to discover problems while testing.

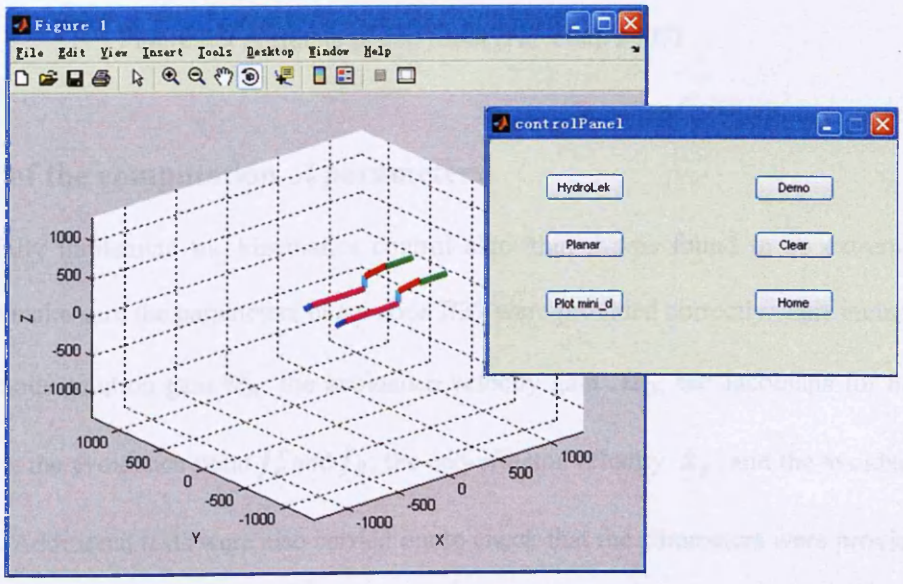


Figure 6.7 MATLAB simulation package

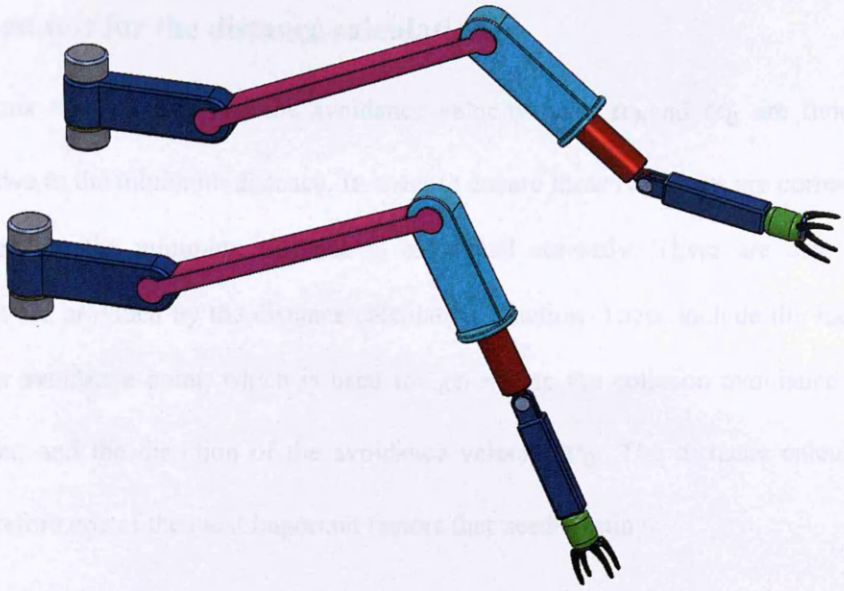


Figure 6.8 3D model of Hydro-Lek arms (Bakari 2008)

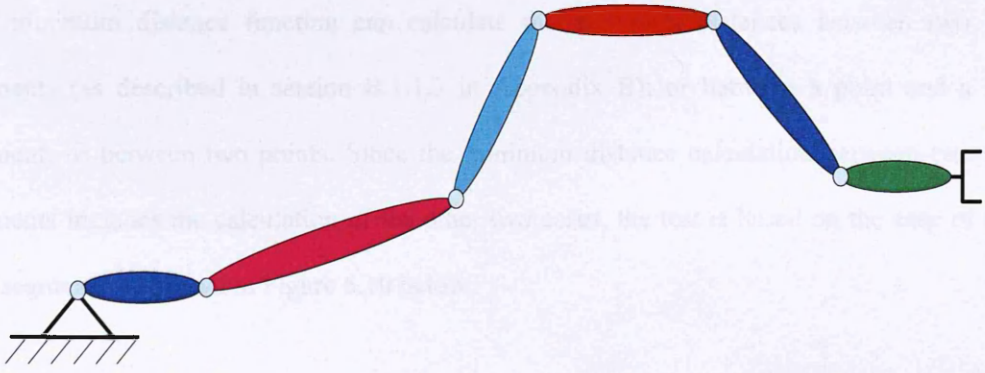


Figure 6.9 A simple planar robot (Hu Yang 2007)

6.5.2 Test of the computation of parameters

To successfully implement the kinematics control algorithm it was found to be extremely important to make sure the parameters in equation B39 were provided correctly. This included the homogenous solution gain $\boldsymbol{\alpha_h}$; the avoidance velocity gain $\boldsymbol{\alpha_0}$; the Jacobians for both end-effectors; the avoidance point $\boldsymbol{J_e^+}$ and $\boldsymbol{J_0}$; the end-effector velocity $\dot{\boldsymbol{x}}_{\boldsymbol{e}}$; and the avoidance velocity $\boldsymbol{v_0}$. Additional tests were also carried out to check that the parameters were provided correctly.

6.5.3 Function test for the distance calculation

The homogenous solution gain and the avoidance velocity gain, $\alpha_h$and α_0 are functions which are relative to the minimum distance. In order to ensure these two gains are correct it is necessary to ensure the minimum distance is calculated correctly. There are also other parameters that are provided by the distance calculation function. These include the location of the collision avoidance point, which is used for generating the collision avoidance point Jacobian matrix; and the direction of the avoidance velocity v_0. The distance calculation function is therefore one of the most important factors that needs testing.

6.5.4 Test setup

The minimum distance function can calculate the minimum distances between two line segments (as described in section B.1.1.3 in Appendix B); or between a point and a line segment; or between two points. Since the minimum distance calculation between two line segments includes the calculation of the other two cases, the test is based on the case of two line segments, as shown in Figure 6.10 below.

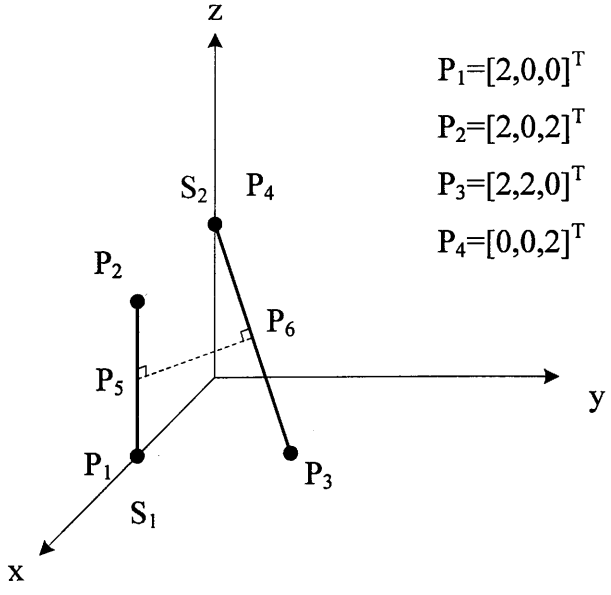


Figure 6.10 Distance calculation function test

Assuming that there are two line segments as shown in Figure 6.10:

Segment one: $\boldsymbol{S}_1 = segment(\boldsymbol{P}_1, \boldsymbol{P}_2), \boldsymbol{P}_1 = [2,0,0]^T, \boldsymbol{P}_2 = [2,0,2]^T$

Segment two: $\boldsymbol{S}_2 = segment(\boldsymbol{P}_3, \boldsymbol{P}_4), \boldsymbol{P}_3 = [2,2,0]^T, \boldsymbol{P}_4 = [0,0,2]^T$

The test is performed using MATLAB. Firstly MATLAB constructs the two segments with specified end-points; it then requests the distance function for the two segments; and finally it puts the return values into three variables and displays them to see the results. All of these activities are achieved using the following commands:

```
>> p1 = [2,0,0]';
>> p2 = [2,0,2]';
>> p3 = [2,2,0]';
>> p4 = [0,0,2]';
>> s1=segment(p1,p2);
>> s2=segment(p3,p4);
>> [d,p,u]=distance(s1,s2);
>>d,p,u

d = 1.4142
p = -1.0000          u =  -0.7071
    -1.0000               0.7071
     1.0000               0.0000
```

Where *d*, *p* and *u* are the three variables, *d* is the minimum distance; *p* is the point which has the minimum distance, represented in a position vector with respect to the first end-point of the 'slave' segment. In this case the 'slave' segment is $\mathbf{S}_2$; and the second end-point is $\mathbf{P}_3$; *u* is the unit vector of the direction of the avoidance velocity.

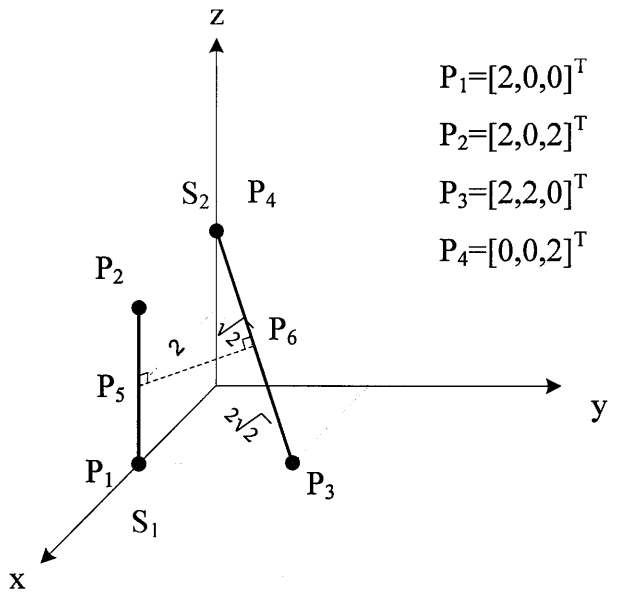


Figure 6.11 Distance calculation function test

In order to check the correctness of the outcomes of the distance calculation function, the three variables also need to be calculated manually. From Figure 6.11, it is easy to see that the minimum distance is the distance from $\mathbf{P_5}$ to $\mathbf{P_6}$; the avoidance velocity direction is from $\mathbf{P_5}$ points to $\mathbf{P_6}$; and the position vector of the minimum distance point, which is $\mathbf{P_6}$ in this case, is $\mathbf{P_6}$- $\mathbf{P_5}$. Therefore the three variables can be calculated as follows:

$$d = \sqrt{2^2 - (\sqrt{2})^2} = \sqrt{2} \approx 1.4142 \tag{6.2}$$

$$p = \boldsymbol{P}_6 - \boldsymbol{P}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \tag{6.3}$$

$$u = \frac{\boldsymbol{P}_6 - \boldsymbol{P}_5}{\|\boldsymbol{P}_6 - \boldsymbol{P}_5\|} \approx \begin{bmatrix} -0.7071 \\ 0.7071 \\ 0 \end{bmatrix} \tag{6.4}$$

The testing process allows the performance of the distance calculation function to be read easily from table 6.1. It shows that the function gives the values exactly as expected.

Variables	Function Outcome	Expected Values	Performance Check
D	1.4142	1.4142	OK
P	$\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$	OK
U	$\begin{bmatrix} -0.7071 \\ 0.7071 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.7071 \\ 0.7071 \\ 0 \end{bmatrix}$	OK

Table 6.1 Test results from the distance calculation function

This test has thus numerically proved the distance calculation function can provide the correct calculations for two line segments. Moreover, the distance calculation function also gives excellent performance for complex situations.

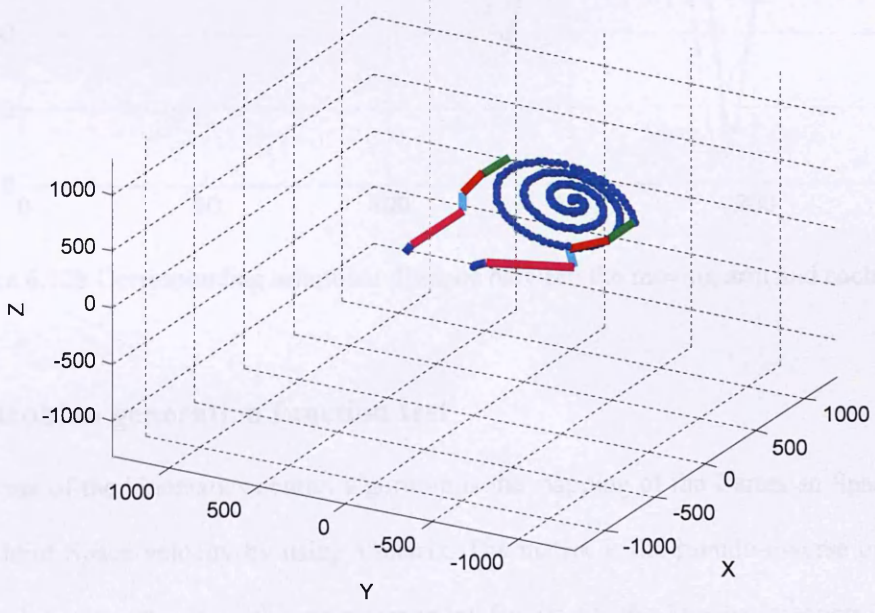


Figure 6.12a Distance calculation test for a multi-obstacle environment

Figure 6.12a shows an example of a distance calculation for a multi-obstacle environment. In this example a manipulator traces a specified end-effector trajectory and another manipulator is considered as seven obstacles. This is because the manipulator consists of seven links and

each link is considered as an obstacle. The distance calculation function tends to calculate the minimum distance from the move manipulator to each obstacle and gives very good results. 'It can be used for the kinematics control algorithm to generate α_h and α_0; and to provide the location of the avoidance point and the direction of the avoidance velocity. Figure 6.12b illustrates the corresponding minimum distance between the moving arm and each obstacle.

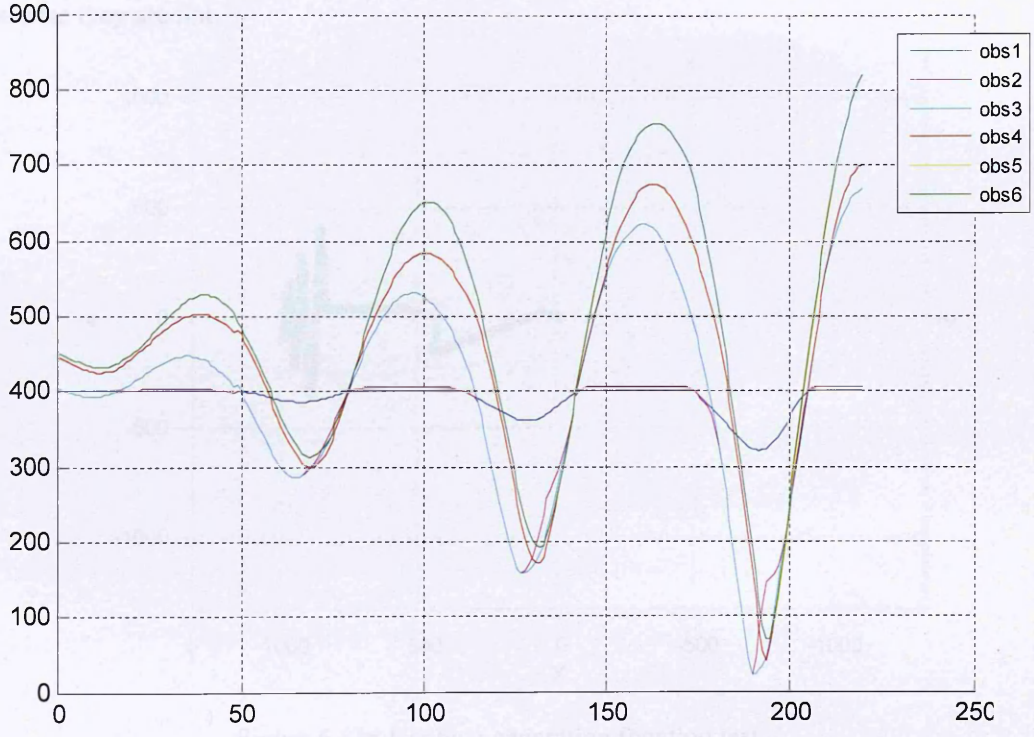


Figure 6.12b Corresponding minimum distance between the moving arm and each obstacle

6.6 Jacobian generation function test

A concept of the kinematics control algorithm is the mapping of the Cartesian Space velocity to the Joint Space velocity by using a matrix. The matrix is the pseudo-inverse of Jacobian, the Jacobians are therefore the most important factors for the kinematics control. The test presented here checks the correctness of the Jacobian generation and tests the correctness of the end-effector velocity determination. Since the Jacobians and the end-effector velocity can also be used for calculating the inverse kinematics for a robot, where only the inverse Jacobian and end-effector velocity are used, the inverse kinematics calculation can also be used for testing the Jacobians and end-effector velocity.

The test method is as follows:

- Specify the end-effector trajectory
- Force the robot to trace the trajectory using the inverse kinematics

If the robot can trace the trajectory then both the Jacobian and end-effector are correct, otherwise they are not.

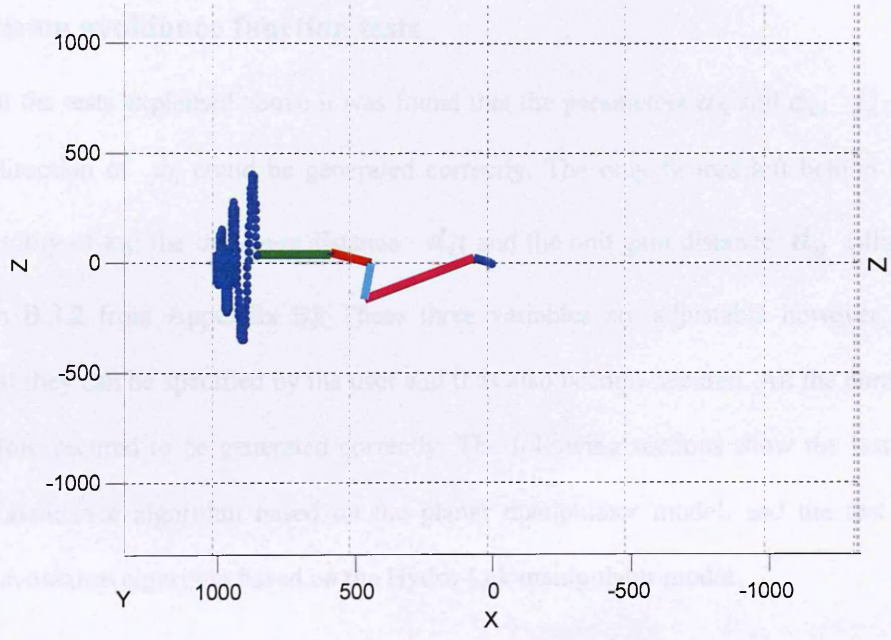


Figure 6.13a Jacobian generation function test

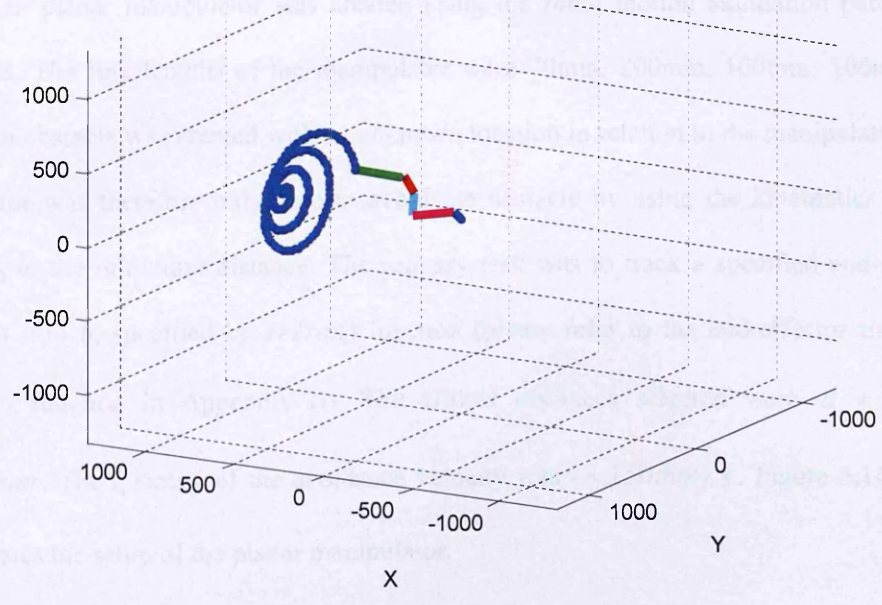


Figure 6.13b Jacobian generation function test

Figures 6.13a and 6.13b show a test for the end-effector Jacobian and end-effector velocity. It can be seen that the end-effector can move to trace the trajectory. The collision avoidance point Jacobian generation function can also be tested in the same way and gives the same result. The Jacobians generation function and the end-effector velocity calculation function therefore appear to be working correctly.

6.7 Collision avoidance function tests

During all the tests explained above it was found that the parameters α_h and α_0, $J_e^+, J_0, \dot{x}_e$ and the direction of v_0 could be generated correctly. The only factors left behind are the scalar quantity of v_0; the influence distance d_i; and the unit gain distance d_u (illustrated in section B.3.2 from Appendix B). These three variables are adjustable however, which means that they can be specified by the user and thus also become secured. All the parameters are therefore secured to be generated correctly. The following sections show the test of the collision avoidance algorithm based on the planar manipulator model, and the test of the collision avoidance algorithm based on the Hydro-Lek manipulator model.

6.7.1 Test on planar manipulator

A six DOF planar manipulator was created using the robot motion simulation package in MATLAB. The link lengths of the manipulator were 70mm, 200mm, 100mm, 100mm and 80mm. An obstacle was created with an unknown location in relation to the manipulator. The manipulator was therefore only able to avoid the obstacle by using the kinematics control according to the minimum distance. The primary task was to track a specified end-effector path from *a* to *b*, specified by *eeTraj*() function (please refer to the end-effector trajectory generation function in Appendix B). The critical distances selected were $d_i = 55mm$, $d_u = 40mm$. The quantity of the avoidance velocity was $v = 150mm/s$. Figure 6.14 below demonstrates the setup of the planar manipulator.

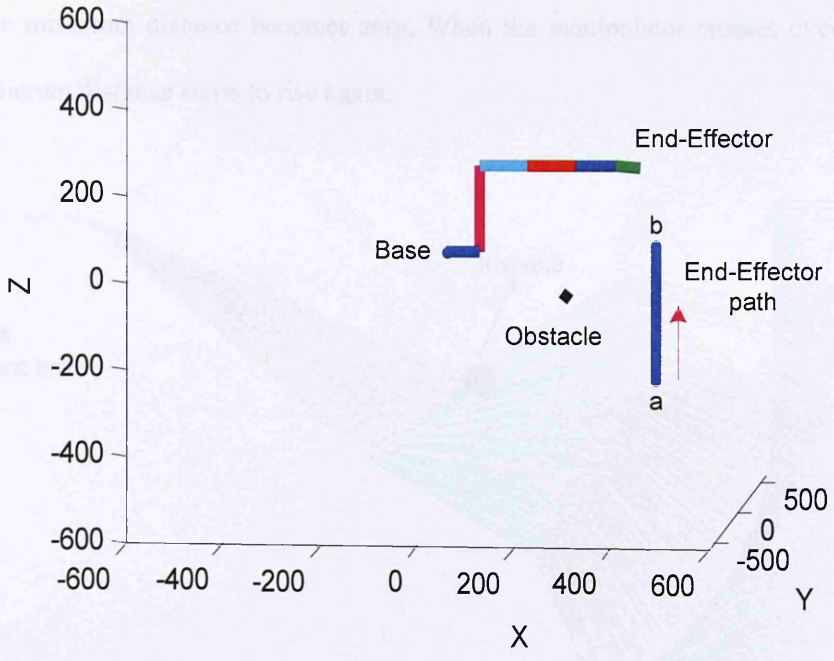


Figure 6.14 Planar manipulator

To find out the effectiveness of the collision avoidance control when implemented with the Planar and HydroLek manipulators, the test was processed using the three steps given below:

1. The robot was forced to track the path without considering the collision avoidance by using the equation $\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^{+}\dot{\boldsymbol{x}}_e$ instead of the kinematics control equation B40 (given in Appendix B)
2. The robot traced the path with the collision avoidance control by using the equation B40
3. The minimum distance was plotted against the time for both tests, with and without the collision avoidance control.

From the test results, given in Figure 6.15a, it can be seen that the manipulator tracked the specified path and ignored the existence of the obstacle. This suggests that the manipulator will collide with the obstacle. Figure 6.15b shows that when the manipulator moves toward to the obstacle the minimum distance decreases until the manipulator collides with the obstacle,

then the minimum distance becomes zero. When the manipulator crosses over the obstacle, the minimum distance starts to rise again.

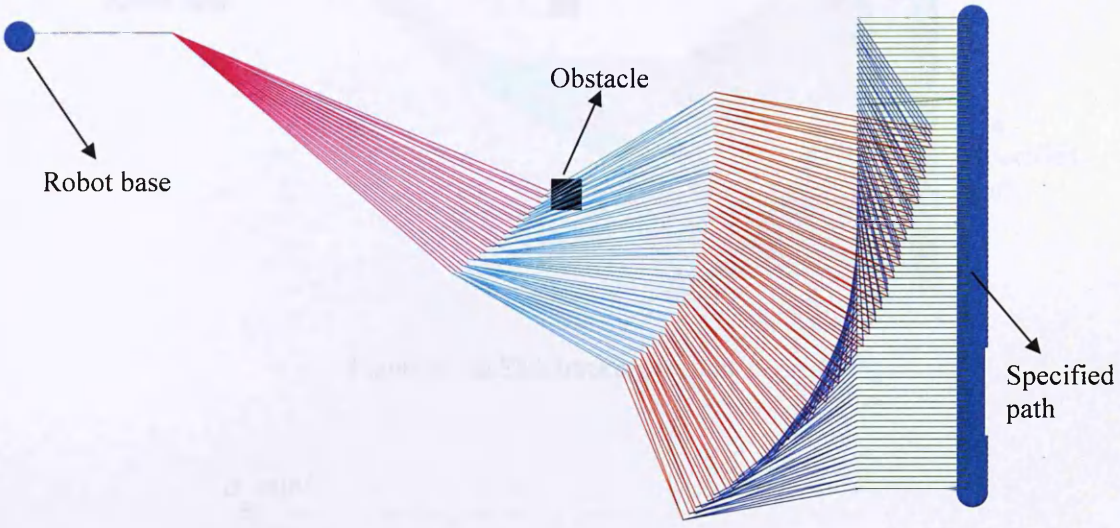


Figure 6.15a Path tracking motion

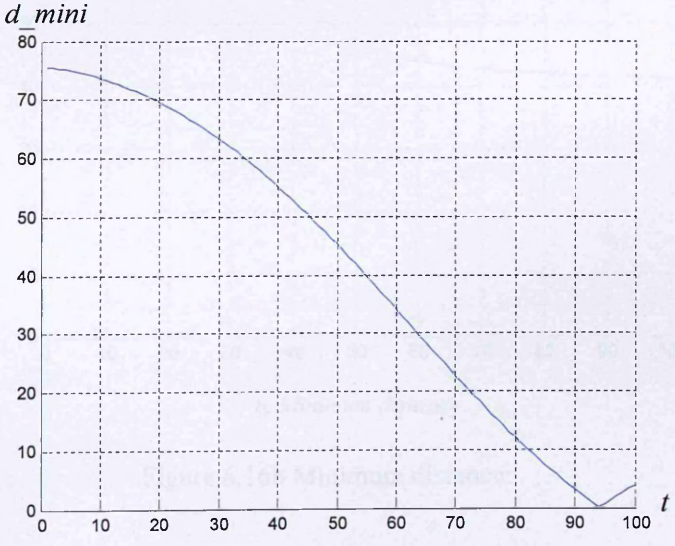


Figure 6.15b Minimum distance

Figure 6.16a shows that the manipulator did not collide with the obstacle while it was tracking the same specified path. Figure 6.16b shows that the minimum distance begins at the same point as shown in Figure 6.15b but decreases smoothly approaching 40mm.

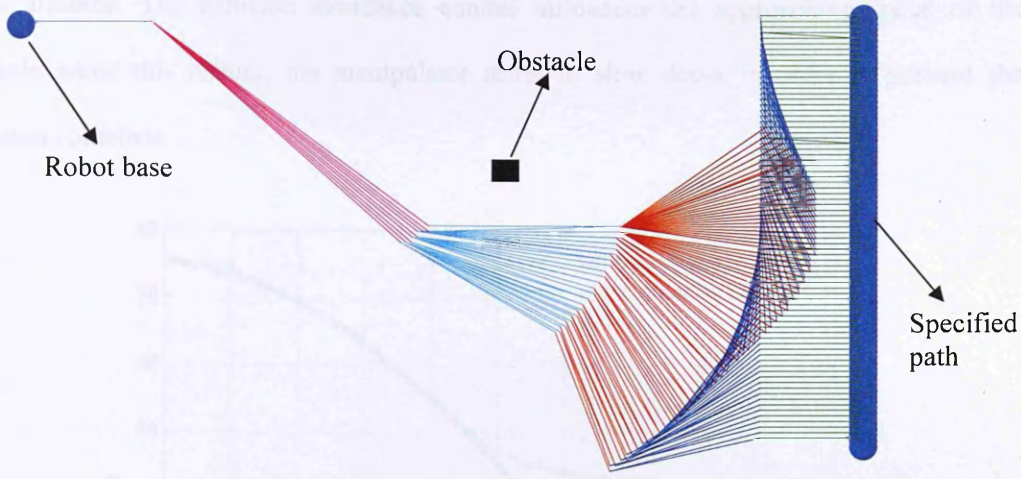
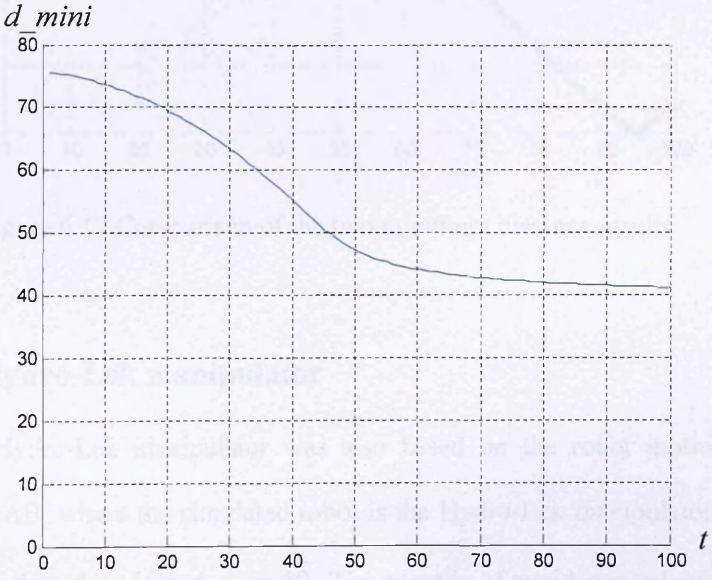


Figure 6.16a Path tracking motion



b. Minimum distance

Figure 6.16b Minimum distance

The green line shown in Figure 6.17 represents the minimum distance effect without using the collision avoidance control algorithm. The blue line in Figure 6.17 represents the minimum distance effect when the collision avoidance control is applied. The differences highlighted by this graph are that the green line decreases to zero and then rises because the manipulator crosses the obstacle. The blue line also decreases to the same extent until it reaches 55 where it is set as the influence distance d_i; it then smoothly approaches 40 where it is set as the unit

gains distance. The collision avoidance control influences the approaching speed of the obstacle when this occurs; the manipulator starts to slow down in order to prevent the potential collisions.

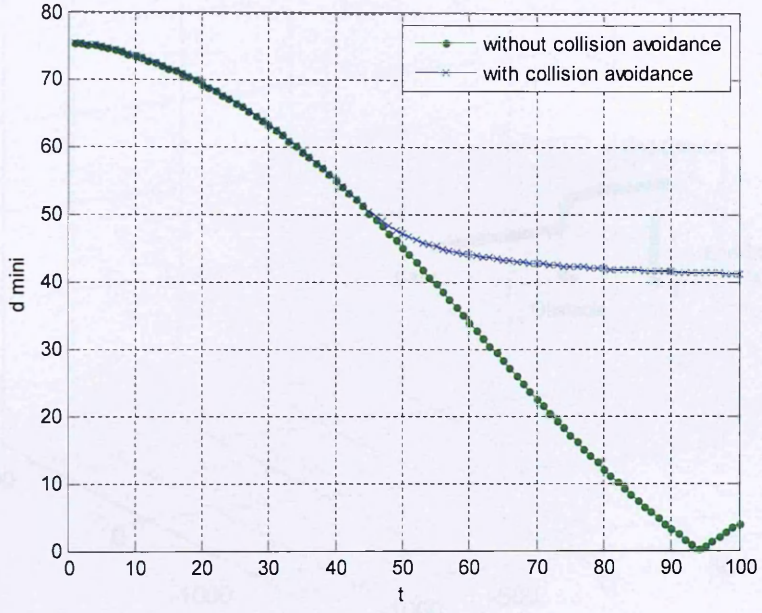


Figure 6.17 Comparison of the two minimum distance results

6.7.2 Test on Hydro-Lek manipulator

The test on the Hydro-Lek manipulator was also based on the robot motion simulation package in MATLAB, where the simulated robot is the Hydro-Lek manipulator. The critical distances are set so that $d_i = 55$ and $d_u = 40$. The quantity of avoidance velocity is $v = 150$. Figure 6.18 shows the initial setup.

The test method for the collision avoidance control for the Hydro-Lek manipulator was the same as that used for the planar manipulator. The test results, given in Figure 6.19, show that potential collision is prevented, but the manipulator does not complete the tracking task. From the perspective of the minimum distance it also shows that the manipulator stops at the point when it reaches the minimum distance 50 as shown in Figure 6.20. The collision avoidance control returned an error message, "Solution wouldn't converge". This message is returned

when the control algorithm cannot find a solution to move the end-effector to the target position without colliding into an obstacle in thousand looping cycles.

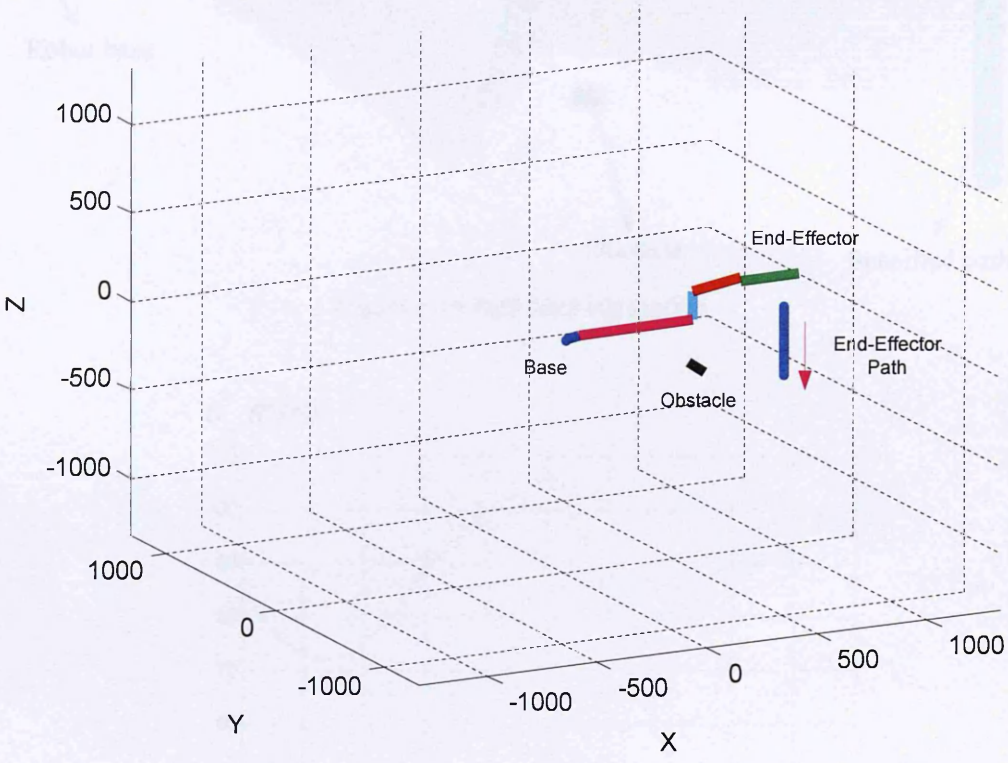


Figure 6.18 Hydro-Lek manipulator test setup

Through the tests it was confirmed that the parameters needed for the kinematics control were secure and could be used for the algorithm. It was also found that the algorithm worked for the 6 DOF planar manipulator, and gave a good performance; it also prevented a potential collision between the robot and the obstacle. It can therefore be concluded that the implementation of kinematics control for collision avoidance would seem to work well, particularly on high redundancy planar manipulators, although some problems were evident. For example in the test on the Hydro-Lek manipulator the potential collision was prevented but the primary task, which was to trace a specified end-effector trajectory, was also suspended. The manipulator still had enough redundancy to reconfigure the joint to continue tracking the path without collisions. This is illustrated in Figure 6.21.

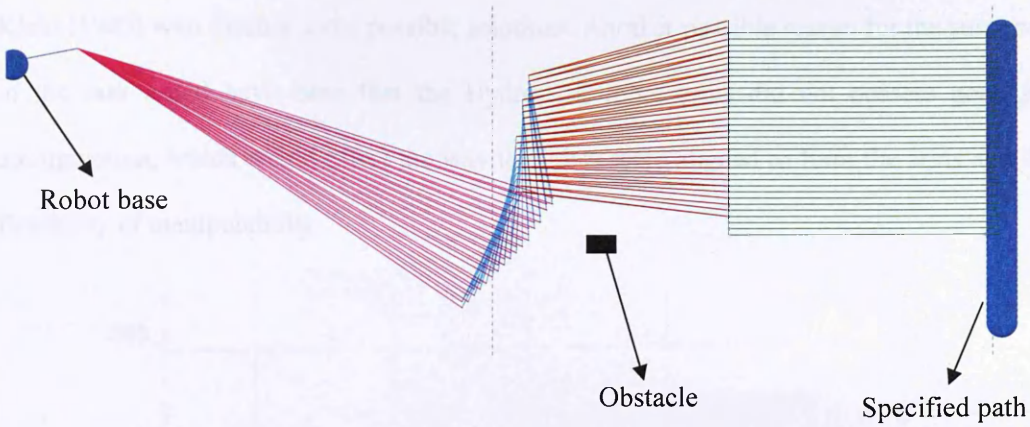


Figure 6.19 Path tracking motion

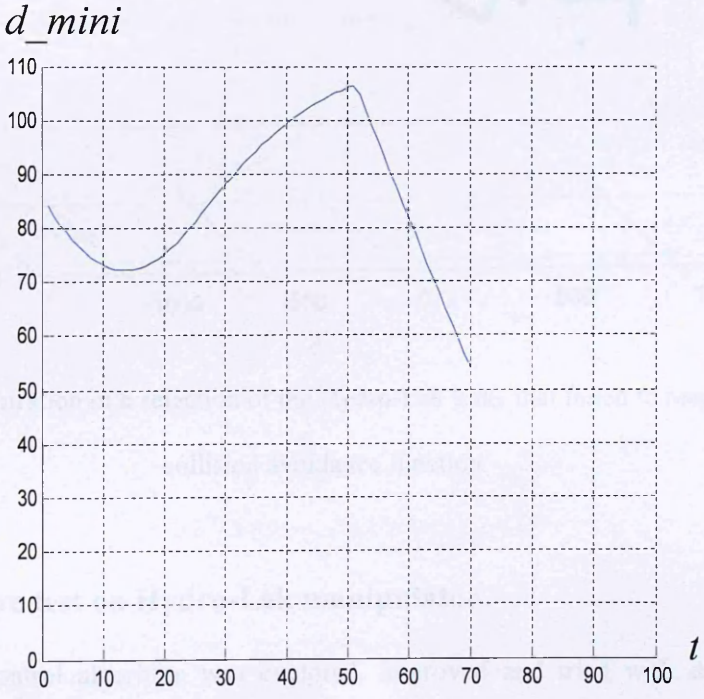


Figure 6.20 Minimum distance calculation

From a programming perspective one of the reasons the task was suspended was because when the manipulator approached the obstacle, the matrix $(J_0 N)^+$ from the equation C40, (which is a component of the kinematics control algorithm), yielded unacceptably large numbers. This meant that the kinematics control loop could not produce the desired joint velocity to satisfy the specified tolerance. The program therefore continued looping until it reached the looping limit. This problem is also identified by A. A. Maciejewski and C. A.

Klein (1985) who discuss some possible solutions. Another possible reason for the suspension of the task could have been that the Hydro-Lek robot arms did not possess good joint configuration, which would affect the way the joints were aligned to form the arms and their flexibility of manipulability.

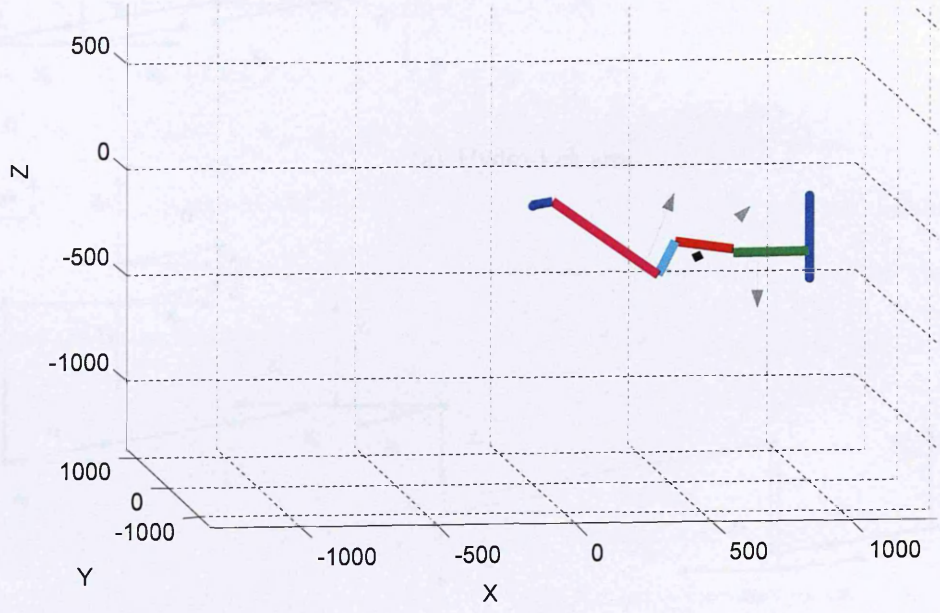
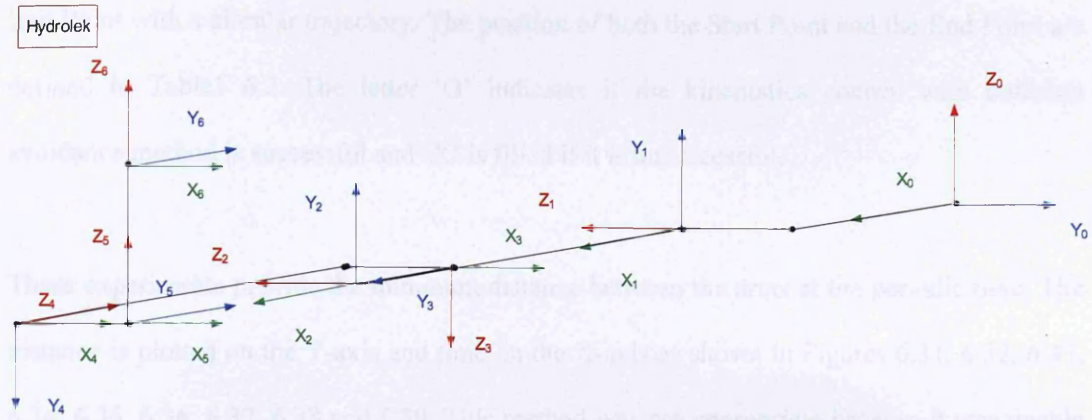


Figure 6.21 Illustration of a selection of the Hydro-Lek links that failed to respond to the collision avoidance function

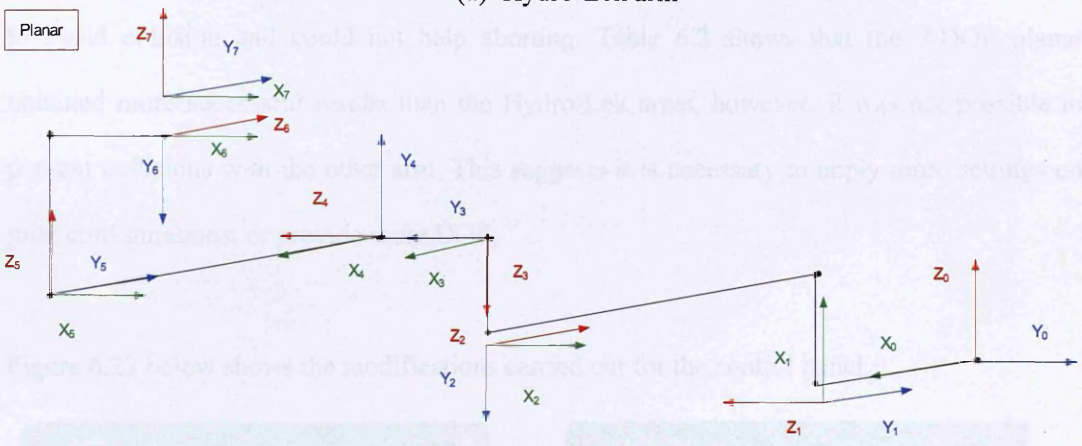
6.7.3 Alternative test on Hydro-Lek manipulator

The kinematics control algorithm was explored, improved and tried with different robot configurations during the author's research collaboration work with the Sugano team at Waseda University, Tokyo, Japan. The aim of this was to test the effectiveness of this algorithm in different contexts and to explore the possibility the adoption of this algorithm by the Sugano team for the TWENDY-ONE robot collision avoidance problem.

In order to validate the kinematics control algorithm for collision avoidance, experiments were undertaken to compare the 6-DOF configuration Hydro-Lek arms and the 7-DOF redundant two planar manipulators. The joint configuration of the 7-DOF planar manipulators is illustrated in Figure 6.22b as compared with the Hydro-Lek arms as shown in Figure 6.22a.



(a) Hydro-Lek arm



(b) 7 DOF planar arm

Figure 6.22 Robot arms configuration

	StartPoint			EndPoint			Success or Not	
	X	Y	Z	X	Y	Z	Hydro-Lek	7DOF Plannar
HorizontalBase	750	0	0	750	-400	0	○	○
HorizontalX+100	850			850			○	○
HorizontalX-100	650			650			×	○
HorizontalZ+100	750		100	750		100	○	○
HorizontalZ-100			-100			-100	○	○
VerticalBase	750	-100	200	750	-100	-200	×	○
VerticalX+100	850			850			×	×
VerticalX-100	650			650			○	○
VerticalY+100	750	100		750	100		×	○
VerticalY-100		200			200		×	○

Table 6.2 Comparison between Hydro-Lek arm and 7 DOF Planar arm

These multi-arms are settled as the Master-Slave manipulators, thus, in this experiment, one arm is in a static position and posture while the other arm moves from the Start Point to the

End Point with a circular trajectory. The position of both the Start Point and the End Point are defined in Table1 6.2. The letter 'O' indicates if the kinematics control with collision avoidance method is successful and 'X' is filled if it is unsuccessful.

These experiments provide the minimum distance between the arms at the periodic time. The distance is plotted on the Y-axis and time on the X-axis as shown in Figures 6.31, 6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38 and 6.39. This method was not appropriate because it was unable to avoid collision and could not help aborting. Table 6.2 shows that the 7-DOF planar obtained more successful results than the Hydro-Lek arms, however, it was not possible to prevent collisions with the other arm. This suggests it is necessary to apply more settings on joint configurations, or provide more DOF.

Figure 6.23 below shows the modifications carried out for the control panel.

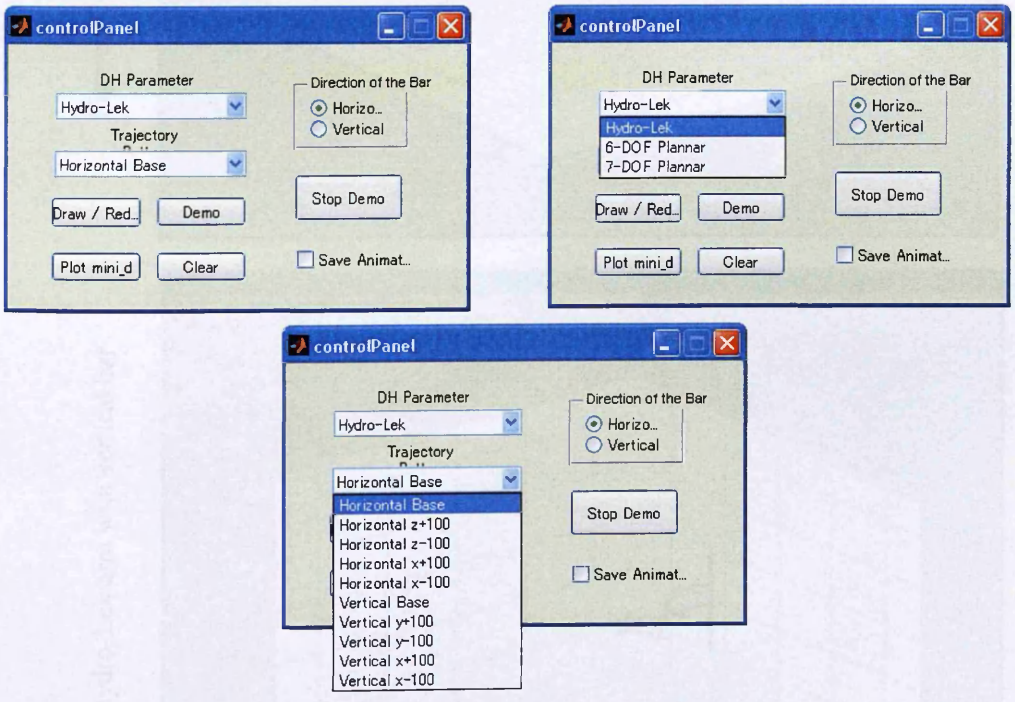


Figure 6.23 Control panel for the kinematic control algorithm in MatLab

The new features added to the control panel shown above are:

(a) 6 DOF planar robot arm configuration with new D-H table

(b) 7 DOF planar robot arm configuration with new D-H table

(c) Horizontal or vertical bars

(d) Distance adjustment between the bar (task) and robot arm end-effectors

(e) Drawing and redrawing of the selected robot arm

(f) Saving the animation of each trajectory generated for the selected arm

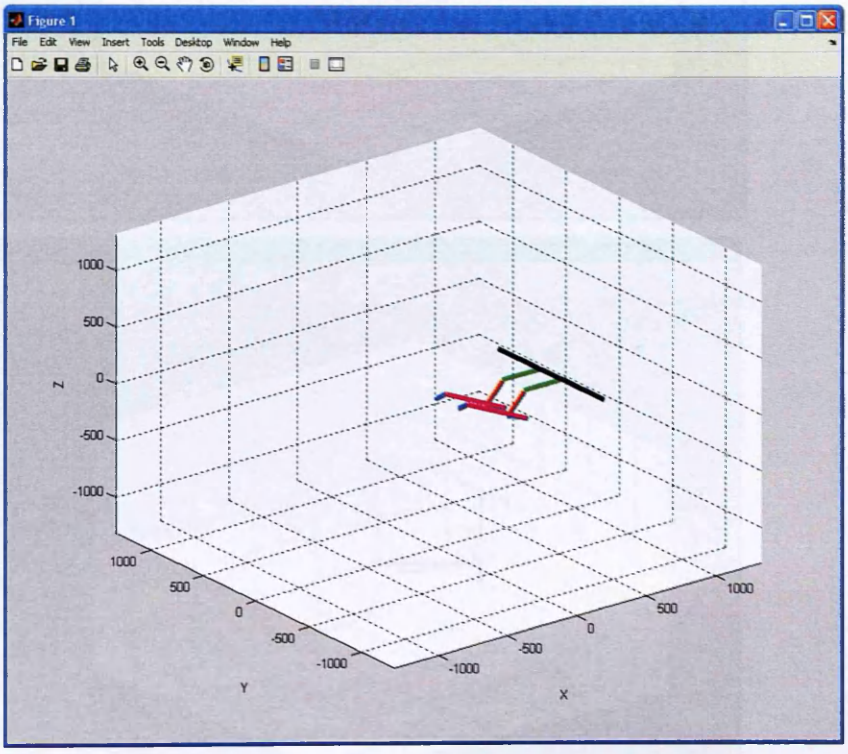


Figure 6.24 Hydro-Lek arm with horizontal bar

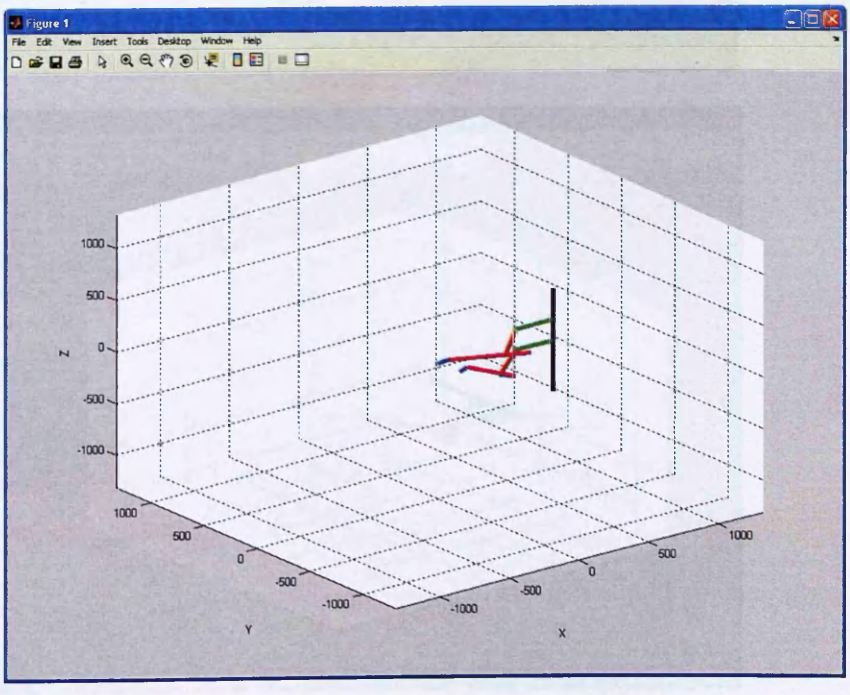


Figure 6.25 Hydro-Lek arm with vertical bar

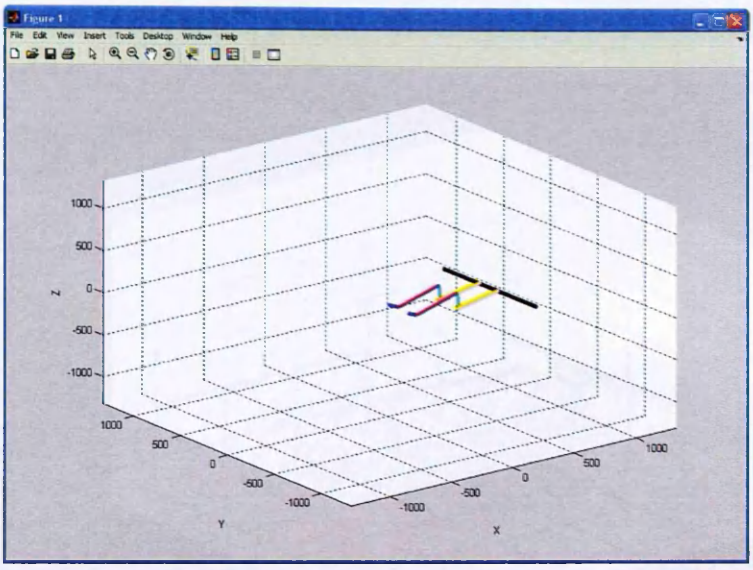


Figure 6.26 7-DOF planar arm with horizontal bar

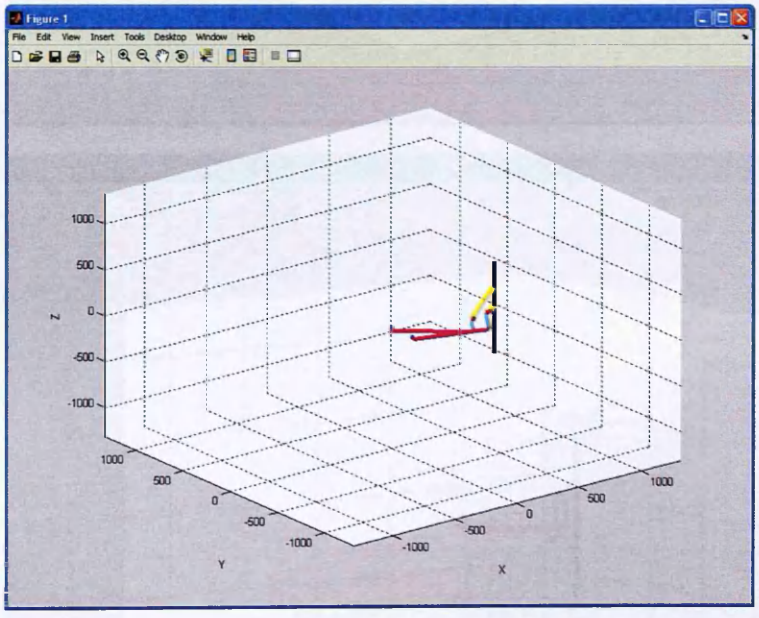


Figure 6.27 7-DOF planar arm with vertical bar

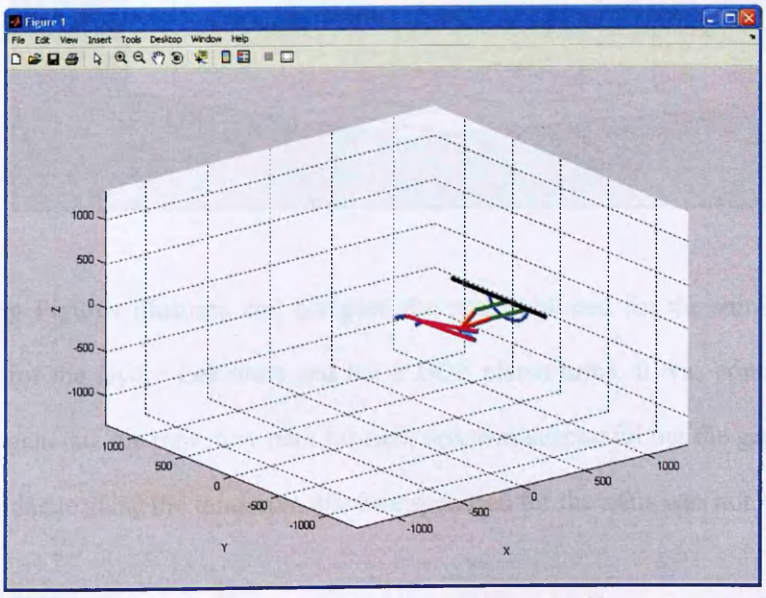


Figure 6.28 7 Trajectory generation for Hydro-Lek arm

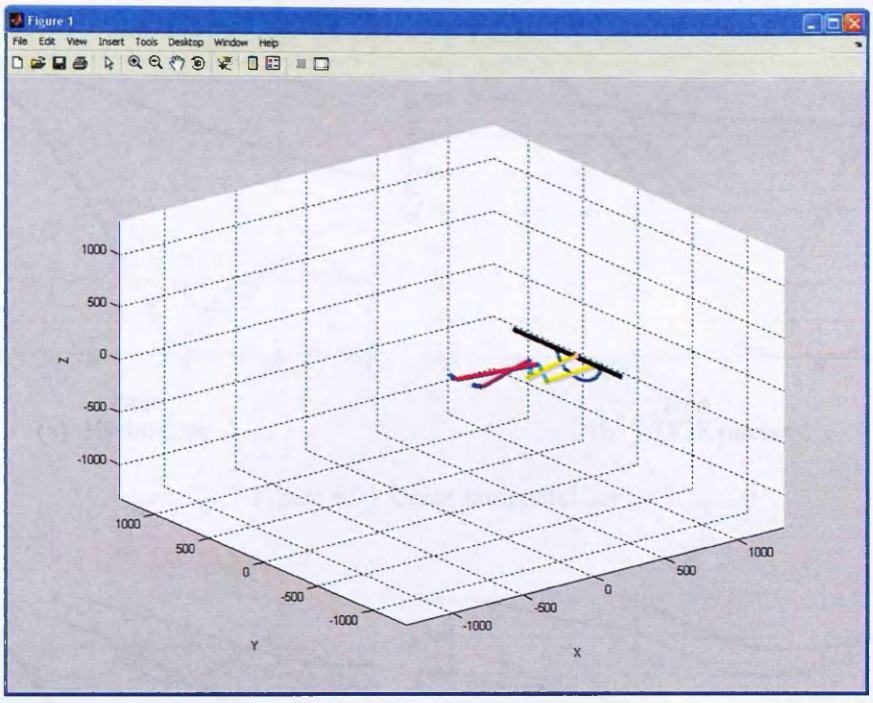


Figure 6.29 7 Trajectory generation for 7-DOF planar arm

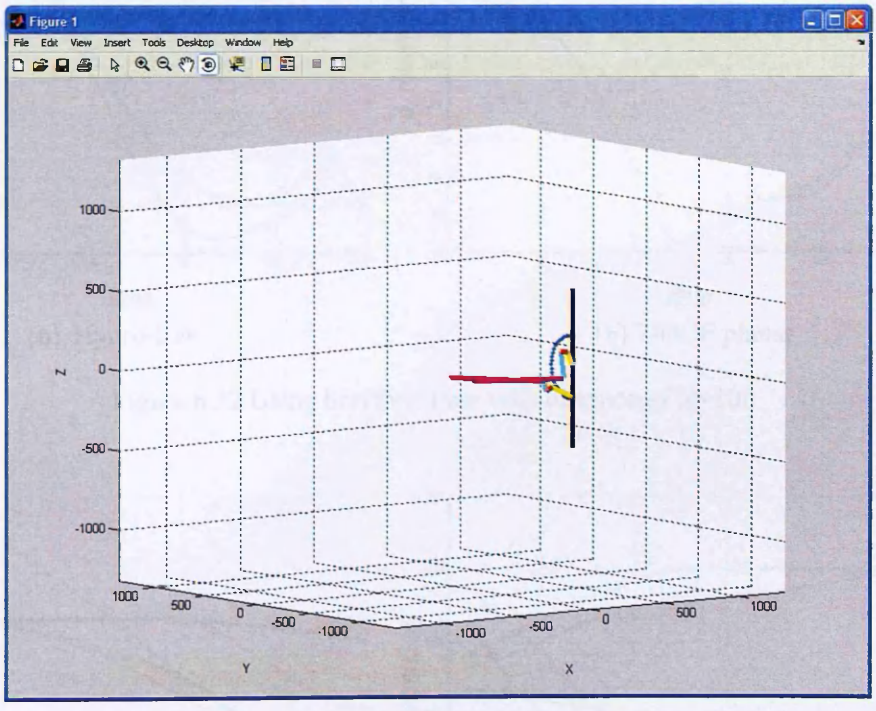
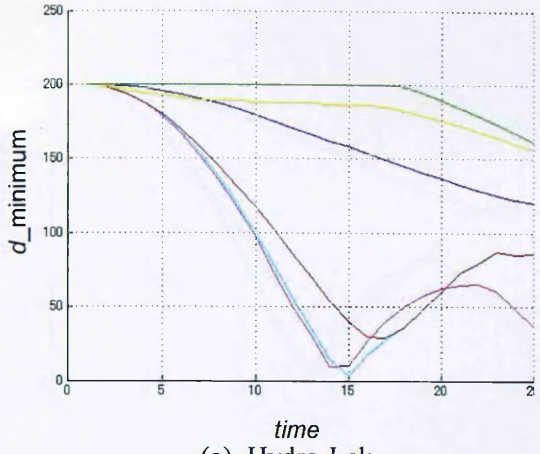
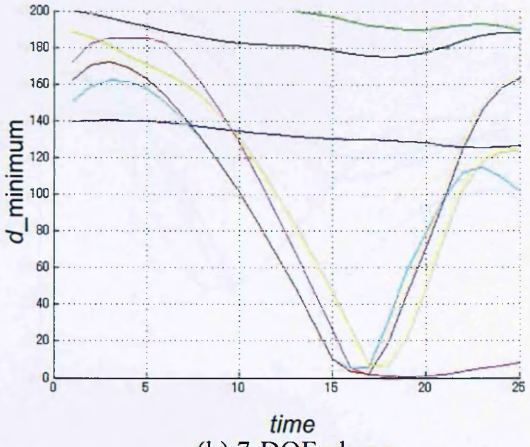


Figure 6.30 7 Trajectory generation for 7-DOF planar arm with vertical bar

The following Figures illustrate and compare the results plotted for the minimum distance against time for the Hydro-Lek arms and the 7 DOF planar arms. It was concluded that the code used to generate the trajectory path for each task was successful but the generation of the collision avoidance using the minimum distance specified for the arms was not responding.

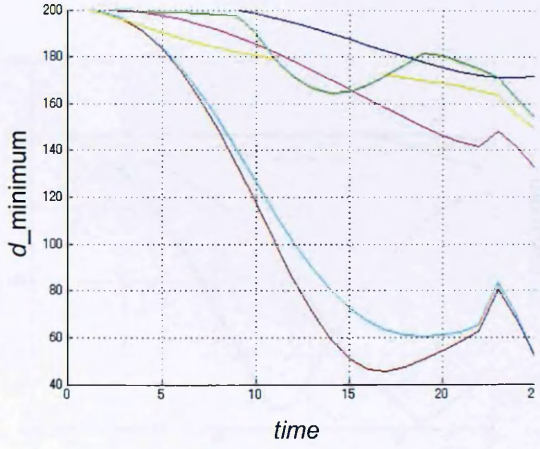


(a) Hydro-Lek

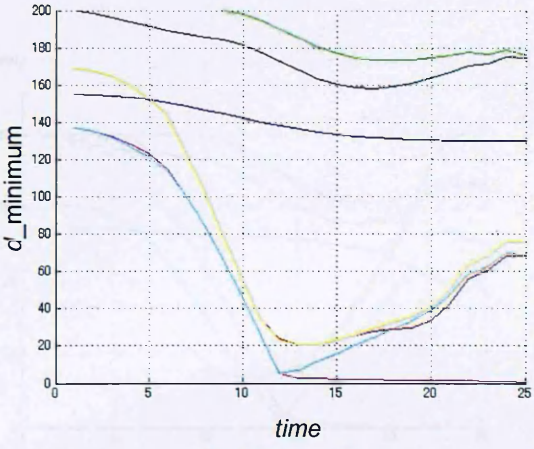


(b) 7-DOF planar

Figure 6.31 Using horizontal bar

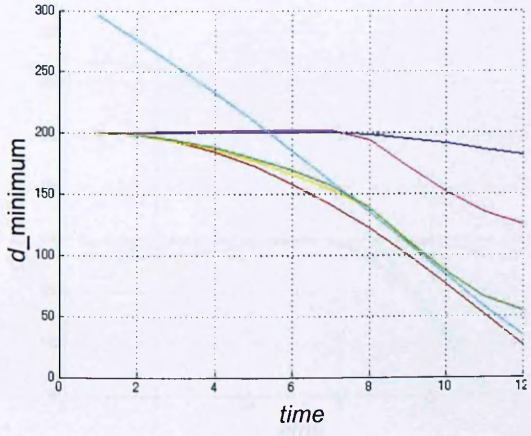


(b) Hydro-Lek

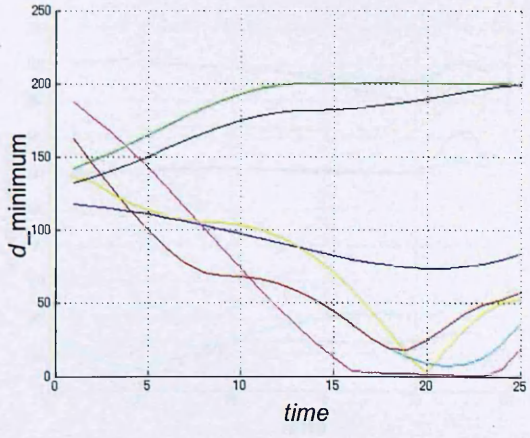


(b) 7-DOF planar

Figure 6.32 Using horizontal bar with distance of X+100

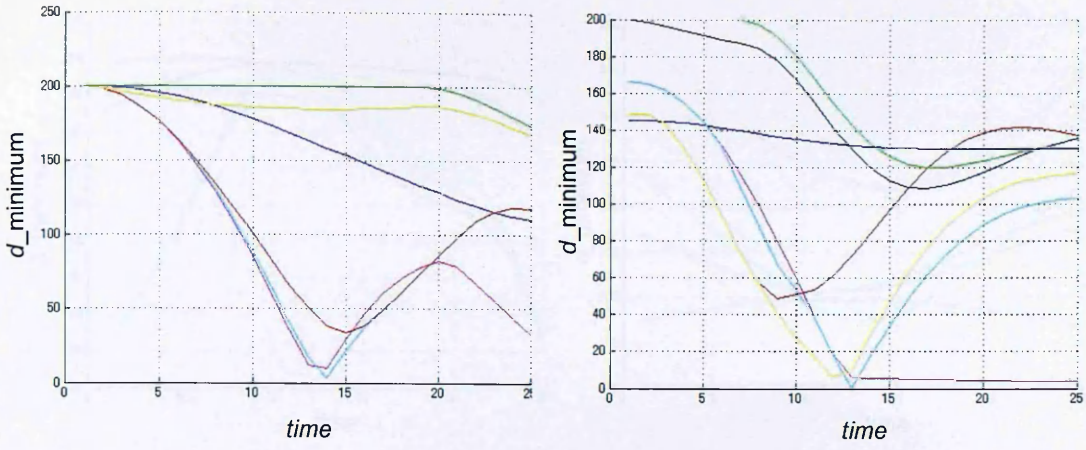


(a) Hydro-Lek



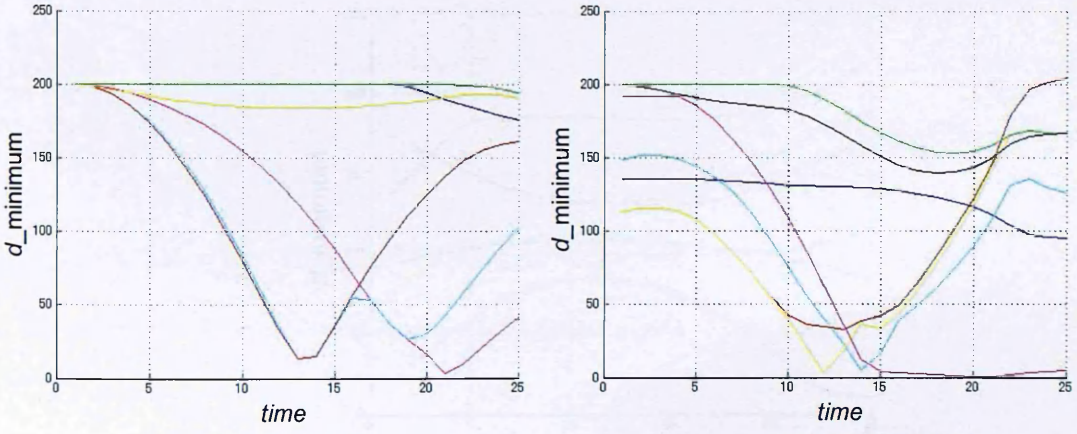
(b) 7-DOF planar

Figure 6.33 Using horizontal bar with distance of X-100



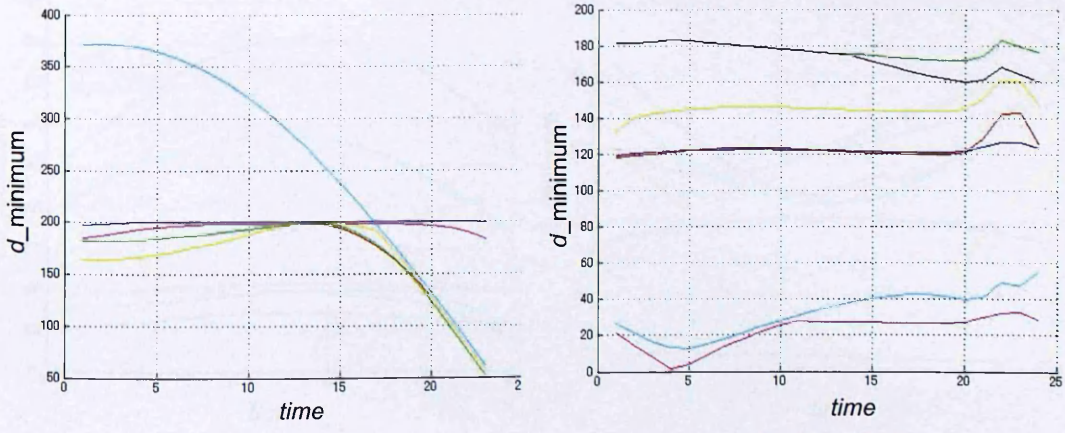
(a) Hydro-Lek (b) 7-DOF planar

Figure 6.34 Using horizontal bar with distance of Z+100



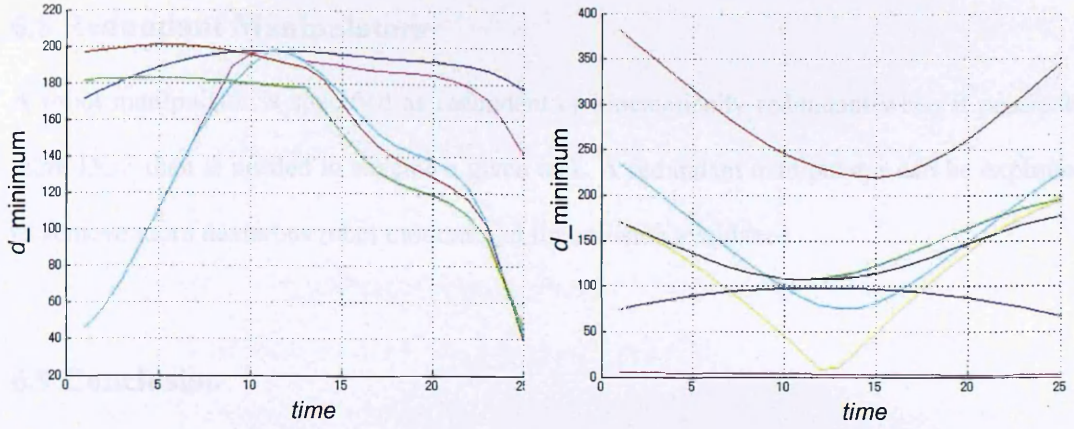
(a) Hydro-Lek (b) 7-DOF Planar

Figure 6.35 Using horizontal bar with distance of Z-100



(a) Hydro-Lek (b) 7-DOF Planar

Figure 6.36 Using vertical bar with distance of X+100



(a) Hydro-Lek (b) 7-DOF Planar

Figure 6.37 Using vertical bar with distance of X-100

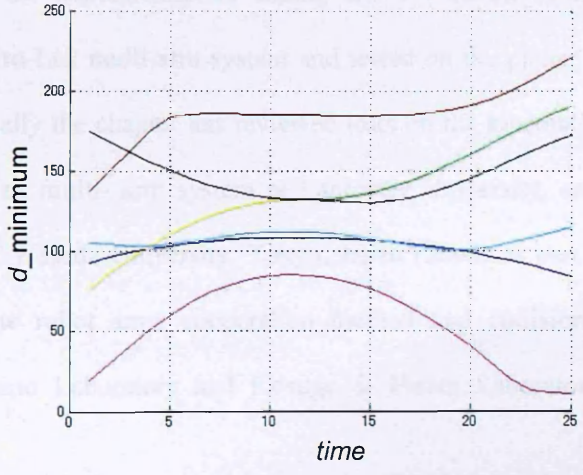
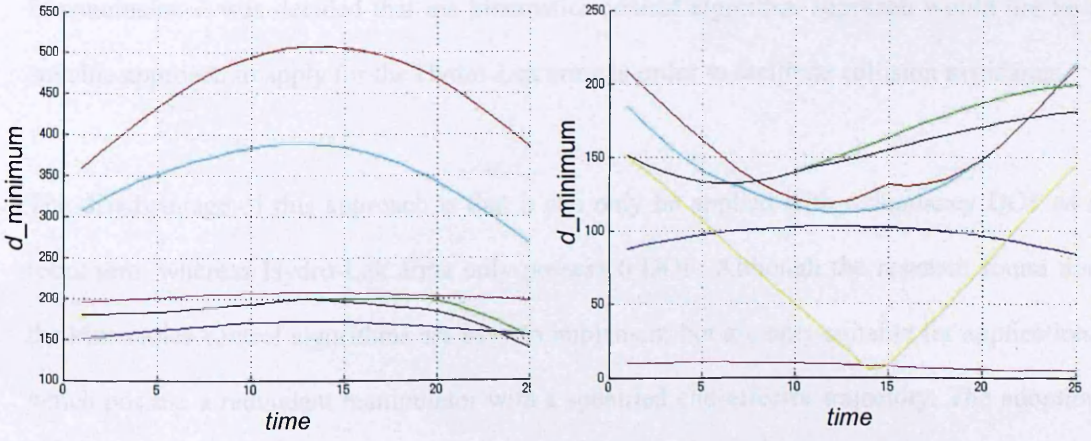


Figure 6.38 Using vertical bar with 7-DOF planar arm



(a) Vertical Y+100 (b) Vertical Y-100

Figure 6.39 Using vertical bar with 7-DOF planar arm with (Y+100 and Y-100)

6.8 Redundant Manipulators

A robot manipulator is specified as redundant or kinematically redundant when it possesses more DOF than is needed to execute a given task. A redundant manipulator can be exploited to achieve more dexterous robot motions and for collision avoidance.

6.9 Conclusion

This chapter has discussed a variety of approaches to collision avoidance methods for a multi-arm robot system; and the importance of using a suitable collision avoidance algorithm for effective task execution while avoiding possible collision. The chapter has also examined the control strategy and the implementation, testing and evaluation of a kinematics control algorithm for the Hydro-Lek multi-arm system and tested on the planar manipulator and the Hydro-Lek arms. Finally the chapter has reviewed tests on the kinematics control algorithm used for the Hydro-Lek multi- arm system at Lancaster University, and undertaken at the Sugano Laboratory at Waseda University, Tokyo, Japan (2008) as part of the collaborative research project. The robot arms cooperation method and collision avoidance method implemented by Sugano Laboratory and Kosuge & Hirata Laboratory are illustrated in Appendix C.

In conclusion it was decided that the kinematics control algorithm approach would not be a suitable approach to apply for the Hydro-Lek arms in order to facilitate collision avoidance.

The disadvantage of this approach is that it can only be applied with redundancy DOF of a robot arm, whereas Hydro-Lek arms only possess 6 DOF. Although the research found that the kinematics control algorithms are easy to implement but are only suitable for applications which possess a redundant manipulator with a specified end-effector trajectory. The adoption of the force potential field algorithm developed by Sugano Laboratory can be applied to MARS-ND with the master-slave coordination system. The master-slave control strategy was

adopted for MARS-ND because it uses a hierarchical control scheme for the two manipulators which assigns different level priorities for the two arms in order to execute their given tasks. The control system can configure the slave arm's trajectory with constraints that will prevent it colliding with the master arm while in motion. The constraints on the slave arm trajectory generation can be used for many purposes including a joint's angular velocity and acceleration control. The master-slave control strategy has been applied by many researchers in the area of multi-arm robotic systems due to the ease of implementation.

When the kinematics control algorithm was tested on both the two Hydro-Lek arms and the 6-DOF planar manipulators, it was found that the algorithm worked well for the 6-DOF planar manipulator but not so well for the Hydro-Lek arms. This 6-DOF planar manipulator possessed a different joint configuration to the Hydro-Lek arms and had high redundancy. When tested with the kinematics control algorithm it gave a good result and prevented a potential collision between the robot and the obstacle. In the test on the Hydro-Lek manipulators the potential collision was prevented and the manipulator had no redundancy to reconfigure the joints to continue tracking the path without collisions. The primary task, however, to trace a specified end-effector trajectory, was also suspended. The reason for the failed test when using the Hydro-Lek arms can be summarised as follows:

1. Hydro-Lek arms is not a redundant manipulator

The tests undertaken at the Sugano Laboratory, using the kinematics control algorithm with a 7-DOF planar manipulator, were compared with new tests on the Hydro-Lek arms based on a simulation using MatLab. The following changes were made under these tests:

(a) Horizontal and vertical bars were used

(b) Different distances between the bar (task) and robot arm end-effectors

(c) Drawing and redrawing the selected robot arm

(d) Saving the animation of each trajectory generated for the selected arm

In response to the tests undertaken regarding the kinematics control algorithm both at Lancaster University and Sugano Laboratory, the following recommendations can be made:

1. More experiments are needed to test the kinematics control algorithms with different types of manipulators. For example adjusting the joint configurations of the manipulator; using 7 or more joints; adjusting link lengths; and adjusting task location before each test
2. The program code of the algorithm may need amendment or improvement. It would then need to be re-tested again for effectiveness and responsiveness
3. The condition of the manipulator redundancy needs to be satisfied in order to fully implement the kinematics control algorithm

The next stage for the advancement of MARS-ND beyond this thesis, therefore, is the development of coordination and collision avoidance algorithms applied with suitable software. The following final chapter will present the conclusions to this thesis.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

The novel aspect to this thesis broadly focuses on two issues which can be divided into two sections, the development of MARS-ND; and the investigation of solutions to the problems of coordination and collision avoidance for a multi-arm robot system such as MARS-ND. Chapters 3, 4 and 5 outlined the development of MARS-ND using modern tools and explained how these tools can be integrated and interfaced together. The integration and interface processes for MARS-ND used a modern development path which requires the engineers to be able to move between different engineering disciplines, and integrate theory and practice in the building of a robotic system. The traditional path for developing a robotic system requires a team of engineers to work together, each engineer with a specified and defined role. Chapters 6 discussed the investigation and finding of a solution to the problem of coordination and collision avoidance for a multi-arm robot system such as MARS-ND. As part of these investigations research collaboration was undertaken with Sugano laboratory at Waseda University, Tokyo, Japan.

7.1 Methodological results

The underlying premise of this thesis was to investigate, study, build, integrate and interface a multi-arm robot system for nuclear decommissioning applications using tested off-the-shelf tools. This process involved the use of the latest technology by a single engineer, this requiring a mechatronics approach to the development of the robotic system. The use of the

latest technology as the basic tools allowed this research to integrate all the different parts purchased and interface them using an easy to use Programmable Automation Controller and single operating software for operating, obtaining feedback, monitoring and updating the robot system. As discussed in Chapter 3, the specific tools chosen for MARS-ND were selected because of the ease with which they communicated with one another, and could be integrated and controlled from one operating system. The application of mechatronics concepts within this research allowed me to develop MARS-ND without the need for external expertise in terms of software and hardware integration, programming and motion control of the robot. These choices proved to be an effective approach to the development of MARS-ND in terms of both the functionality of the robot with respect to the off-the-shelf tools selected; and the ability for one researcher to integrate the software and hardware systems and to build the motion control profile.

The design of the Hydro-Lek arms received for this research did not meet our original design specifications; therefore the weight of the attached Hydro-Lek multi-arm system exceeded the maximum payload of the Brokk arm. This made it essential to apply the front stabiliser of the Brokk machine to counter balance the excessive loads.

The use of LabVIEW operating software facilitated the creation of motion control for the robot arm because of the ability of Lab VIEW to integrate and communicate with all of the off-the-shelf tools within a single user interface. This is a different approach to previous research projects which have used specific pieces of software for each tool, creating several user interfaces to facilitate control of the robot arm. The application of LabVIEW within this research simplified this process because of the use of one user interface.

The design of a motion controller in LabVIEW, as discussed in Chapter 4, provided an interface with external sensors and a simulation package. This type of controller is called a Low Level Controller. The development and application of a single motion controller for the

Hydro-lek arms with the use of LabVIEW as the operating software is a step forward from previous projects which have used many different controllers and software for the various movements of the robotic arms. Thus the use of LabVIEW and the Low Level Controller simplified the motion control process of the Hydro-lek arms. Valve calibration was undertaken in order to help provide the arm joints with meaningful input values. This process was carried out by normalising the input voltage of each joint into input demands. The low-level controller design is based on the individual joint control strategy. In this strategy each joint of the robot arm was modelled by applying a trial and error method with the PID gains using LabVIEW PID toolkit. The effectiveness of the low level controller for each joint of the robot arm was tested. This method was chosen because of the relative ease of its application.

Chapter 4 also commented on the issues that emerged during the research process in relation to structural problems of the Hydro-Lek HLK-7W arms. The problems highlighted the importance of carrying out FK and IK as part of the design process, before the manufacturing stage. The problems also demonstrated the necessity for rigorous research within the robotics industry.

Using 3D robotic simulation software, the FK and IK of the Hydro-Lek arms can be formed and used to build the low-level and high-level controllers for MARS-ND. These issues were demonstrated in Chapter 5. Chapter 5 also showed that the motion control of the robot arms can be improved significantly, in terms of operator flexibility and control of the robot arms, when third party devices such as SpacePilot and USB joystick are interfaced with the LabVIEW high-level controller of MARS-ND. The advantages and limitations of these third party devices became evident during the simulation, testing and experimentation with the Hydro-Lek arms.

7.2 Collision avoidance algorithm for MARS-ND

The coordination of two or more robot arms, the classification of coordinated motion control algorithms, and the selected arm coordination approach for MARS-ND was examined. Master-slave motion control algorithm was applied with MARS-ND. In this approach one arm controls motions (the master) while the other arm kinematically follows and is responsible for complying with the interactive force (the slave) (T. Ishida 1977). The kinematics of the multi-arm robot system is controlled in such a way as to keep the slave arm in a symmetrical relationship to the master arm during point-to-point motion (A. Hemami 1986). Simple control strategies for multi-arm robot systems can be derived from control strategies for single arm systems (A. K. Ramadorai *et al.* 1994).

The type of coordination method used for the robot TWENDY-ONE and how it was implemented using their motion controller, has the most significance for the development of a system for arm coordination and collision avoidance for MARS-ND. This modelling method models each joint of the arm as a sphere, and then exploits the kinematics of both arms with the use of force potential field method. The task execution using this motion controller is achieved using position control, posture control and force control. This coordination and control method is useful for MARS-ND because the kinematics of MARS-ND can also be exploited and used with the force potential field method for collision avoidance. In order to explore this it is first necessary to model the arm coordination of MARS-ND which can then lead to the implementation of the Sugano arm collision avoidance technique using the force potential field method. This process is discussed in more detail in Appendix C. If it was decided, however, to use this identical method for MARS-ND then it would be necessary to adopt the Sugano idea of collision avoidance using force potential field method into the LabVIEW environment. Many modifications need to be carried out to the LabVIEW control system currently applied with MARS-ND, because the control system used at Sugano Laboratory was built using a traditional programming language with its own architecture.

Another possible solution that would allow application of the Sugano method is to use NI MathScript within the LabVIEW environment. This would allow the Sugano potential field method to be imported into the LabVIEW user interface and then simplified and linked with the MARS-ND motion control VI.

7.3 Experimental results

Chapter 6 outlines two experiments that were undertaken to examine the control strategy; and the implementation, testing and evaluation of a kinematics control algorithm for collision avoidance for:

1. The Hydro-Lek multi-arm system and a planar manipulator with 6-DOF
2. The Hydro-Lek multi-arm system and a planar manipulator with 7-DOF

The first experiment reviewed tests on the kinematics control algorithm used for the Hydro-Lek multi-arm system at Lancaster University using one Hydro-Lek arm and an obstacle; the second experiment tested the same algorithm at the Sugano Laboratory at Waseda University using the Matlab models of Hydro-Lek arms.

Through the results of these experiments, it became clear that the kinematics control algorithm approach would not be a suitable solution to the problem of collision avoidance for the Hydro-Lek multi-arm robot system. The reasons being that this approach can only be applied to redundant robot arms, whereas the Hydro-Lek arm possessed 6-DOF. In reality, however, robot arm redundancy implies 6-DOF or more. The first simulation tests carried out at Lancaster University showed little effect and no response from the collision avoidance algorithm function within the program code when the configuration of the Hydro-Lek arm was used to carry out a straight line movement using its end-effector while an obstacle was placed under the arm. When the 6-DOF planar robot arm with a different joint configuration was applied to undertake the same task, the arm followed its trajectory until it moved close to

the obstacle and then stopped to avoid the obstacle. The positive result of this test was that the minimum distance function for calculating the distances between the planar arm segments and the obstacle responded, but not fully, for the collision avoidance algorithms to allow the arm to continue following its path until it reached the specified goal for its end-effector. The simulation tests may have failed because of lack of redundancy of the Hydro-Lek robot arm.

The second simulation tests carried out at Sugano Laboratory, compared two Hydro-Lek arms and two 7-DOF planar manipulators. The test when using two 7-DOF planar manipulators for executing a task on horizontal and vertical bars showed improved results compared to the tests carried out at Lancaster University. The arms executed their given trajectories perfectly but the collision avoidance algorithms failed to respond when a potential collision between the two arms appeared to be imminent. The results of the experiments led to the following conclusions as to why the collision avoidance algorithm may not have responded as expected, however, further tests would be necessary to clarify these conclusions:

1. The method used for implementing the collision avoidance algorithm during the programming process may not have been well structured and executed
2. The type of planar robot arm used may have been unsuitable with this algorithm
3. The tests at Sugano Laboratory proved that a redundancy DOF robot arm must be used when the kinematics control algorithm is applied for collision avoidance. This is a condition of this algorithm and possibly a limitation. Further tests are also therefore necessary to prove the effectiveness of this algorithm for multi-arm robot systems

This thesis has also laid out the groundwork for the kinematics control algorithm and the following suggestions have been made for further work that may help achieve the collision avoidance needed when a redundancy multi-arm robotic system is used:

1. The use of advanced distance calculation algorithm. This research project has used a simple distance calculation algorithm that represents objects as line segments or points. The algorithm is fast for computation and easy to implement, but it has less accuracy. Most of the practical manipulators, however, have complex shapes and

cannot be simply represented as lines or points. It is therefore recommended to implement an algorithm that can calculate the minimum distances between complex convex objects such as the GJK algorithm. The GJK algorithm was presented by E. G. Gilbert *et al* (1988) for computing the minimum distance between complex objects in 3-dimensional space.

2. Slave manipulator end-effector path planning. The collision avoidance control for a multi-arm robotic system still needs a separate path planned for the end-effector. In addition, the generated path for the slave arm should not be disturbed by the master arm or the path re-planned when it has been disturbed.
3. Adjustment on variables. In the kinematics control algorithm there are some adjustable variables such as d_i, d_u and v_o. It will be useful to find more information regarding how these variables influence the collision avoidance control algorithm.
4. Decision making for using the pseudo-inverse of matrices. This refers to the problem mentioned in the test on the Hydro-Lek manipulators. Sometimes the algorithm yields a very large number which causes the kinematics control loop to not have the desired joint velocity in order to satisfy the specified tolerance. The result of this is that the program continues looping until it reaches the looping limit. This phenomenon was also reported by A. A. Maciejewski and C. A. Klein (1985) and some possible solutions were discussed in their paper.

7.4 Novel aspects, achievements and contributions of this research

- The design, building and testing of a multi-arm robot system for nuclear decommissioning tasks by a sole researcher using a mechatronics approach
- Studying and identifying specific modern off-the-shelf tools for the development of MARS-ND
- Integration of two Hydro-Lek arms and a universal bracket for the formation of a multi-arm system

- Integration of the multi-arm system and a Brokk mobile platform for the development of MARS-ND
- Development and application of hardware and software interfaces
- Using single operating software for the design of a simple PID motion control system to control the multi-arm robot system. Testing the PID algorithms with the Hydro-Lek arm joints individually to test their effectiveness. These tests were positive and therefore demonstrated that the simple PID algorithms developed for the Hydro-Lek arms were successful
- Specifying an arms coordination system for the multi-arm system
- Finding an appropriate collision avoidance algorithm for the multi-arm robot system when undertaking a given task; and assessing the risk of potential collision between the two arms. Testing a kinematically controlled collision avoidance algorithm on Hydro-Lek multi-arm system at Lancaster University, then improving and re-testing it at Sugano Laboratory. The findings of these experiments using simulation created a foundation for this algorithm to be developed further. It also showed that in order for the algorithm to work it needs arms with redundancy. The Hydro-Lek arms, although apparently possessing adequate redundancy in actuality did not possess sufficient redundancy to meet the requirements for MARS-ND
- Undertaking research collaboration with Sugano laboratory at Waseda University in Tokyo, Japan in order to develop further understanding and to find a suitable collision avoidance algorithm for MARS-ND

7.5 Recommendations for future work and research

Recommendations for future research can similarly be divided into the methodological and application areas. To begin with the methodological developments, the most obvious avenue for future research to emerge from this thesis would be to continue and fully develop and test the collision avoidance algorithm for MARS-ND by adopting the Sugano method, the force

potential field algorithm. The Sugano method needs to be used in conjunction with the master-slave arm coordination control strategy that was adopted for MARS-ND because it uses a hierarchical control scheme for the two manipulators which assigns different level priorities for the two arms in order to execute their given tasks. The Sugano method will need to be tested within the LabVIEW environment or by importing the C-code using NI MathScript which is an add-on tool in LabVIEW software. The reason for this is that the motion control system for MARS-ND is built and implemented within the LabVIEW operating software.

In terms of practical applications of this research, existing research has focused on the use of vision systems as a feedback mechanism for the operator in the area of robotic systems for use in hazardous nuclear applications where humans are not allowed to operate. It is therefore recommended that the next stage of this research is to obtain vision systems and interface them with the LabVIEW environment for feedback. The advantages of using a vision system with MARS-ND are as follows:

1. To help the operator precisely track the movement of the end-effectors when executing given tasks
2. To enhance the implementation of a collision avoidance method for potential collision between the two manipulators
3. To increase operator confidence when working with the robot while it is in a separate location

Other recommendations for future research include:

1. The use of the Proportional-Integral-Plus (PIP) control system introduced by Young *et al.* (1987) as a low level controller of all of the robot joints for improved motion control for MARS-ND
2. Further developments in MARS-ND are likely to focus on the high level control system that supervises the pipe cutting task and other selected tasks with the use of

additional sensing devices such as a vision system such as cameras as mentioned earlier

3. Similarly research using MARS-ND for pipe cutting tasks may benefit from the use of force control sensors to support the existing positional controller
4. To obtain and use a specialised joystick for MARS-ND to provide set points for position and force control
5. The consideration and use of on board rechargeable battery so that the MARS-ND system become completely wireless

To conclude, MARS-ND is a good example of a robotic system specifically designed for hazardous nuclear decommissioning applications. It demonstrates the complexity of such a system from a number of aspects such as the need for mobility, control, sensor and system design, and integration using modern tools that are available off-the-shelf. In addition the use of these modern tools allows a single mechatronics engineer to design, integrate, interface and build a motion control system for MARS-ND as compared to the traditional way of building a similar robot by a team of specialised engineers.

REFERENCES

ABSL Power Solutions: http://www.abslpower.com/ [accessed March 2009]

Aerial Venn diagram: http://www.eng.mu.edu/~craigk/ [accessed April 2007]

Aditya, A., B. Riyanto., (2000). "Implementation of Java 3D Simulation for Internet Telerobotic System". *Proceedings IASTED International Conference on Modelling and Simulation*, Pittsburgh, PA, USA.

Agate, R., Pace, C., Seward, D., Shaban, E., and Bakari, M., (2006). "Control Architecture Characteristics in Autonomous Mobile Construction Robots". *23rd International Symposium on Automation and Robotics in Construction (ISARC)*. Tokyo, Japan.

Alford, C. O., and Belyeu, S. M., (1984). "Coordinate control of two robot arms". *Proceedings of IEEE International Conference of Robotics*, pp. 468-473.

Al-Jarrah, O., and Y. F. Zheng., (1994). "Efficient trajectory planning for two manipulators to deform flexible material". *Proceedings of International Conference on Intelligent Robots Systems*, vol. 1, pp. 1056–1063.

Anauth, S., and Goldenberg, A. A., (1992). "Identification of kinematic parameters of multiple closed chain robot manipulators working in coordination", *Proceedings of ICRA.*

Anthony, Lai., (2005). "Meeting the Demands of Robotic Space Applications with Compact PCI". Aitech Defense Systems.

Anderson, J. N., (1986). "Decoupled simulation of two-arm robot systems". *Proceedings of American Control Conference*, Seattle, USA, pp.127-129.

Andrade, H. A., and Kovner, S., (1998). "Software synthesis from dataflow models for G and LabVIEW"., *Conference on Signals, Systems & Computers*, vol. 2, pp. 1705-1709.

Arkin, R. C., (1987). "Towards cosmopolitan robots: Intelligent navigation in extended man-made environments". *Technical Report 87-80*, COINS Department, University of Massachusetts.

Arkin, R. C., T. Balch., and E. Nitz., (1993). "Communication of behavioral state in multi-agent retrieval tasks". *International Proceedings IEEE Conference on Robotics and Automation*, vol. 3, pp. 588–594.

A standard IEEE 802.11 (WLAN Standards): http://www.ieee802.org/11/ [accessed May 2007]

Baroth, E. and Hartsough, C. (1995). "Visual Programming in the Real World, *In Visual Object-oriented Programming, Concept and Environment*". *Manning Publications Co.*, Greenwich, CT, pp. 21-42

Baroth, E., Hartsough, C., Holst, A., Wells, G., (1999). "Evaluation of LabVIEW 5.0 and HP VEE 5.0. Part 1", *EE: Evaluation Engineering*, vol. 38, no. 4.

Baroth, E., Hartsough, C., Holst, A., Wells, G., (1999). "Evaluation of LabVIEW 5.0 and HP VEE 5.0. Part 2", EE: Evaluation Engineering, vol. 38, no. 5.

Bahr, N., (1997). "System safety engineering and risk assessment: a practical approach". *Taylor & Francis Inc*, ISBN: 1560324163.

Barraquand, J., Bruno Langlois., and Jean-Claude Latombe., (1989). "Robot motion planning with many degrees of freedom and dynamic constraints". *5th International Symposium Proceedings on Robotics Research.*

Bakari, M. J., Zied, K. M., and Seward, D. W., 2007. Development of a Multi-Arm Mobile Robot for Nuclear Decommissioning Tasks. *International Journal of Advanced Robotic Systems (ARS)*, Vol. 4, No. 4, pp. 387 – 406.

Bakari, M. J., Seward, D. W., Joyce, M. J., and Mackin, R. O., (2008). "The Development of Tele-Operated Two-Armed Mobile Robot System for Nuclear Decommissioning Operations". *2nd International Joint Topical Meeting on Emergency Preparedness & Response and Robotic & Remote Systems*. Albuquerque, New Mexico, USA.

Bakari, M. J., Seward, D. W., Shaban, E. M., and Agate, R. Y., (2006). "Multi-Arm Mobile Robot for Hazardous Nuclear Decommissioning Tasks". *23rd International Symposium on Automation and Robotics in Construction (ISARC)*. Tokyo, Japan.

Bakari, M. J., and Seward, D. W., (2006). "HUMAN-LIKE MECHANICAL MANIPLUATOR, The Design and Development of Multi-Arm Mobile Robot for Nuclear Decommissioning". *3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Setubal, Portugal.

Bejczy, A. K., T. J. Tarn., and X. Tun., (1986). "Coordinated control of two robot arms". International Proceedings IEEE Conference on Robotics and Automation, San Francisco, USA.

Bennett D., and Hollerbach, J., (1988). "Self calibration of single loop, closed kinematic chains formed by dual or redundant manipulators". *27th IEEE Conference Proceedings on Decision & Control.*

Benest, T., and Wise, M., (2002). "Decommissioning the Windscale AGR: The UK demonstration project for four reactor decommissioning". *IBC conference on Nuclear Decommissioning*, UK.

Belanger, P., (1992). "Estimation of angular velocity and acceleration from shaft encoder measurements". *IEEE International Conference on Robotics and Automation*, Nice, France.

Belanger, P. R., P. Dobrovolny., A. Helmy., and X. Zhang., (1998). "Estimation of Angular Velocity and Acceleration from Shaft-Encoded Measurements". *The International Journal of Robotic Research*, vol.17, no.11, pp.1225-1233.

Bien, Z., and J. Lee., (1992). "A minimum-time trajectory planning method for two robots". *IEEE Transactions on Robotics and Automation*, vol.8 (3), pp.414–418.

Bicknell, A., and Hardey, G., (1998). "Factory Automation Adapted for the Decommissioning of Nuclear Reprocessing Facilties". *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, Belgium.

Bobrow, J. E., J. M. McCarthy., and V. K. Chu., (1990). "Minimum-time trajectories for two robots holding the same workpiece," Proceedings of 29th IEEE Conference on Decision and Control, pp.3102-3107.

Brooks, R. A., (1983). "Planning Collision-Free Motions for Pick-and-Place Operations"., *International Journal of Robotics Research*, vol. 2, no. 4, pp. 19-44.

Brokk 40 demolition robot: http://www.castle pryor.co.uk/pdf/Brokk%2040%20tech%20data.pdf [accessed May 2005]

Bruzzone, L. E., R. M. Molfino., and M. Zoppi., (2003). "Mechatronic design of a parallel robot for high-speed, impedance-controlled manipulation". *Proceedings of the 11th Mediterranean Conference on Control and Automation*, Rhodes, Greece.

Carignan, C. R., and D. L. Akin., (1989). "Optimal force distribution for payload positioning using a planar dual-arm robot," *ASME Journal of Dynamics, Systems, Measurements, and Control*, vol. 11 1, pp.205-210.

Candelas, F. A., S. T. Puente., F. Torres., F. G. Ortiz., P. Gil., and J. Pomares., (2003). "A Virtual Laboratory for Teaching Robotics. *International Journal of Engineering Education* (especial issue Remote Access/Distance Learning Laboratories), vol. 19, pp. 363.

Cameron, S., (1998). "Motion Planning and Collision Avoidance with Complex Geometry". *IECON '98, Proceedings of the 24th Annual Conference of the IEEE*, Vol. 4, pp. 2222-2226, Germany.

Challinor, S., (1996). "The Application of Remote Techniques for the Decommissioning of the First primary Separation Plant at Sellafield". BNES *Conference on Remote Techniques for Hazardous Environment*, UK.

Chang, P. H., (1986). "A Closed-Form Solution for the Control of Manipulators with Kinematic Redundancy". *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA.

Chang, C., M. J. Chung., and B. H. Lee., (1994). "Collision avoidance of two robot manipulators by minimum delay time". *IEEE Transactions on Robotics and Automation, Man and Cybernetics*, vol.24 (3), pp/517–522.

Chan, S. F., Prof. R. H. Weston., and K. Case., (1988). "Robot simulation and off-line Programming". *IEEE Journal of Computer-Aided Engineering*, Vol. 5, Issue 4, pp. 157-162.

Chen, I. M., S. H. Yeo., G. Chen., and *G.* L. Yang., (1999). "Kernel for Modular Robot Applications – Automatic Modelling Techniques". *International Journal of Robotics Research*, vol.18(2), pp.225 -242.

Chen, I. M., and G. L. Yang., (1998). "Inverse Kinematics for Modular Reconfigurable Robots". *International Proceedings of IEEE Conference on Robotics and Automation*, pp. 1647-1652.

Chen, W. H., G. L. Yang., and K M. Goh., (2002). "Kinematic Control for Fault-Tolerant Modular Robots Based on Joint Angle Increment Redistribution". *Proceedings of 7th International Conference on Control, Automation, Robotic and Vision*, pp.396-401.

Cheng, F. T., and Orin, O. E., (1990). "Efficient Algorithm for Optimal Force Distribution-The Compact-Dual LP Method". IEEE Transactions on Robotics and Automation, vol. 6, no. 2.

Colbaugh, R., and Jamshidi, M., 2007. "Robot manipulator control for hazardous waste-handling applications". *Journal of Robotic Systems (JRS)*, Vol.9, No.2, pp.215-250.

Cox, D., (2004). "Cooperative manipulation Testbed Development-Kinematics". *17th Annual Florida Conference on the Recent Advances in Robotics (FCRAR),* Florida.

Corke, P. I. (1996). A Robotics Toolbox for MATLAB., *IEEE Robotics and Automation Magazine*, Vol. 3, No. 1, pp.24-32.

Corke, P. I., (1996). "A Robotics Toolbox for MATLAB". *IEEE Robotics and Automation Magazine*, vol.3 (1), p.24-32.

Cox, J. D., Rackers, K., and Tesar, D., (1995). "Cooperative Manipulation Experiments Using a Dual-Arm Robot". *IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 104-109, Orlando, USA.

Connolly, C.I., Burns, J.B., and Weiss, R., (1990). "Path planning using Laplace's equation". *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, vol.3, pp.2102-2106.

Craig, L., and Hu, H., (2003). "Application of Mobile Agents to Robust Tele-operation of Internet Robots in Nuclear Decommissioning". *Proceedings of IEEE International Conference on Industrial Technology – ICIT 2003*, pp.1214-1219, Slovenia.

Dauchez, P., (1986). "An easy way of controlling two cooperating robots handling light objects at low speed". *25th IEEE conference on Decision and Control*, vol.25, pp.1271-1272.

Daly, P. N., Claver, C. F., (2000). "Real Time Linux and LabVIEW as a control environment for the WIYN tip-tilt module"., *Proceedings of SPIE – The International Society for Optical Engineering*, vol. 4009, pp. 71-78.

Desbats, P., 2005. "Status and trends of remote technology applied to spent fuel management in France". CEA-CEREM/Tele-operation & Robotics Department. *IAEA-TECDOC-1433 'Remote technology applications in spent fuel management'*. International Atomic Energy Agency, ISBN: 92-0-101405-8.

De Silva, C.W., (2007). "SENSORS AND ACTUATORS—Control System Instrumentation". *Taylor & Francis/CRC Press*, Boca Raton, FL.

De Silva, C.W., (2005). "Mechatronics—An Integrated Approach". *Taylor & Francis/CRC Press*, Boca Raton, FL.

Desai, J. P., (2005). "D-H Convention". In: *Robotics and Automation Handbook,* Kurfess, T. R. (Ed.) pp. 8.1- 8.21, CRS Press, Drexel University.

Denavit, J., and Hartenberg, R. S., (1955). "A Kinematic Notation for a Lower-Pair Mechanism based on Matrices". *Journal of Applied Mechanics*, vol.1, pp.215-221.

Dixon, R., Chotal, A., Young, P. C., and Scott, J. N., (1997). "The automation of piling rig positioning utilising multivariable proportional-Integral-Plus (PIP) control". *Proceedings of the 12th International Conference on Systems Engineering, ICSE,* Coventry University, UK, pp.211-216.

Driels, M., and Pathre, U., (1990). "Simulation experiments on parameter identification for Robot Calibration". *International Journal of Advanced Manufacturing Technology,* v.5.

Driels, M., (1993). "Using passive end point motion constraints to calibrate robot manipulators", *Journal of Dynamic Systems, Measurement and Control,* v.115 n0.3.

Dormido, S., (2003). "The role of Interactivity in Control Learning". *Proceedings 6th IFAC Symposium on Advances in Control Education*, Oulu, Finland, pp. 11-22.

Driewer, F., Markus, Sauer., and Klaus, Schilling., (2007). "Discussion of Challenges for User Interfaces in Human-Robot teams". *3rd European Conference on Mobile Robots (ECMR)*, Freiburg, Germany.

Edward, M. S., and Andrew, A. G., (1992). "A Universal Robot Control System (URCS) Based on the TUNIS Multiprocessor". *IEEE Transactions on Industrial Electronics*, Vol.39, No.3.

Erdmann, M., and T. Lozano-Perez., (1986). "On Multiple Moving Objects". *Proceedings IEEE International Conference on Robotics and Automation*, pp.1419-1424.

European Website on Decommissioning of Nuclear Installation, "decommissioning in Short", 2003. (http://www.eu-decom.be/about/initabout.htm)

Esposito, C.; Sullivan D.; Frank U. & Cibulskis R. (1993). "Field application of robotic systems in hazardous waste site operation". In: *Robotics and Remote System for Hazardous Environments*, Jamshidi, M. & Eicker, P. (Eds), pp. 1-31, PTR Prentice Hall, Englewood Cliffs, New Jersey, USA.

Falter D. D. *et al.* (1995). "Remote Systems for Waste Retrieval from the Oak ridge National Laboratory Gunite Tanks". *American Nuclear Society Meeting*, San Francisco, USA.

Fidani, A., and Baraona, P., (2001). "The challenges of industrial remotely operated systems for the 21st century". *9th International Topical meeting on Robotics and Remote Systems*, Washington.

Fisher, W. D., (1984). "Kinematic control of redundant manipulators". *PhD Thesis*, School of Electrical Engineering, Purdue University.

Fountain, Tim., (1999). "LabVIEW moves into the real-time domain",. *Electronic Engineering* (London), vol.71, no.867, pp.16-18.

Fong, T., I. Nourbakhsh., and K. Dautenhahn., (2003). "A survey of socially interactive robots". *Robotics and Autonomous Systems*, vol.42, pp. 143-166.

Freund, E., and Hoyer, H., (1985). "On the On-Line Solution of the Findpath Problem in Multi-Robot Systems"., *The 3rd International Symposium on Robotics Research, Gouvieux, France* (MIT Press, Cambridge, MA).

Freund, E., and Hoyer, H., (1986). "Path finding in Multi-Robot Systems: Solution and Applications"., *Proceedings IEEE International Conference on Robotics and Automation*, pp. 103-111.

Fu, K. S., Gonzalez, R. C., and Lee, C. S. G., (1987). "Robotics, Control, Sensing, Vision, and Intelligence". *McGraw-Hill*, New York.

Fukuda, T., S. Kawauchi., and H. Asama., (1990). "Analysis and evaluation of cellular robotics as a distributed intelligent system by communication information amount". International *Proceedings IEEE/RSJ Workshop on Intelligent Robots Systems*, pp. 827–834.

Gerken, D. J., (1998). "Quick, good, and inexpensive"., *Laboratory Robotics and Automation*, vol. 2, pp. 95-97.

Girone, M., G. Burdea., M. Bouzit., V. Popescu., and J. E. Deutsch., (2001). "A Steward Platform-Based System for Ankle Tele-rehabilitation". *Autonomous Robots*, pp.203-212.

Goldberg, K., R. Siegwart., (2002). " Beyond Webcams: An Introduction to Online Robots". The MIT Press.

Goldenberg, A.A., Benhabib, B., and Fenton, R.G., (1985). "A Complete Generalized Solution to the Inverse Kinematics *of* Robots", *IEEE Journal of Robotics and Automation,* vol. RA-I, No.1, pp.14-20.

Griesmeyer, J. M., McDonald, M. J., Harrigan, R. W., (1992). "Generic Intelligent System Controller (GISC)"., *Sandia Internal Report*, SAND92-2159, Sandia National Laboratories, New Mexico.

Gu, J., Taylor, J., and Seward, D., (2004). "Proportional-Integral-Plus Control of an Intelligent Excavator". *Computer-Aided Civil and Infrastructure Engineering*, vol.19, pp.16-27.

Gupta, K.C., Kazerounian, K., (1985). "Improved Numerical Solutions of Inverse Kinematics of Robots". *IEEE International Conference on Robotics and Automation,* vol.1, pp.743-147.

Hanafusa, H., and Asada, H., (1977). "A robot hand with elastic fingers and its application to assembly process". *Proceedings of IFAC Symposium on Information Control Problems in Manufacturing Technology,* pp.127-138.

Hashinuma, H. *et al.* (2002). "A Tele-operated Humanoid Robot Drives a Lift. Truck," *Proceedings of IEEE International Conference on Robotics and Automation.*

Hanselmann, H., (1998). "Development speed-up for electronic control systems". *Proceedings of Convergence 98 - International Congress on Transportation Electronics*, pp.1-15. Dearbon, USA.

Hayes-Roth, F., Erman, L. D., London, R., and Terry, A., (1992). "Tools and Methods for Developing Distributed Intelligent Control and Management (DICAM) Application Systems"., *Position Paper for the Joint Services Guidance and Control Committee (JSGCC) Software Initiative Workshop.*

Hägglund, T., and K. J. Åström., (2004). "Revisiting the Ziegler-Nichols step response method for PID control." Journal of Process Control, pp. 635–650.

Hayashi, A., Park, J. and Kuipers, B. J., (1990). "Toward Planning and Control of Highly Redundant Manipulators". *Fifth IEEE International Symposium on Intelligent Control.*

Hayati, S., (1986). "Hybrid Position/Force Control of Multi-Arm Cooperating Robots". *Proceedings of IEEE International Conference on Robotics and Automation*, pp.82-89.

Harrigan, R. W., (1993). "Automating The Operation of Robots in Hazardous Environments"., *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, July 26-30.

Hemami, A., (1986). "Kinematics of two-arm robots". IEEE Journal of Robotics and Automation, vol. RA-2, no. 4, pp.225-228.

Herrera-Bendezu, L. G., Mu, E., and Cain, J. T., (1988). "Symbolic computation of robot manipulator kinematics". *Proceedings of IEEE International Conference on Robotics and Automation*, Philadelphia, USA, vol.2, pp.993-998.

Hemami, A., (1987). "On a Human-Arm Like Mechanical Manipulator". *Robotics*, vol. 5, pp.23-28.

Hemami, A., (1986). "Kinematics of Two-Arm Robots". *IEEE Journal of Robotics and Automation,* vol. RA-2 (4), pp. 225-228.

Hirose, S. and Ma, S., (1991). "Coupled Tendon-driven Multijoint Manipulator". *IEEE Conference on Robotics and Automation*, Sacramento, CA, pp.1268-1275.

Hirose, S., and S. Amano., (1993). "The Vuton: High Payload, high efficiency, holonomic omni-directional vehicle". *JSME Annual Conference on Robotics and Mechatronics*, pp.350-355.

Honda's ASIMO humanoid robot: http://world.honda.com/ASIMO/ [accessed May 2008]

Hollerbach, J. M., (2000). "Some Current Issues in Haptics Research". *International Proceedings of IEEE Conference on Robotics and Automation.*, San Francisco, USA, pp.757-762.

Hollerbach, J. M., (2000). "Some Current Issues in Haptics Research". *International Proceedings of IEEE Conference on Robotics and Automation.*, San Francisco, USA, pp.757-762.

Hoadley, D., siegel, J., and Scarfe, D. (1998). 2In-vehicle data acquisition, transfer, and real-time processing"., Darren Electronic Engineering (London), vol.70, no.862, pp.37-38.

Hydro-Lek: http://www.hydro-lek.com/ [accessed April 2005]

IAEA 2003, IAEA 1995, IAEA 1999, IAEA 2001: safety standards on decommissioning strategies: http://www-ns.iaea.org/standards/documents/related-docs.asp?sub=180 [accessed July 2005]

International Atomic Energy Agency (IAEA), (2001). "Methods for The Minimization of Radioactive Waste From Decontamination and Decommissioning of Nuclear Facilities". *Technical report, no. 401*, Vienna.

International Atomic Energy Agency (IAEA) (2001). "State of the Art Technology of Decommissioning and Dismantling of Nuclear Facilities". *Technical Report, no. 395*, Vienna.

Isermann, R., (1996). "On the Design and Control of Mechatronic Systems-A Survey". *IEEE Transactions on Industrial Electronics*, Vol.43, No.1, pp.4-15.

Ishida, T., (1977). "Force Control in Coordination of Two Arms". *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 717-722.

Iwata, H., S. Kobashi., T. Aono., T. Kobayashi., and S. Sugano., (2007). "Development of 4-DOF Anthropomorphic Tactile Interaction Manipulator with Passive Joint". *Journal of Robotics and Mechatronics*.

Iwasawa, N., M. Uchiyama., and K. Hakomori., (1987). "Hybrid position/force control for coordination of a two-arm robot". International Proceedings IEEE Conference on Robotics and Automation, Raleigh, NC, pp.1242-1247.

Iwata, H., and S. Sugano., (2004). "Human Robot Interference Adapting Control Coordinating Human Following and Task Execution". *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System*, Sendai, Japan.

James Fisher Nuclear: http://www.jfnuclear.co.uk/ [accessed April 2006]

Java micro edition (J2ME) mobile phone using Bluetooth. Java 2007: http://www.javaworld.com/channel_content/jw-j2me-index.shtml.

Judd, R. P., and Knasinski, A. B., (1987). "A technique to calibrate industrial robots with experimental verification". *Proceedings of ICRA.*

Kazuhiro, Kosuge., H. Kakuya., and Y. Hirata., (2001). "Control Algorithm of Dual Arms Mobile Robot for Cooperative Works with Human". *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. TA10-3.

Kazuhiro, Kosuge., Manabu, Sato., Norihide, Kazamura., (2000). "Mobile Robot Helper". *Proceedings of IEEE International Conference on Robotics and Automation*, pp.583-588.

Kamezaki, M., H. Iwata., and S. Sugano., (2008). "Development of an Operational Skill-Training Simulation for Double-Front Construction Machinery, Training Effect for a House Demolition Work". *Journal of Robotics and Mechatronics*, vol.20, no.4, pp.602-609.

Kamezaki, M., Iwata, H., Sugano, S., (2007). "Development of an Operational Skill-Training Simulation for Double-Front Construction Machinery". *Proceedings of the JSME Conference on Robotics and Mechatronics,* Japan.

Kant, K., and S. W. Zucker., (1986). "Toward efficient trajectory planning: The path-velocity decomposition". *International Journal of Robotics Research*, vol.5, no.3, pp.72-89.

Kapoor, C., and Tesar, D., (1999). "Kinematic Abstractions for General Manipulator Control". *ASME Design Engineering Technical Conferences,* Las Vegas.

Khatib, O., K. Yokoi., O. Brock., K. Chang., and A. Casal., (1999). "Robots in Human Environments: Basic Autonomous Capabilities". *The International Journal of Robotics Research,* vol. 18, no. 7, pp. 684-696.

Khatib, O., (1986). "Real-time obstacle avoidance for manipulators and mobile robots" *International Journal of Robotics Research,* vol. 5, no. 1, pp.90–98.

Khatib, O., (1985). " Real-time obstacle avoidance for manipulators and mobile robots". *IEEE International Conference on Robotics and Automation,* pp.500-505.

Khatib, O., (1987). "A Unified Approach for Motion and Force Control of Robot Manipulator: The Operational Space Formulation". *IEEE Journal of Robotics and Automation*, vol. RA-3, No. 1, pp. 43-53.

King Sun Fu., R. C. Gonzalez., and C. S. G. Lee., (1987). "Robotics: control, sensing, vision, and intelligence". *McGraw-Hill*, New York, NY.

Kirchner H.O.K., *et al.* (1987). "A perturbation approach to robot calibration". International Journal of Robotic Research, vol.6, no.4.

Killough, S., Martin, H., and Hamel, W., 1986. "Conversion of a servomanipulator from analog to digital control". *IEEE International Conference on Robotics and Automation*. Vol.3, pp.734-739.

Klein, C. A., and Huang, C. H., (1983). "Review of pseudoinverse control for use with kinematically redundant manipulators". *IEEE Transactions on Systems, Man and Cybernetic,* vol.13, pp. 245–250.

Kozlowski, K., (1998). "Modelling and Identification in Robotics". *Springer Verlag London Limited*, Great Britain.

Koga, M., Kosuge, K., Furuta, K., and Nosaki, K., (1992). "Coordinated motion control of robot arms based on the virtual internal model". *IEEE Transactions on Robotics and Automation*, vol.8, no.1, pp.77-85.

Koditschek, D. E., (1987). "Exact robot navigation by means of potential functions: some topological consideration". *Proceedings of IEEE International Conference on Robotic and Automation*, pp.1-6.

Kobayashi, H., (1985). "Control and geometrical considerations for an articulated robot hand". *International Journal of Robotics Research*, vol.4, no.1, pp.2-12.

Kosuge, K., S. Sato., and K. Furuta., (1990). "Task-Oriented Control of Master-Slave Manipulator". *Proceedings of Japan-USA Symposium on Flexible Automation*, Kyoto, Japan.

Krogh, B. H., (1984). "A generalized potential field approach to obstacle avoidance control". *International Robotics Research: The Next Five Years and Beyond.* Society of Manufacturing Engineers.

Kuban, D. P., M. W. Noakes., and E. C. Bradley., (1987). "The Advanced Servo manipulator System: Development Status and Preliminary Test Results". *Proceedings of the ANS Topical Meeting on Remote Systems and Robotics in Hostile Environments*, Pasco, Washington, pp. 638-644, 1987.

Kuffner, J. J., K. Nishiwaki., S. Kagami., Y. Kuniyoshi., M. Inaba., and H. Inoue., (2001). "Efficient leg interference detection for biped robots". *19th Annual Conference of Robotics Society of Japan,* Tokyo, Japan.

Kurono, S., (1975). "Coordinated Computer Control of a Pair of Artificial Arms". Bio mechanism 3, The Tokyo University Press, pp.182-193.

Lee, B. H., and C. S. G. Lee., (1986). "Collision –free motion planning of two robots". *IEEE Transactions on Systems, Man and Cybernetics*, vol.17, no.1, pp.21-32.

Lee C.S.G and Ziegler, M., (1982). "A Geometric Approach in Solving the Inverse Kinematics of PUMA Robots". *Proceedings of the 13th International Symposium on Industrial Robots*, Chicago, USA, pp.61-1 to 61-18.

Lewis, F.L., Dawson, D.M., and Abdallah, C.T., (2004). *"Robot manipulator Control Theory and Practice"*. 2nd ed. New York: Marcel Dekker.

Lewis, F. L.; Dawson, D. M. & Abdallah, C.T. (2004). "*Robot Manipulator Control Theory and Practice*". 2nd ed. Marcel Dekker, New York.

Lees, M. J.; Taylor, C. J.; Young, P. C. & Chotai, A. (1998). "Modelling and PIP control design for open-top chambers". *Control Engineering Practice*, Vol. 6, pp. 1209-1216.

Lee, S., (1989). "Dual redundant arm configuration optimization with task oriented dual arm manipulability". *IEEE Transactions on Robotics and Automation*, vol.5, no. 1, pp. 78–97, 1989.

Lee, B. H., and C.S.G. Lee., (1987). "Collision-free motion planning of two robots". *IEEE Transactions on Systems, Man, and Cybernetic*, vol. SMC-17, no.1, pp.21-32.

Lee, S. W., Lee, B. H., and Lee, K. D., (1998). "A configuration space approach to collision avoidance of a two-robot system". *Cambridge University Press*, Vol. 17, No. 2, pp. 131-141, USA.

Lim, J., and D. H. Chyung., (1985). "On a control scheme for two cooperating robot arms". *24th International Proceedings IEEE Conference on Decision and Control*, Fort Lauderdale, FL, USA, pp. 334-337.

Liegeois, A., (1977). "Automatic supervisory control of the configuration and behaviour of multibody mechanisms". *IEEE Transactions on Systems, Man and Cybernetic,* vol. 7, pp. 868–871.

Lin, M.C., Manoucha, D., and Canny, J., (1994). "Fast contact determination in dynamic environments". *Proceedings of IEEE International Conference on Robotics and Automation*, vol.1, pp.602 – 608.

Lozano-Perez, T., (1981). "Automatic Planning of Manipulator Transfer Movements"., *IEEE Trans. Syst., Man, Cybern.* SMC-11, no. 6, pp. 681-698.

Luh, J. Y. S., and Campbell, C. E., (1984). "Minimum Distance Collision-Free Path Panning for Industrial Robot with Prismatic Joint"., *IEEE Transaction on Automatic Control.* AC-29, no. 8, pp. 675-680.

Luh, J.Y.S., Zheng, Y.F., (1987). "Constrained relations between two coordinated industrial robots", *International Journal of Robotics Research*, Vol. 6 pp.60-70.

Lumelsky, V.J., (1984). "Iterative Coordinate Transformation Procedure for One Class of Robots". *IEEE Trans. Systems, Man, and Cybernetics,* vol.14, pp.500-505.

Lyons, D., (1986). "Tagged potential fields: An approach to specification of complex manipulator configurations". *IEEE International Conference on Robotics and Automation,* pp.1749-1754.

Mark, D. McKay., (1993). "Idaho national engineering laboratory decontamination and decommissioning robotics development program". *Test/demonstration report for robotic plasma arc cutting for D&D applications*, Westinghouse Idaho Nuclear Company, Inc.

Marian, F. A., and H. T. Rowan., (1987). "The Cost-Benefit of Robotic Devices in Nuclear Power Plants", *Public Service Electric and Gas Company*, NJ, USA.

Mäntylä, M., and Andersin, H., *Enterprise systems integration*, Helsinki, HUT, 1998, 313p.

Marin, R., P. J. Sanz., A. P. del Pobil., (2002). "A predictive Interface Based on Virtual and Augmented Reality for task Specification in a Web Telerobotic System". *Proceedings IEE/RSJ Int. Conference on Intelligent Robots and Systems*, Lausane, Switzerland, pp. 3005-3010.

Maslowski, A., P. Szynkarczyk., A. Czerniewska–Majewska., A. Andrzejuk., and L. Szumilas., (2002). "Simulation System for Analysis of Specialized Mobile Robots". *Computer-Aided Civil and Infrastructure Engineering*, vol. 12, Issue 1, pp. 5-14.

Maciejewsky, A. A., and Klein, C. A., (1985). "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments". *International Journal of Robotics Research,* vol. 4, pp.109–117.

Maciejewsky, A. A., and Klein, C. A., (1985). "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments". *International Journal of robotic research*, vol.4, pp.109-117.

Martin, M., (1992). "Draft Volume I of Next Generation Workstation/Machine Controller (NGC) Specification for an Open System Architecture Standard (SOSAS)". *Document No. NGC-0001-13-000-SYS.*

McKee, G. T., (2003). "An online robot system for projects in robot intelligence". *International Journal of Engineering Education* - Especial Issue Remote Access/Distance Learning Laboratories, vol.19, pp.356.

Meredith, M., and Maddock, S., (2004). Real-Time inverse kinematics: the return of the Jacobian. In:*Department of Computer Science Research Memorandum*, CS-04-06, University of Sheffield, Sheffield, U.K.

Milenkovic, V., and Huang, B., (1983). "Kinematics of Major Robot Linkages". *Proceedings of the 13th International Symposium on Industrial Robots*, vol.1 pp.16-31 to 16-47.

Miller, D. J., (1993). "Standards and Guidelines for Intelligent Robotic Architectures", *Proceedings of AIAA Space Programs and Technologies Conference and Exhibit.*

Miyabe, T., Konno, A., Uchiyama, M., and Yamano, M., (2004). "An Approach toward an Automated Object Retrieval Operation with a Two-Arm Flexible Manipulator". *The International Journal of Robotics Research*, vol. 23, No. 3, pp. 275-291.

Mota, A., Jose, A. F., and Santos, F., (1998). "Low cost data acquisition systems based on standard interfaces". *Proceedings of the IEEE International Conference on Electronics, Circuits, an Systems*, vol. 3, pp. 433-437.

Mort, P. E., (1998). "Size Reduction Techniques for the First Primary Separation Plant at Sellafield". *IMechE/BNES International Conference on Nuclear Decommissioning (DeCom)*, London.

Moon, S. B., and S. Ahmad., (1990). "Time optimal trajectories for cooperative multi-robot systems," Proceedings 29th IEEE Conference on Decision and Control, pp.1126-1127.

Moon, S. B., and S. Ahmad., (1993). "Time-Optical trajectories for cooperative multi-robot systems". *IEEE Transactions on Systems, Man, and Cybernetic.*

Moon, S. B., and S. Ahmad., (1991). "Time-scaling of cooperative multirobot trajectories". *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, pp. 900–908.

Myers, John. K., (1985). "Multi-arm collision avoidance using the potential field approach". International *Space Station Automation,* pp.78-87.

Nakamura, H., (2005). "The state of Japanese remote technology in nuclear fuel cycle". Japan Nuclear Cycle Development Institute. *IAEA-TECDOC-1433 'Remote technology applications in spent fuel management'*. International Atomic Energy Agency, ISBN: 92-0-101405-8.

National Instruments (NI) LabVIEW [online]. Available from: http://www.ni.com/ [accessed 2006].

National Instruments Compact FieldPoint 2120 Conroller Benchmarks and Performance [online]. Available from: http://zone.ni.com/devzone/cda/tut/p/id/3668 [accessed 2008].

Nakamura, Y., H. Hanafusa., and T. Yoshikawa., (1987). "Task priority based redundancy control of robot manipulators". *International Journal of Robotic Research*, vol.6 (2), pp.3-15.

Nakano, E., S. Ozaki., T. Ishida., and I. Kato., (1974). "Co-operational Control of the Anthropomorphosis Manipulator 'MELARN'". *Proceedings of the 4th International Symposium on Industrial Robot*, Toyko, Japan, pp.251-260.

Newman, W. S., and N. Hogan., (1986). "High speed robot control and obstacle avoidance using dynamic potential functions". *Technical Report TR-86-042*, Philips Laboratories.

Nethery, J. F., and M. W. Spong., (1994). "Robotica: a Mathematica package for robot analysis". *IEEE Robotics and Automation magazine*, vol.1 (1), pp.13-20.

NETGEAR wireless USB: http://www.ciao.co.uk/NETGEAR_WG111T_108_Mbps_Wireless_USB_2_0_Adapter__Review_5647470 [accessed May 2007]

NI measurement and Automation Explorer (MAX): http://techteach.no/tekdok/usb6009/ [accessed April 2006]

Noakes, M, W., (1998). "Remote Dismantlement Tasks for the CP5 Reactor: Implementation, Operation, and Lessons Learned". *Spectrum 98 International Conference on Decommissioning and Decontamination and on Nuclear Hazardous Waste Management*, Denver, USA.

Noakes, M. W., L. J. Love, and P. D. Lloyd, "Telerobotic Planning and Control for DOE D&D Operations," *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., May 11–15, 2002, Vol. 4, pp. 3485–3492.

Noakes, M. W., D. C. Haley., and R. Vagnetti., (2000). "Deployment of Remote Systems in U.S. Department of Energy Decontamination and Decommissioning Projects". *Proceedings of the American Nuclear Society Spectrum 2000*, Chattanooga, Tenn.

Nuclear Decommissioning Authority Final Strategy (2006). (http://www.nda.gov.uk/strategy/).

Ollero, A., S. Boverie., R. Goodall, J. Sasiadek, H. Erbe., and D. Zuehlke., (2006). "Mechantronics, robotics and components for automation and control: IFAC milestone report". *Annual Reviews in Control*, vol.30, Issue 1, pp. 41-54.

O'Toole, K., and Salopek, P., (1998). "Next generation graphical development environment for test". *AUTOTESTCON (Proceedings)*, pp. 145-148.

Paper 5: Waste and decommissioning (2006). An evidence-based report by the Sustainable development commission with contribution from Nirex and AMEC NNC. (www.sd-commission.org.uk).

Parker, L. E., and Drapper, J. V., (1998). "*Robotics Application in Maintenance and Repair*". 2nd ed. New York: Wiley.

Paul, P. R., Renaud, M., and Stevenson, C. N., (1984). "A Systematic Approach for Obtaining the Kinematics of Recursive Manipulators Based on Homogeneous Transformations". *Robotics Research, The First International Symposium*, pp.706-726.

Paredis, C. J. J., (1996). "An Agent-Based Appmach to the Design of Rapidly Deployable Fault Tolerant Manipulators". *PhD Dissertation*, Carnegie Mellon University, USA.

Park, F. C., (1994). "Computational Aspect of Manipulators via Product of Exponential Formula for Robot Kinematics". *IEEE Trans. on Automatic Control*, vol.39 (9), pp. 643-647.

Paul, R., (1981). "Robot manipulator: Mathematics, programming". *MIT Press*, Cambridge, Mass.

Paul, R. P., Shimano, B., and Mayer, G. E., (1981). "Kinematic Control Equations for Simple Manipulators". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, pp.449-455.

Paul, R. P., (1982). "Robot Manipulators: Mathematics, Programming and Control". *The MIT Press*.

Pimental, J. R., (1990). "*Communication Networks for Manufacturing*". Englewood Cliffs, NJ: Prentice-Hall, pp.418-446.

Pleinevanx, P., and J. D. Decotignie., (1998). "Time critical communication networks: Field buses," *IEEE Network*, vol.2.

Ramadorai, A. K., Tzyh-Jong, Tara., Bejczy, A. K., and Ning, Xi., (1994). "Task-driven control of multi-arm systems". *IEEE Transactions on Control Systems Technology*, vol.2, Issue 3, pp.198-206.

Research Opportunity for Deactivation and Decommissioning: Department of Energy Facilities, National Academy Press, Washington DC, 2001.

Red, W. E., and Dixon, M., (1990). "Survey of Robotic Enabling Technologies for WINCO/DOE". *Study sponsored by WINCO/DOE and conducted for Idaho Nuclear Engineering Laboratories (INEL) under contract C85-110766 (HRS-200-90).*

Red, W. E., and Troung-Cao, H. V., (1987). "Configuration Maps for Robot Path Planning in Two-Dimension"., ASME *Journal of Dynamic Systems, Measurement and Control*, pp. 292-298.

Richardson, B. S., D. C. Haley, A. M. Dudar, and C. R. Ward, "A Mobile Automated Characterization System (MACS) for Indoor Floor Characterization," *Proceedings of the 6th Topical Meeting on Robotics and Remote Systems*, Monterey, Calif., February 5–10, 1995.

"Robots deliver real benefits for PSE&G". *Nuclear Engineering International*, vol. 35, no. 435, pp. 120-121.

Robertshaw, E., (2007). *MSc dissertation.* Lancaster University, UK.

RODDIN, Crane-deployed work platform information [online]. Available from: http://www.cybernetix.fr/modules.php?&op=modload&name=cybernetix&file=product&top=14&bt=nucleairenewlang=eng [accessed November 2005]

Rolf, Isermann., (2007). "Mechantronic systems- Innovative products with embedded control". *Control Engineering Practice*, vol.16, Issue 1, pp.14-29.

Rolf, Isermann., (1996). "Digital Control Systems". Springer-Verlag New York, Inc. Secaucus, NJ, USA .

Robinson, Robert L., (1998). "Landing gear system simulation using LabVIEW"., *Scientific Computing & Automation*, vol.15, no.11.

Roth, Z., Mooring, B. W., and Ravani, B., (1987). "An overview of robot calibration". *Journal of Robotics and Automation*, vol. RA-3.

Rossol, L., (1993). "Nomad Open Architecture Motion Control Software". Proceedings *of the International Robots and Vision Automation Conference*, Detroit, MI.

Salvatore, Cavalieri., Antonella, Di Stefano., and Orazio, Mirabella., (1995). "Pre-Run-Time Scheduling to Reduce Schedule Length in the FieldBus Environment". *IEEE Transactions on Software Engineering*, vol.21(11), pp.865-880.

Salvador, B. M., A. P. del Pobil., and M. P. Francisco., (1998). "Very fast collision detection for practical motion planning Part I: The spatial representation". *Proceedings of IEEE International Conference of Robotics and Automation*, pp. 624-629.

Salisbury, J. K., and B. Roth., (1983). "Kinematic and force analysis of articulated mechanical hands". *Journal of Mechanical, Transmission, Automation in Design*, pp.35-41, no. 105.

Salisbury, J.K., and J.J. Craig., (1982). "Articulated Hands: Force Control and Kinematic Issues". *International Journal of Robotica Research*, vol.1, pp. 4-17.

Schilling Manipulators: http://www.schilling.com/index.php [accessed November 2005]

Schreier, Paul G., (1999). "In-Depth Evaluation of Embedded LabVIEW". *EE Evaluation Engineering*, vol.38, no.11.

Sciavicco, L., and B. Siciliano., (1996). "Modelling and control of robot manipulators". McGraw-Hill, NY, USA.

Seto, F., Kazuhiro, Kosuge., and Yasuhisa, Hirata., (2005). "Self-collision Avoidance Motion Control for Human Robot Cooperation System using RoBE". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.50-55.

Seto, F., K. Kosuge., and Y.Hirata., (2004). "Real-time Self-collision Avoidance System for Robots using RoBE". *International Journal of Humanoid Robotics*, vol. 1, no. 3, pp. 533-550.

Seto, F., K. Kosuge., R. Suda., and H. Hirata., (2003). "Real-time control of self-collision avoidance for robot using RoBE", *IEEE International Conference on Humanoid Robots*, Karslruhe–Munchen, Germany.

Seward, D., (1999). The Development of Intelligent Mobile Robots. PhD Thesis, Lancaster University, UK.

Seward, D. W., and Bakari, M. J., (2005). "The use of Robotics and Automation in Nuclear Decommissioning". *22nd International Symposium on Automation and Robotics in Construction (ISARC)*. Ferrara, Italy.

Sheth, P. N., and Uicker, J. J., (1971). "A generalized symbolic notation for mechanisms". *ASME Journal of Engineering for Industry*, vol.93, no.1.

Shin, Y., and Z. Bien., (1989). "Collision-free trajectory planning for two robot arms". *Robotica 7*, Part 3, pp.205-212.

Sklar, M., (1987). "Geomtric calibration of industrial manipulators by circle point analysis". *Proceedings of Second Conference on Recent Advances in Robotics*.

Smith, M. H., (1999). "Towards a more efficient approach to automotive embedded control system development". *Proceedings of the 1999 IEEE International Symposium on CACSD*, pp.219-224, USA.

Sorenti, P., (1997). ""Efficient Robotic Welding in Shipyards – Virtual Reality Simulation Holds the Key". Industrial Robot, vol.24 (4), pp.278-281.

Software Development Kits (SDKs). (Microsoft 2007): http://www.microsoft.com/downloads/details.aspx?familyid=06111A3A-A651-4745-88EF-3D48091A390B&displaylang=en.

Stevens, R., Brook, P. Jackson., K. and Arnold, S., (1998). "Systems engineering, coping with complexity". *Prentice Hall Europe*, ISBN: 0-13-095085-8, London.

Sugano Laboratory (2008). Department of Mechanical Engineering, School of Science and Engineering, Waseda University, Tokyo, Japan.

Tarn, T. J., A. Bejzy., A. Ishidori., and Y. Chen., (1988). "Nonlinear feedback in industrial robot and manipulators". *International Journal of Robotic Research*, vol.1, no1.

Takase, K., (1985). "Representation of constrained motion and dynamic control of manipulators under constrained". *Transactions of SICE*, vol.21, no.5, pp.508-513.

Tarn, T. J., Bejczy, A. K., and Yun, X., (1985). "Design of dynamic control of two coordinated robots in motion". *Proceedings of 24th IEEE Conference on Decision and Control*, pp.1761-1765.

Taylor, C. J.; Chotai, A. & Young, P.C. (1998). Proportional-integral-plus (PIP) control of time delay systems, *Proceeding of the Institution of Mechanical* Engineers, *Part 1:]Journal of systems and Control Engineering* Vol. 212 No. 1, pp. 37-48, ISSN:0959-6518.

Terry, L. Huntsberger., Ashitey, Trebi-Ollennu., Hrand, Aghazarian., Paul S. Schenker., Paolo, Pirjanian., and Hari, Das Nayar., (2004). "Distributed Control of Multi-Robot Systems Engaged in Tightly Coupled Tasks". *Journal of Autonomous Robots*, Springer Netherlands, vol. 17, No. 1, pp. 79-92.

Thomas, M., and Tesar, D., (1982). "Dynamic Modelling of Serial Manipulator Arms". *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 218-228.

Todt, E., Rausch, G., and Suarez, R., (2000). "Analysis and classification of multiple robot coordination methods". *Proceedings of IEEE International Conference on Robotics and Automation*, vol.4, pp.3158-3163.

Toshio, Tsuji., Achmad, Jazidie., and Makoto, Kaneko., (1997). "Distributed Trajectory generation for Cooperative Multi-Arm Robots via Virtual Force Interactions". IEEE Transactions on Systems, Man, and Cybernetics, vol.27, no. 5, pp.862-867.

Tomislav, Reichenbach., and Zdenko, Kovacic., (2005). "Collision-Free Planning in Robot Cells Using Virtual 3D Collision Sensors". *Journal of Advanced Robotic System (ARS)*, pp. 683-697.

Tsuji, T., (1993). "Trajectory generation of a multi-arm robot utilizing kinematic redundancy". *Journal of Robotics and Mechatronics*, vol. 5, no. 6, pp. 601–605.

Turro, N., O. Khatib., E. Coste-Maniere, (2001). "Haptically Augmented Tele-operation". *International Proceedings of EEE Conference on Robotics and Automation*, Korea, pp.386-392.

Vajta, L., and Juhasz, T., (2005). "3D simulation in the advanced robotic design, test and control". *International Journal of Simulation and Modelling*, vol.4, no.3, pp.105-117.

Vukobratovic, M. and Potkonjak, V., (1982). "Dynamics of Manipulation Robots: Theory and Application". *Springer-Verlag*, Berlin, Germany.

Uchiyama, M., (1990). "A unified approach to load sharing, motion decomposing, and force sensing of dual arm robots". *Robotic research, 5th International Symposium*, MIT Press, Cambridge, MA, pp.225-232.

Uicker, J.J., Denavit, J., and Hartenberg, R.S., (1964). "An Iterative Method for the Displacement Analysis of Spatial Mechanisms", Transactions of ASME. *Journal of Applied Mechanics*, vol.31, pp.309-314.

USB joystick called Predator GM-2500 (Trust Company Products 2007): http://myahead.com/go/look/product.show_product?v_id=20223 [accessed March 2007]

USB SpacePilot (2005). 3D Connexion 2005: http://www.3dconnexion.com/

Wang, F. Y., and B. Pu., (1993). "Planning time-optimal trajectory for coordinated robot arms". *Proceedings of IEEE International Conference on Robotics Automation*, vol.1, pp. 245–250.

West, H., and H. Asada., (1985). "A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment," *IEEE International Conference on Robotics and Automation,* St. Louis, pp. 251-259.

Wen, J. T., and S.H. Murphy., (1991). "Position and force control of robot arms," *IEEE Transactions on Automatic Control*, vol.36 (3), pp. 365-374.

Wen-Tung, Chang., Ching-Huan, Tseng., and Long-long, Wu., (2004). "Creative mechanism design for a prosthetic hand". *Proceedings of the Institution of Mechanical Engineering, Part H: Journal of Engineering in Medicine*, vol.218, no.6, pp.451-459.

Whimey D. E., hzinski C. A., Rourke J. M., (1986). "Industrial robot forward calibration method and results". *Journal of Dynamic* Systems, *Measurement and Control*, v.108, no.1.

Wright, M., (1998). "Multi-computer VXI bus system setup and use". *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 9, pp. 17-21.

Workspace 5 (Flow Software Technologies 2005): http://www.workspace5.com/

Xiao, J. Z., H. Dulimarta., N. Xi., and R. L. Tummala., (2002). "Motion Planning of a Bipedal Miniature Crawling Robot in Hybrid Configuration Space". *International Proceedings of IEEE/RSJ Conference of Intelligent Robots and Systems*, Lausanne, Switzerland, pp.2407-2412.

Yamamoto, M., and A. Mohri., (1991). "A study for coordinated tasks of multirobot system problem statement and task planning for two-robot system". *9th Annual International Proceedings Conference on Robotics Society. Japan,* pp.489–492.

Yanqiong Fei, Y. Q., Ding, Fuqiang., and Zhao, Xifang., (2004). "Collision-free motion planning of dual-arm reconfigurable robots". *Robotics and Computer-Integrated Manufacturing,* vol.20, Issue 4, pp.351-357.

Yang, G. L., W. H. Chen., and H. L. Ho., (2002). "Design and Kinematic Analysis of a Modular and Hybrid Parallel Serial Manipulator". *Proceedings of 7th International Conference on Control, Automation, Robotic and Vision*, Singapore, pp.45-50.

Yang, G. L., W. H. Chen., and I. M. Chen., (2002). "A Geometrical Method for the Singularity Analysis of 3-RRR Planar Parallel Robots with Different Actuation Schemes". *International Proceedings of IEEEKSJ Conference of Intelligent Robots and Systems*, Lausanne, Switzerland, pp.2055-2060.

Yang, H., (2007). 2Collision Avoidance Control for Dual-Arm Manipulators". *MSc Dissertation*, Lancaster University, UK.

Young, P. C., (1996). "A general approach to identification, estimation and control for a class of nonlinear dynamic systems". In M. I. Friswell and J. E. Motherhead (Eds), *Identification in Engineering and Systems*, pp. 436-445.

Young, P. C., Taylor, C. J., Pedregal, D. J., Tych, W., and McKenna, P. G., (2003). "Systems identification, time series analysis and forecasting, the Captain toolbox". *Centre for research on Environmental Systems and Statistics*, Lancaster University.

Young, P.C., Behzadi, M.A., Wang C.L. and Chotai A., (1987). "Direct Digital and Adaptive Control by Input-Output, State Variable Feedback pole Assignment". *International Journal of Control*, Vol.46, pp.1867-1881.

Zheng, Y. H., and J. Y. S. Luh., (1986). "Joint torques for control of two coordinated moving robots". *International proceedings IEEE Conference on Robotics and Automation*, San Francisco, pp.1375-1380.

Zied, K., (2004). Investigation of processes and Tools for Rapid Development of Intelligent Robotic Systems. PhD thesis, Lancaster University, UK.

Zied, K., Seward, D., Dolman, A., and Reihl, J., (2000). "The development of a robotic system for tool deployment in hazardous environment". ISARC 17, Taiwan, pp.179-184.

Zlajpah, L., and B. Nemec., (2002). "Kinematic Control Algorithm for On-Line Obstacle Avoidance for Redundant Manipulators", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp.1898-1903.

10 CFR 20 (1999). "Nuclear Regulatory commission Standards". U.S. Nuclear Regulatory Commission (NRC). USA. (http://www.nrc.gov/about-nrc.html).

APPENDIX A

MODERN OFF-THE-SHELF TOOLS

A.1 LabVIEW Software

LabVIEW software (National Instruments 2005) is a general-purpose programming system that contains a complete library of built-in elements for open connectivity and system integration. It is designed for the development of open connectivity with a broad range of devices to speed test system integration quickly and effectively. The devices include off-the-shelf tools for robotic systems, protocols, and interfaces required by test and measurement applications. Baroth *et al* (1999) conducted a comparison study for the performance of LabVIEW with other well known graphical programming software. They found that the advantages of LabVIEW are its efficient control of PC cards used for data acquisition, instrument control and industrial automation. LabVIEW also offers other features such as multithreading, programmatic menu bar, graphical differencing tools, and translation and documentation tools. LabVIEW simplifies the overall process of designing a user graphical interface for a robotic system for researchers and this is extremely important in the context of this research.

LabVIEW as illustrated in Figure A.1 was chosen as the universal operating software for this research because of its connectivity with a broad range of devices, protocol, and interfaces required for measurement applications. It contains libraries of functions and development tools specifically designed for data acquisition, instrument control and automation control.

LabVIEW also uses VI (as explained in Chapter 2, section 2.3.3) for its graphical programming through which the object in graphical programming simulates the actual instruments. VI has been useful in this research because it has allowed the operator to use the LabVIEW libraries which contain all the necessary icons needed to build the control model for the MARS-ND. This has eliminated the need for expertise in programming language such as C++. The VI structure consists of: *Front Panel*, which is an interactive user interface for data entry and output visualisation. *Block diagram*, which represents the source code for the application and consists of icons connected together by wires through which the data flows. The iconic functions in the *block diagram* can be other VIs, called sub VIs.

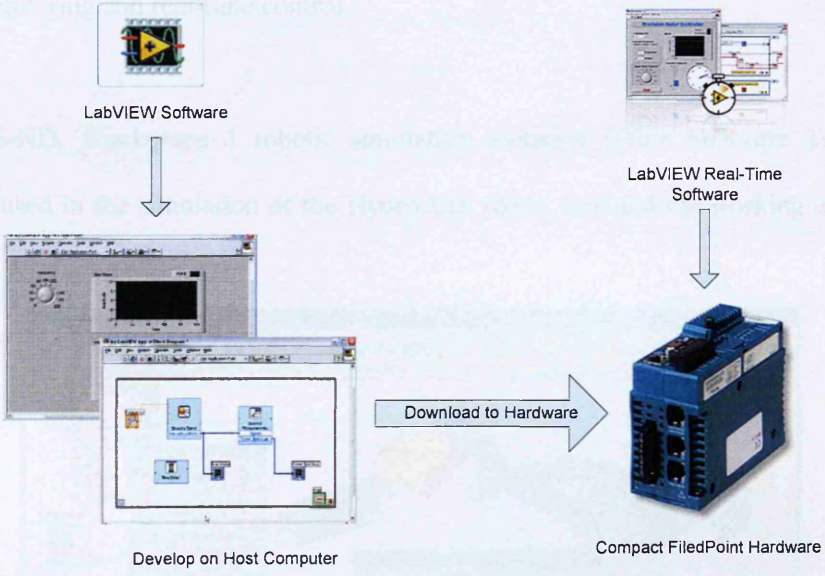


Figure A.1 LabVIEW software (National Instruments 2006)

A.2 Graphical Simulation Software

The graphical simulation process is used extensively in the robotics industry to examine robot programs before they are downloaded to the actual robots. The use of this method reduces the standstill time. The simulation tools are used to aid the choice of manipulator configuration and to test the usefulness of the robot by virtual scenarios. Different simulation packages are used for real-time control for robots such as IGRIP and TELEGRIP (Jo *et al* 1996). These packages can be used in conjunction with sensors to continuously update a virtual robot and

adapt its motion to the new environment information sent by the sensors. From this information a new position can be sent back again to the actual robot. In other situations simulation is used for monitoring the actual robot on-line (Nilson and Johansson 1999). The use of graphical simulation has many applications in the construction and decommissioning industries, which can be summarised as follows:

1. Understanding the kinematics and the configuration of the robot
2. Environmental modelling
3. Tools, equipment and robot modelling
4. Motion planning
5. Monitoring and real-time control

For MARS-ND, *Workspace 5* robotic simulation software (Flow Software Technologies 2004) was used in the simulation of the Hydro-Lek robot, tool and the working environment (see Figure A.2).

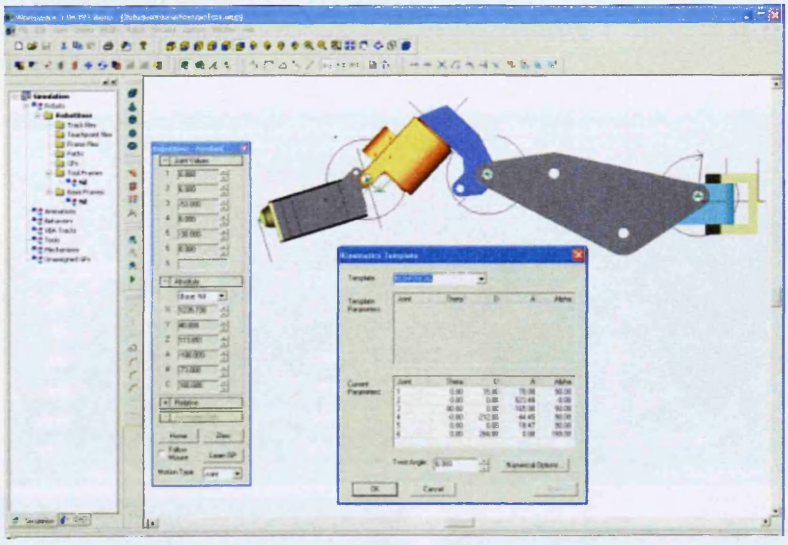


Figure A.2 *Workspace 5* software with D-H table

The graphical simulation process of the robot using *Workspace 5* depends on the Denavit-Hartenberg (D-H) parameters (Denavit-Haretenberg 1955), which can be derived by constructing a kinematical diagram for the Hydro-Lek robot arm. The derivation of the D-H

parameters is discussed in detail in Chapter 4. Figure 3.12 also shows the D-H parameters produced by the *Workspace 5* simulation software.

A.3 SolidWorks 3D Software

SolidWorks is a 3D mechanical Computer-Aided Design (CAD) program. Its core product includes tools for 3D modelling, assembly, drawing, sheet metal and freeform surfacing. It can import numerous file types from other 2D and 3D CAD programs. It also has an API for custom programming in Visual Basic and C. In this research, SolidWorks was used to model and import the 3D aspects of the Hydro-Lek robot arms, assemble them for motion analysis and interface them with NI LabVIEW Software for off-line simulation.

A.4 NI LabVIEW-SolidWorks Mechatronic Toolkit

The NI LabVIEW-SolidWorks Mechatronic Toolkit (National Instruments 2006) is designed to help the development of complex multi-axis motion profiles for the robot manipulators and validate them using simulation.

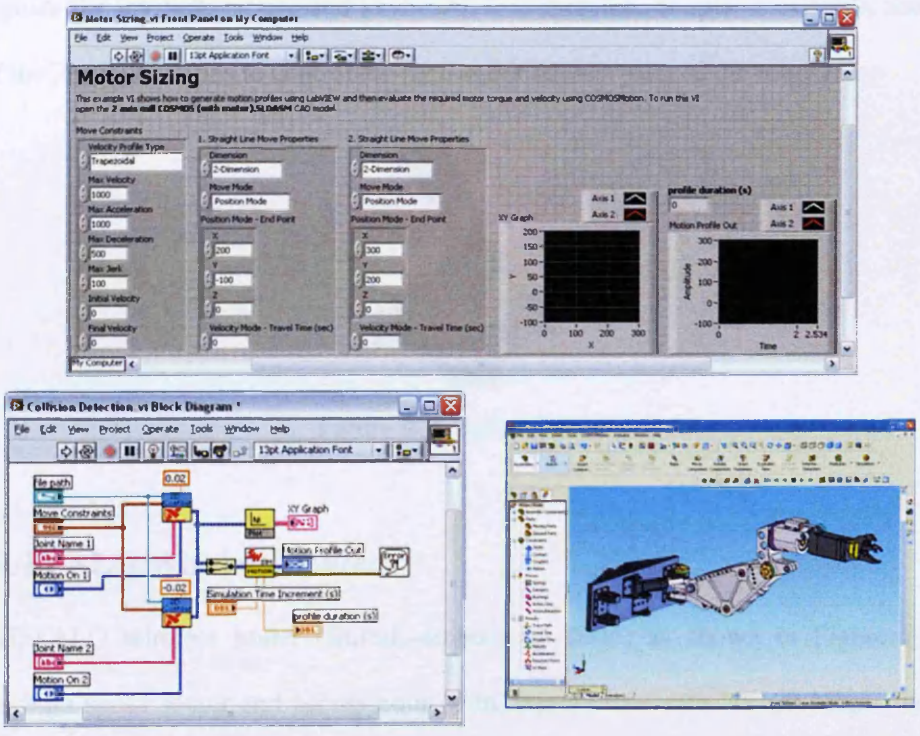


Figure A.3 LabVIEW-SolidWorks Mechatronic Tookit

This toolkit enables engineers to design motion profile; collision detection; simulate the mechanical dynamics of the robot system including mass and friction effects; estimate cycle time performance; and validate component selections for motors, drivers and mechanical transmissions. Through the use of SolidWorks 3D CAD model the simulation of the mechanical and electrical performance of the robot system, as shown in Figure A.3, can be carried out in a short time. To simulate the performance of a machine such as a robot system that contains both mechanical and electrical components in the past, would have been a difficult and time consuming process that required highly specialised expertise.

A.5 USB Joystick

A USB Joystick called Predator GM-2500, shown in Figure A.4, is a 12-user definable buttons joystick with accurate analogue movement control. It is a simple Plug & Play device designed for the gaming industry. It has been used in this research because it can be used without the need for any third party code. The LabVIEW library contains an interface for reading the joystick buttons through the USB port. LabVIEW interface VI has been designed to recognise the joystick, mouse and keyboard. It is therefore simple to calibrate and assign each of the joystick buttons to control the movement of each joint of the robot arms.

Figure A.4 USB Joystick

A.6 BUFFALO Wireless Router

The BUFFALO wireless router (Buffalo-technology 2005) as shown in Figure A.5 is a wireless high speed router and access point with data transfer rate 10/100 Mbps. Its access mode is infrastructure mode and automatically detects and configures a cable or DSL internet

connection. In this research, the BUFFALO router is used to send sensor data signals and receive control signals between the cFP controller and the PC desktop by communicating with a wireless USB NETGEAR.

Figure A.5 BUFFALO wireless router

A.7 Wireless USB NETGEAR

A wireless USB NETGEAR as shown in Figure A.6 provides wireless communication access between the PC desktop and the BUFFALO wireless router connected to the cFP controller. It delivers consistent wireless connections.

Figure A.6 Wireless USB NETGEAR

A.8 Compact FieldPoint 2120 Controller

Compact FieldPoint (cFP) Controller is an expandable programmable automation controller (PAC) composed of I/O modules and intelligent communication interface. The cFP device is used to design embedded control applications for industrial control applications performing:

- Advanced embedded control
- Data logging
- Network connectivity

The PAC shown in Figure A.7 (a) is a reliable platform for rugged industrial environments that demonstrate shock, vibration, and temperature extremes, such as nuclear decommissioning environments. The PAC runs on LabVIEW Real-Time in order to transform signals from the controller into meaningful command signals to the robot manipulators. This provides the functionality, connectivity, and flexibility of NI LabVIEW software

cFP 2120 Specifications:

Processor: 188 MHz processor
Memory: 128 MB non-volatile; 128 MB DRAM
Network: 10BaseT and 100BaseTX Ethernet
Serial Ports: 3 RS232; 1 RS485
Power: 11 to 30 VDC
Weight: 278g
Operating Temp.: -40 to 70 °C

Figure A.7 (a) cFP-2120 controller (National Instruments 2005)

The cFP-2120 controller is also designed for use in intelligent distributed applications such as process and discrete control systems. It is used to open and close valves, run control loops, log data on a centralised or local level, perform real-time simulation and analysis, and communicate over serial and Ethernet networks. The cFP-2120 controller can manage a bank of up to eight Compact FieldPoint I/O modules. The controller is mounted securely on a metal backplane that provides the communication bus as well as a solid surface for the cFP I/O modules and the controller. Through the use of LabVIEW software, programming is simplified with drag and drop functionality. Local or distributed I/O can be added from any Compact FieldPoint bank by dragging I/O from a LabVIEW project to the VI required to read or write to the specific FieldPoint tag. LabVIEW VI with FP Read and FP Write is shown in Figure A.7 (b) and the following is specifically identified:

1. I/O with FP Read icon from cFP-Analog Input-100 module on the cFP-2120 bank added to a new VI. This reads information such as a measurement signal submitted by a sensor, for example the potentiometer sensor or pressure sensor.

2. I/O with FP Write icon from cFP-Analog Output-210 module on the cFP-2120 bank added to the same VI. This sends the control signal needed and can be used, for example, to move the robot arm to a certain angle by controlling the hydraulic actuator valves.

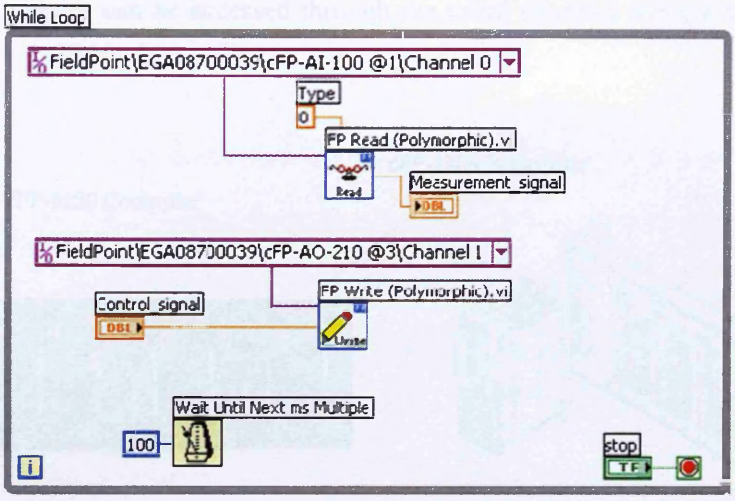


Figure A.7 (b) LabVIEW VI with FP Read and FP Write

This PhD research used a cFP-2120 controller with LabVIEW motion control models downloaded to the cFP embedded controller. The network communication interfaces automatically publish all the measurements with either an Ethernet network or wireless network.

A.9 Ethernet/Serial Interface for Compact FieldPoint

The NI cFP-1808 is designed to communicate with a cFP real-time embedded controller (PAC) and a window computer running LabVIEW. The NI cFP-1808 used in this research is an integrated network interface (Ethernet or serial) and 8-slot backplane as shown in Figure A.8. This NI cFP-1808 interface connects eight cFP I/O modules to a high-speed Ethernet network or to an RS232 serial port. It connects directly to Ethernet networks using a protocol based on standard TCP/IP to maintain full compatibility with existing networks. This network interface monitors the connected I/O modules and publishes I/O data only when the value

changes. The analog signals can change value within selectable ranges called deadbands, without causing the system to report data. This method, along with data compression, helped maximise communication efficiency. The use of RS232 serial communication through direct connectivity allows data to be read and written directly from the PC or embedded controller. In this way I/O modules can be accessed through the serial interface using a serial protocol called Optomux.

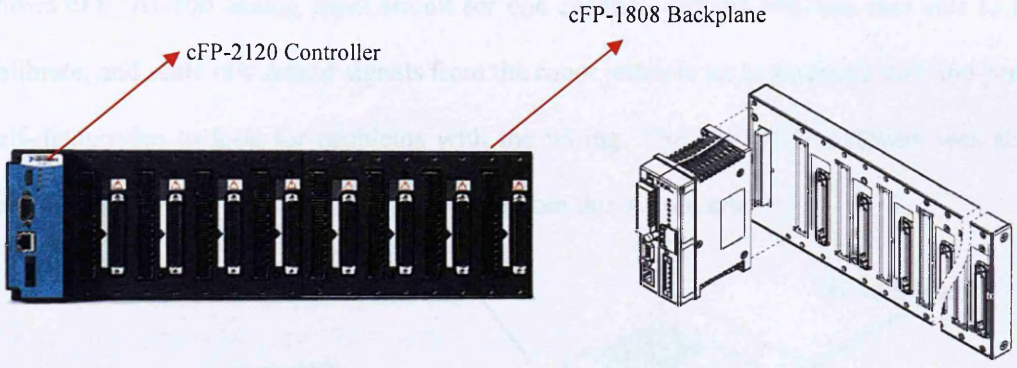


Figure A.8 NI cFP-1808 backplane (National Instruments 2005)

The capability of NI cFP-1808 to act as an interface between the controller, I/O modules and the host PC made it easier to build a flexible, modular distributed measurement and automation system such as MARS-ND.

A.10 Analog Input and Analog Output (I/O) Modules

In this research, cFP-AI-100 with 8 input channels (input type: voltage and current) and cFP-AO-210 modules with 8 input channels (output type: voltage) were used with a Compact FieldPoint controller to measure signals from the sensors and send signals to the actuators. These NI modules were used with Compact FieldPoint built-in signal conditioning to connect directly to high-voltage, milliamp and low-voltage signals; thermocouples, and bridge circuits such as strain gauges. The I/O modules provide a power distribution bridge to power sensors or to connect 2-wire current loops.

A.10.1 cFP-AI-100 Module

The cFP-AI-100 as shown in Figure A.9 (a) has eight single-ended input channels; all eight channels share a common ground reference that is isolated from other modules in the FieldPoint system.

For the building of MARS-ND the analog input module cFP-AI-100, was directly connected to sensors that were under test in order to get high accuracy measurements. Figure A.9 (b) shows cFP -AI-100 analog input circuit for one channel. cFP-AI-100 was also able to filter, calibrate, and scale raw sensor signals from the robot joints to an engineering unit and perform self-diagnostics to look for problems with the wiring. The LabVIEW software was able to read a linearised, calibrated, and scaled value from this AI module.

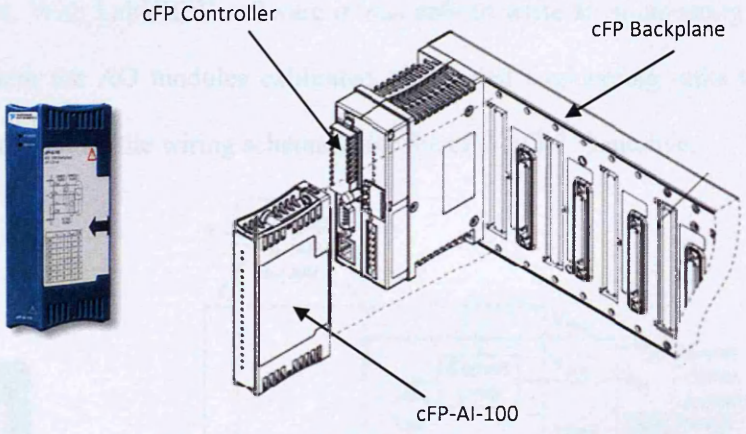


Figure A.9 (a) cFP-AI-100 module (National Instruments 2005)

Each input channel on the AI-100 has four terminals:

1. Voltage input (V_{IN})
2. Current input (I_{IN})
3. Common (COM)
4. Power connection to power field devices or loop powered current loops (V_{SUP})

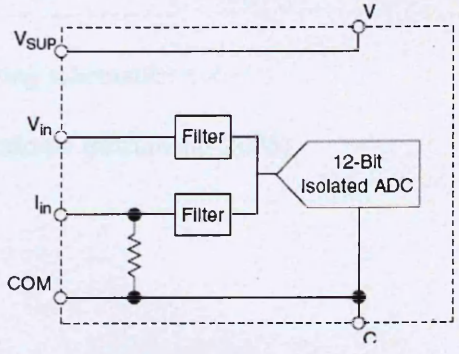


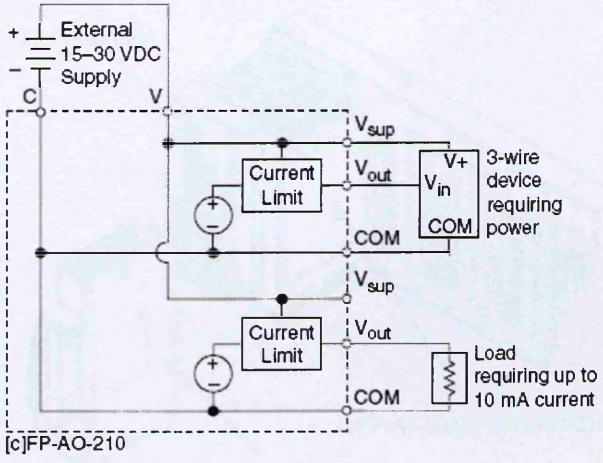
Figure A.9 (b) Analog input circuit

A.10.2 cFP-AO-210 Module

The analog output module cFP-AO-210 as shown in Figure A.10 (a) includes eight 0-10 V output channels, each channel provides a voltage source capable of supplying up to 1mA. Upto 10mA can be obtained by powering it with an optional external power supply of 15 to 30 VDC. The COM terminals of all the channels are connected internally to each other and to the C terminals. The V_{SUP} terminals are all connected to each other and to the V terminals. The cFP-AO-210 has an output range of 0-10V and the factor default power-up setting for each channel is 0V.

For MARS-ND the cFP-AO-210 was connected directly to the actuators in order to obtain high accuracy control. With LabVIEW software it was able to write an engineering value to the AO modules where the AO modules calibrated and scaled engineering units to sensor signals. Figure A.10 (b) shows the wiring schematic for the cFP-AO-210 module.

(a) cFP-AO-210



(b) Wiring schematic

Figure A.10 cFP-AO-210 module (National Instruments 2005)

The cFP-AO-210 has:

1. Eight 0-10 Voltage output terminals (V_{OUT})
2. 16 common terminals (COM)
3. 8 power connections for field devices or current loops (V_{SUP})

A.11 Integrated Connector Block for Wiring to Compact FieldPoint I/O

The cFP-CB-1 as shown in Figures A.11 (a) and A.11 (b) is a general purpose connector block with strain relief. It is designed for hazardous voltage operation and is suitable for use with any NI I/O modules. The cFP-CB-1 contains 36 terminals and also features both a built-in strain relief bar and a separate wire tie connector in order to create secure wiring setups for high shock and vibration applications.

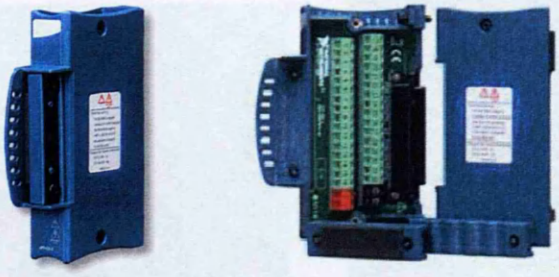


Figure A.11 (a) cFP-CB-1 connector block

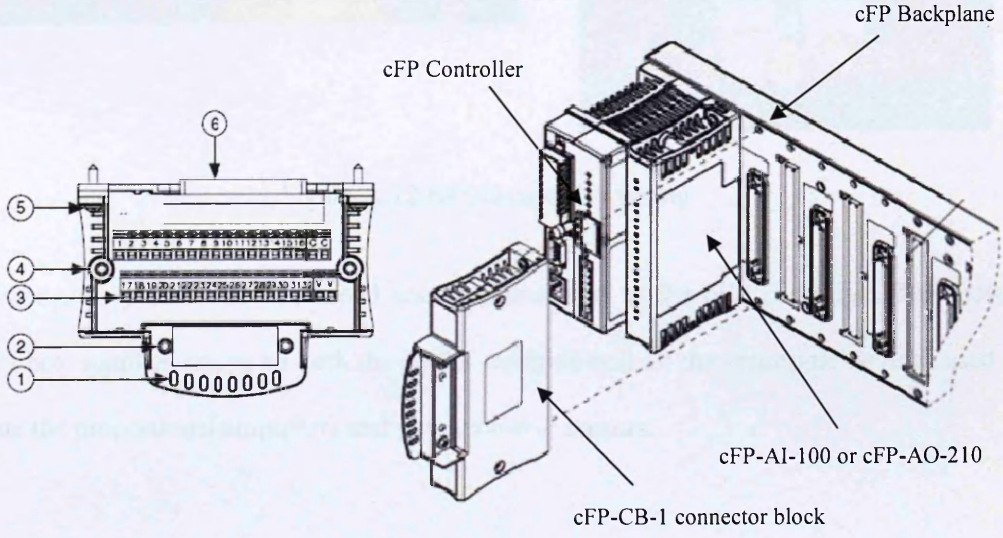


Figure A.11 (b) cFP-CB-1 parts and mounting on cFP backplane (National Instruments 2005)

The cFP-CB-1 parts are:

1. Tie-wrap anchors
2. Screw hole for strain-relief bar
3. Screw terminals
4. Screw hole for top cover
5. Mounting screw
6. 37-pin I/O connector

A.12 Wiring I/O Modules

Three analog output and two analog input modules were used in this research to accommodate the signals sent to and from the robot joint actuators and potentiometer sensors, as they possess sufficient channels for this purpose. Each module was installed with a connector block in a designated slot of the backplane. The power supply and signal cables were wired inside the connector block (see Figure A.12).

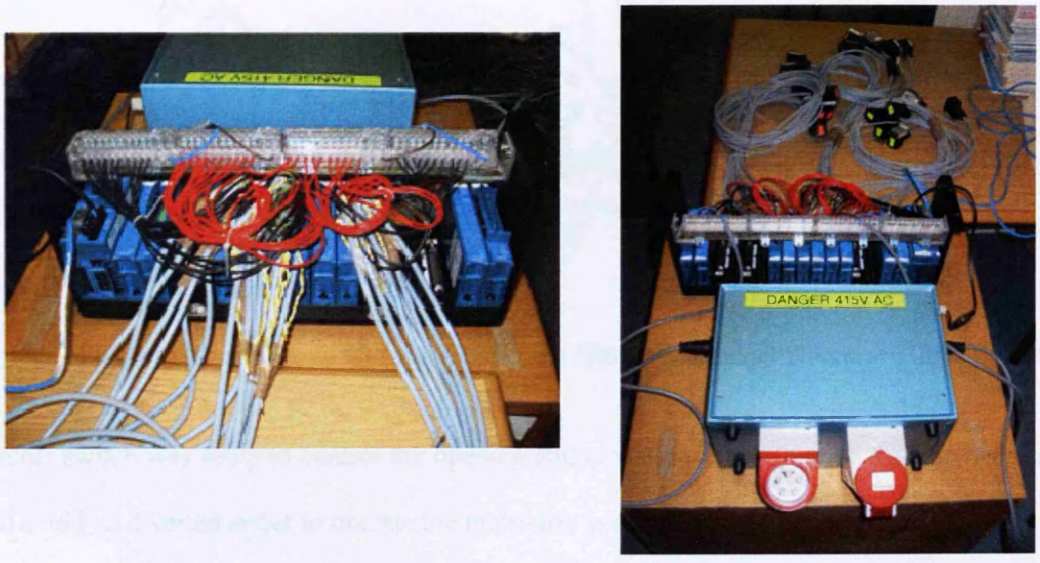


Figure A.12 NI I/O modules wiring

A power supply box was developed and installed next to the cFP controller. This power supply box supplies power to both the Brokk machine and all the electronic devices used to operate the proportional amplifiers and potentiometer sensors.

A.13 Hydraulic System Integration

The BROKK 40 hydraulic system has enough power to drive the Hydro-Lek arms either separately or together. The Brokk 40 robot is designed with an extra hydraulic function for the operation of rotators and other demolition tools such as cutters and drillers as demonstrated in Figure A.13 (a). This extra hydraulic function is a proportional function for applications where precise operation is required. The pressure of the hydraulic function is 175 Bar.

The hydraulic integration is setup in such a way that the hydraulic pump used for operating the Brokk machine also operates the multi-arm system; this occurs through an on/off switch. The

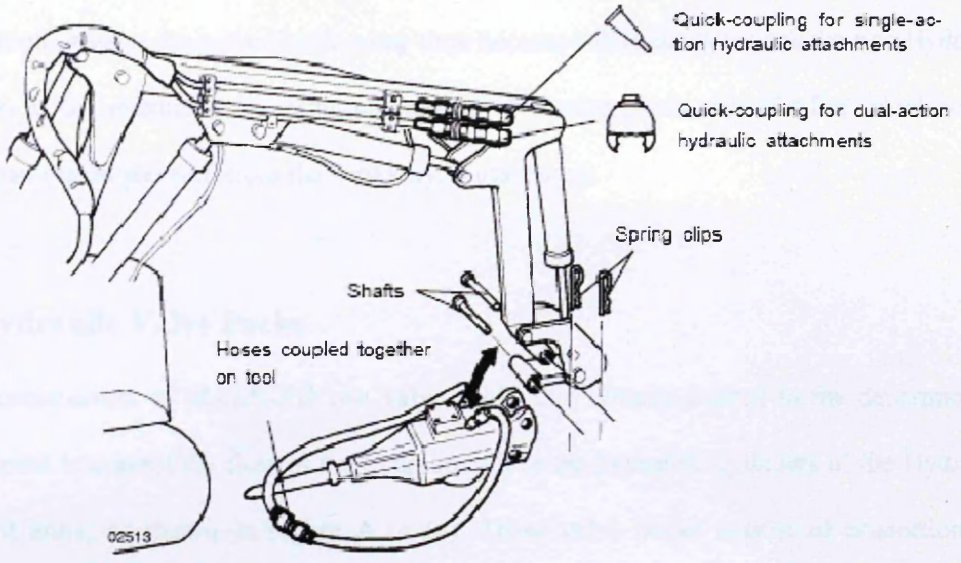


Figure A.13 (a) Brokk 40 tools attachment (Brokk Technical Document 2006)

on/off switch was setup to control the opening and closing of the valves so that the hydraulic oil could be diverted either to operate the multi-arm system or the Brokk machine itself.

Figure A.13 (b) Brokk 40 extra hydraulic functions

Figure A.13 (b) shows the extra hydraulic function in the Brokk machine used to supply the hydraulic oil to and from the valve packs. It became apparent during the initial setup

however, that a pressure difference was needed to operate the solenoid valves. Fortunately, it was found that this problem was due to incorrect fitting of the inlet and outlet hoses to the valve packs. The solution was to fit the inlet and outlet hoses to the correct ports of the valve pack. The oil pressure from the Brokk pump then became sufficient to control the two Hydro-Lek arms, as the maximum operating pressure for the valve packs was 250 Bar which was greater than the oil pressure from the Brokk hydraulic pump.

A.14 Hydraulic Valve Packs

For the construction of MARS-ND two valve packs that already existed in the department were adapted to control the flow of the hydraulic oil to the hydraulic cylinders of the Hydro-Lek robot arms, as shown in Figure A.14 (a). These valve packs consist of proportional solenoid operated spool valves, as shown in Figure A.14 (b), which were used in this research to control the flow direction to and from the hydraulic cylinders.

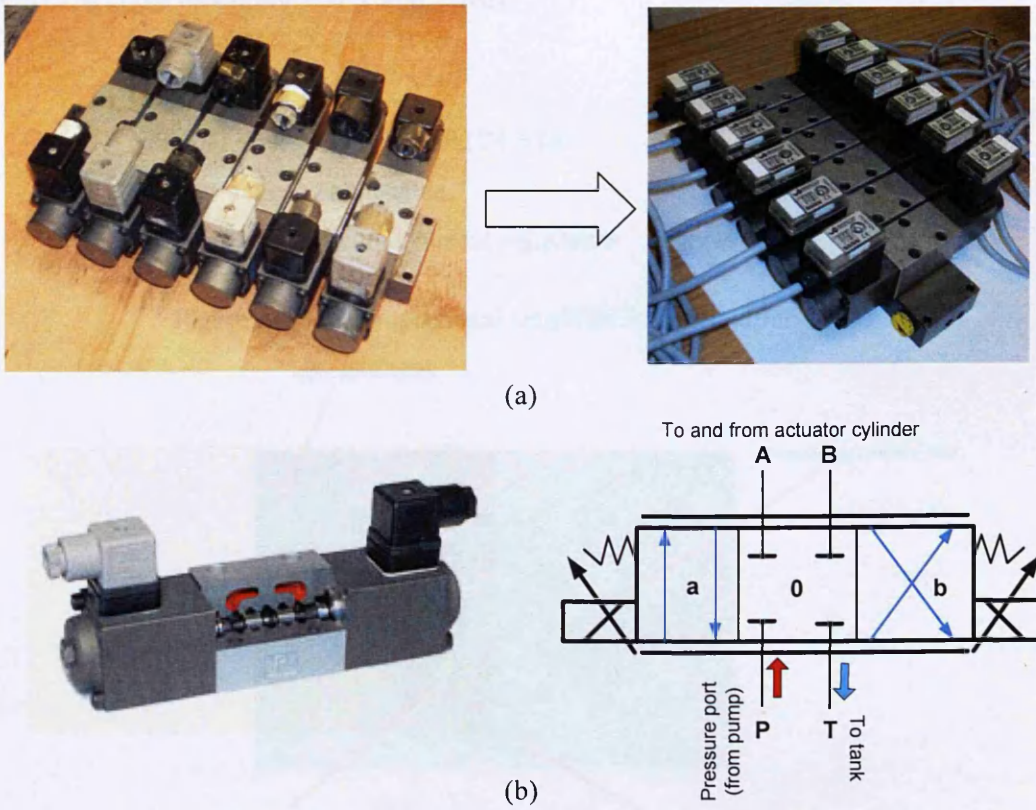


Figure A.14 Valve pack with solenoid spool valves (Wandfluh 2006)

The proportional solenoid spool valves are manufactured by Wandfluh and are attached to a manifold. This type of solenoid valve uses an electromechanical valve with liquid. It is controlled by running or stopping an electrical current through the solenoid, a coil of wire, which in turn changes the state of the valve. Most solenoid valves have two main parts, the solenoid and the valve. The solenoid converts electrical energy into mechanical energy which in turns opens or closes the valve. By controlling the amount of the electrical current the opening and closing of the valve can be controlled proportionally thus controlling the amount of hydraulic oil fed to the joints of the robot arms in order to move them to the desired angles at a required speed.

A.15 Proportional Amplifier P02

The P02 proportional amplifiers used in this research are also standard components manufactured by Wandfluh. The P02 amplifier shown in Figure A.15 (a) was directly installed on to each of the proportional solenoid valves.

Features:
Nominal voltage 12 VDC and 24 VDC
Dither frequency adjustable
Min. and Max. solenoid current adjustable

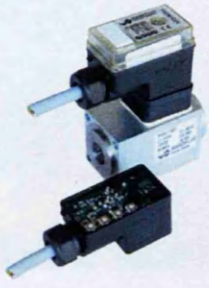


Figure A.15 (a) Proportional amplifier P02 (Wandfluh 2006)

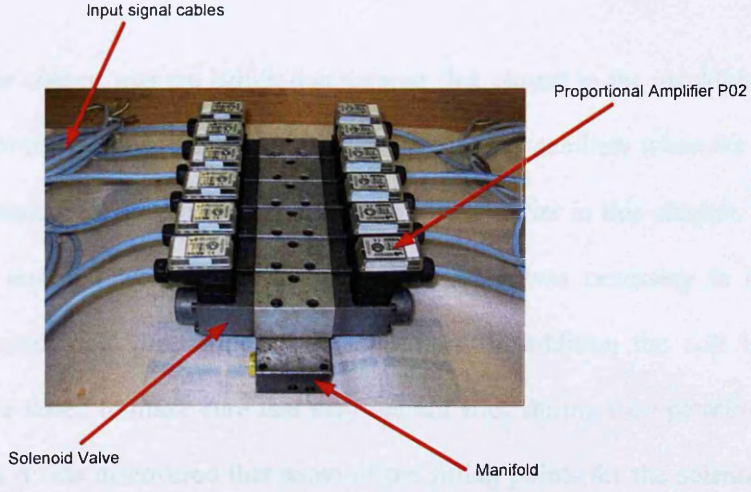


Figure A.15 (b) Plug in proportional amplifier P02

The proportional amplifier is a plug-in device as shown in Figure A.15 (b) and supplies the solenoid valve with current proportional to an input control voltage (0-24 VDC). A superimposed dither reduces the effects of friction and increases the linearity of the valve. In order to open and close the oil ports to a desired proportion the dither frequency and amplitude, ramp time, and maximum and minimum current can be adjusted to match the input signals required to move the spool valve.

A.16 Fitting the Valve Packs

The hydraulic valve packs for the Hydro-Lek arms were installed onto one of the Brokk arm links close to the hydraulic integration point as shown in Figure A.16.

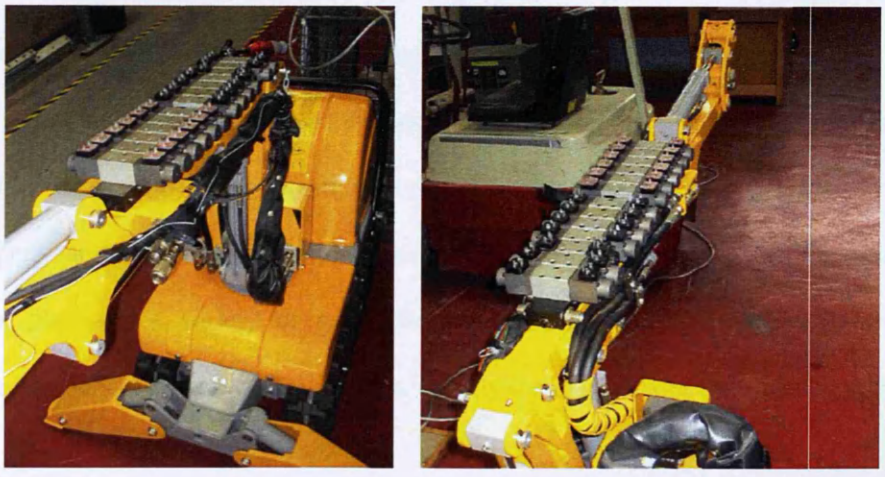


Figure A.16 Valve packs fitting

The specific location chosen was the Brokk manipulator link closest to the machine base and to the connection points, to allow the Brokk machine to act as a stabiliser when the link with the attached valve packs was moved up and down. As stated earlier in this chapter, the valve packs used in this research were nearly 20 years old and it was necessary to undertake maintenance and install new proportional P02 amplifiers. In addition the coil inside the solenoid valves were tested to make sure that they did not stick during the operation. During the research process it was discovered that some of the fitting points for the solenoid valves leaked and the valve packs heated up when they were operated for a long time.

APPENDIX B

COLLISON AVOIDANCE ALGORITHM

B.1 Collision detection

'Collision Detection' is about detecting colliding, or potential colliding, between the manipulator links and obstacles, or between the manipulator links themselves. The idea is to find the minimum distances between the corresponding objects. There are various methods of detection. For example sensor detecting or measuring may use laser sensors, ultrasonic, sonar, or bumpers; and object visualisation uses geometrical calculations. The literature in this appendix discusses a variety of techniques to calculate the minimum distance between visualised models of corresponding objects, namely the manipulator links or obstacles.

B.1.1 Basic minimum distance functions

The simplest way to model the links of manipulators or obstacles is to represent the objects as points and lines. The minimum distance can be obtained by calculating the minimum distance between point and point, point and line, or line and line. In this appendix the minimum distance calculations are represented in three-dimensional space (Weisstein, E. W. 2007).

B.1.1.1 The distance between point and point

The calculation of the distance between two points is represents the easiest distance calculation between two objects in three-dimensional space. Figure B.1 shows two points

$P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ in three-dimensional space; d is the distance between the two points.

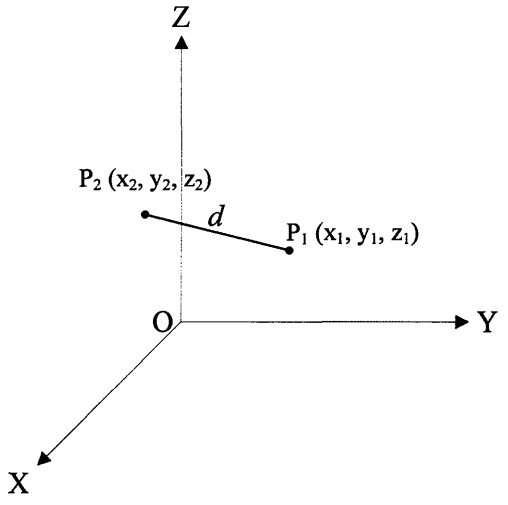


Figure B.1 Distance between two points

In order to represent the distance d, the points P_1 and P_2 are represented in vector form as:

$$P_1 = [x_1, y_1, z_1]^T = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad \text{(B1)}$$

$$P_2 = [x_2, y_2, z_2]^T = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad \text{(B2)}$$

Therefore, the distance, d, can be represented as the magnitude of vector $\mathbf{P_1P_2}$.

$$d = |P_1P_2| = |P_1 - P_2| \quad \text{(B3)}$$

Where

$$P_1P_2 = P_1 - P_2 = \begin{bmatrix} (x_1 - x_2) \\ (y_1 - y_2) \\ (z_1 - z_2) \end{bmatrix} \quad \text{(B4)}$$

Therefore, the distance, d, is:

$$d = |\mathbf{P_1} - \mathbf{P_2}| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad \text{(B5)}$$

B.1.1.2 The distance between point and line segment

The shortest distance between a point and a line is the distance of the normal from the point. The shortest distance between a point and a line segment can be different.

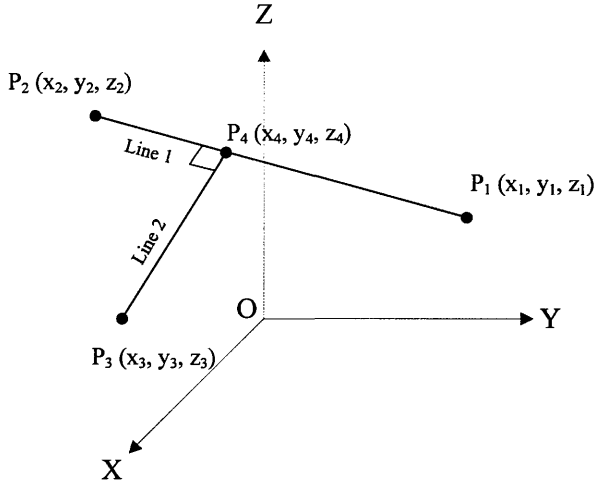


Figure B.2 Distance between a point and line segment

Figure B.2 shows Line 2 is the normal of Line 1 from $P_3(x_3, y_3, z_3)$. The distance of Line 2 is the shortest distance between point $P_3(x_3, y_3, z_3)$ and Line 1. If $P_4(x_4, y_4, z_4)$ is on segment P_1P_2, then the shortest distance is the distance of Line 2, otherwise, the shortest distance is the distance between $P_3(x_3, y_3, z_3)$ and one of the end points of the segment P_1P_2. The key to determining the minimum distance between a point and a segment therefore is to determine the coordinates of $P_4(x_4, y_4, z_4)$.

Any point on a line can be represented as:

$$\text{Point} + \text{Vector} \times \text{Scalar} \quad \text{(B6)}$$

Because $P_4(x_4, y_4, z_4)$ is on both Line 1 and Line 2 the point $P_4(x_4, y_4, z_4)$ can be represented as:

$$P_4 = P_1 + (P_2 - P_1)\, a \tag{B7}$$

$$P_4 = P_3 + (P_4 - P_3)\, b \tag{B8}$$

These two equations can be rewritten as:

$$P_4 = \begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} a = \begin{bmatrix} x_1 + (x_2 - x_1)a \\ y_1 + (y_2 - y_1)a \\ z_1 + (z_2 - z_1)a \end{bmatrix} \tag{B9}$$

$$P_4 = \begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} + \begin{bmatrix} (x_4 - x_3) \\ (y_4 - y_3) \\ (z_4 - z_3) \end{bmatrix} b = \begin{bmatrix} x_3 + (x_4 - x_3)b \\ y_3 + (y_4 - y_3)b \\ z_3 + (z_4 - z_3)b \end{bmatrix} \tag{B10}$$

Since Line 1 and Line 2 are perpendicular to each other,

$$(P_2 - P_1) \bullet (P_4 - P_3) = 0 \tag{B11}$$

This equation can be rewritten as:

$$\begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} \bullet \begin{bmatrix} (x_4 - x_3) \\ (y_4 - y_3) \\ (z_4 - z_3) \end{bmatrix} = \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} \bullet \begin{bmatrix} x_1 + (x_2 - x_1)a - x_3 \\ y_1 + (y_2 - y_1)a - y_3 \\ z_1 + (z_2 - z_1)a - z_3 \end{bmatrix} = 0 \tag{B12}$$

Solve the above equation for a,

$$a = \frac{-(x_2 - x_1)(x_1 - x_3) - (y_2 - y_1)(y_1 - y_3) - (z_2 - z_1)(z_1 - z_3)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{B13}$$

The coordinates of $P_4(x_4, y_4, z_4)$ can be determined once the scalar a' is available.

$$a' = \begin{Bmatrix} 1, \text{if } a > 1 \\ 0, \text{if } a < 0 \\ a, \text{if } 0 \le a \le 1 \end{Bmatrix} \tag{B14}$$

and

$$P_4 = \begin{bmatrix} x_1 + (x_2 - x_1)a' \\ y_1 + (y_2 - y_1)a' \\ z_1 + (z_2 - z_1)a' \end{bmatrix} \tag{B15}$$

The problem has thus become the calculation of the distance between point $P_3(x_3, y_3, z_3)$ and point $P_4(x_4, y_4, z_4)$. The minimum distance, d, between the point $P_3(x_3, y_3, z_3)$ and the segment P_1P_2 is therefore:

$$d = |P_3 - P_4| = \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2 + (z_3 - z_4)^2} \tag{B16}$$

B.1.1.3 The distance between two line segments

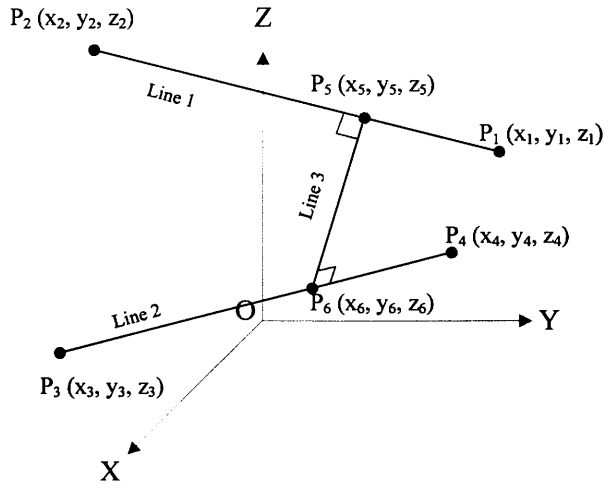


Figure B.3 Distance between two line segments

Figure B.3 shows two line segments P_1P_2 and P_3P_4. In order to represent the shortest distance between Line 1 and Line 2, Line 3 has to be perpendicular to both of them. In order to calculate the length of Line 3, the coordinates of point $P_5(x_5, y_5, z_5)$ and point $P_6(x_6, y_6, z_6)$

must be determined. Because P_1P_2 and P_3P_4 are two segments, if $P_5(x_5, y_5, z_5)$ or $P_6(x_6, y_6, z_6)$ are on the two segments, then the minimum distance of the two segments is the length of Line 3. Otherwise the minimum distance of the two segments is not the length of Line 3. The solution of this will be discussed at the end of this section.

To determine the distance between $P_5(x_5, y_5, z_5)$ and $P_6(x_6, y_6, z_6)$ the same strategy is used as in the previous section. Because $P_5(x_5, y_5, z_5)$ is on Line1 and $P_6(x_6, y_6, z_6)$ is on Line 2, the point $P_5(x_5, y_5, z_5)$ and $P_6(x_6, y_6, z_6)$ can be represented as:

$$P_5 = P_1 + (P_2 - P_1)\, a \tag{B17}$$

$$P_6 = P_3 + (P_4 - P_3)\, b \tag{B18}$$

These two equations can be rewritten as:

$$P_5 = \begin{bmatrix} x_5 \\ y_5 \\ z_5 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} a = \begin{bmatrix} x_1 + (x_2 - x_1)a \\ y_1 + (y_2 - y_1)a \\ z_1 + (z_2 - z_1)a \end{bmatrix} \tag{B19}$$

$$P_6 = \begin{bmatrix} x_6 \\ y_6 \\ z_6 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} + \begin{bmatrix} (x_4 - x_3) \\ (y_4 - y_3) \\ (z_4 - z_3) \end{bmatrix} b = \begin{bmatrix} x_3 + (x_4 - x_3)b \\ y_3 + (y_4 - y_3)b \\ z_3 + (z_4 - z_3)b \end{bmatrix} \tag{B20}$$

Since Line 3 is perpendicular to both Line 1 and Line 2,

$$(P_2 - P_1) \bullet (P_6 - P_5) = 0 \tag{B21}$$

$$(P_4 - P_3) \bullet (P_6 - P_5) = 0 \tag{B22}$$

The equations can be rewritten as:

$$\begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} \bullet \begin{bmatrix} (x_6 - x_5) \\ (y_6 - y_5) \\ (z_6 - z_5) \end{bmatrix} = \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ (z_2 - z_1) \end{bmatrix} \bullet \begin{bmatrix} x_3 + (x_4 - x_3)b - x_1 - (x_2 - x_1)a \\ y_3 + (y_4 - y_3)b - y_1 - (y_2 - y_1)a \\ z_3 + (z_4 - z_3)b - z_1 - (z_2 - z_1)a \end{bmatrix} = 0 \tag{B23}$$

$$\begin{bmatrix}(x_4-x_3)\\(y_4-y_3)\\(z_4-z_3)\end{bmatrix}\bullet\begin{bmatrix}(x_6-x_5)\\(y_6-y_5)\\(z_6-z_5)\end{bmatrix}=\begin{bmatrix}(x_4-x_3)\\(y_4-y_3)\\(z_4-z_3)\end{bmatrix}\bullet\begin{bmatrix}x_3+(x_4-x_3)b-x_1-(x_2-x_1)a\\y_3+(y_4-y_3)b-y_1-(y_2-y_1)a\\z_3+(z_4-z_3)b-z_1-(z_2-z_1)a\end{bmatrix}=0 \quad \text{(B24)}$$

A new equation can be generated by manipulating the two equations above and collecting terms as given below:

$$\begin{bmatrix}-(x_2-x_1)^2-(y_2-y_1)^2-(z_2-z_1)^2 & (x_4-x_3)(x_2-x_1)+(y_4-y_3)(y_2-y_1)+(z_4-z_3)(z_2-z_1)\\-(x_4-x_3)(x_2-x_1)-(y_4-y_3)(y_2-y_1)-(z_4-z_3)(z_2-z_1) & (x_4-x_3)^2+(y_4-y_3)^2+(z_4-z_3)^2\end{bmatrix}\times\begin{bmatrix}a\\b\end{bmatrix}$$
=

$$\begin{bmatrix}(x_1-x_3)(x_2-x_1)+(y_1-y_3)(y_2-y_1)+(z_1-z_3)(z_2-z_1)\\(x_1-x_3)(x_4-x_3)+(y_1-y_3)(y_4-y_3)+(z_1-z_3)(z_4-z_3)\end{bmatrix} \quad \text{(B25)}$$

This can be solved for a and b.

In order to constrain the points $P_5(x_5, y_5, z_5)$ and $P_6(x_6, y_6, z_6)$ on the line segments, the scalars a' and b' must be defined as:

$$a'=\begin{Bmatrix}1, \text{if } a>1\\0, \text{if } a<0\\a, \text{if } 0\le a\le 1\end{Bmatrix} \quad \text{(B26)}$$

$$b'=\begin{Bmatrix}1, \text{if } b>1\\0, \text{if } b<0\\b, \text{if } 0\le b\le 1\end{Bmatrix} \quad \text{(B27)}$$

Then the coordinates of $P_5(x_5, y_5, z_5)$ and $P_6(x_6, y_6, z_6)$ can be determined:

$$P_5=\begin{bmatrix}x_5\\y_5\\z_5\end{bmatrix}=\begin{bmatrix}x_1+(x_2-x_1)a'\\y_1+(y_2-y_1)a'\\z_1+(z_2-z_1)a'\end{bmatrix} \quad \text{(B28)}$$

$$P_6=\begin{bmatrix}x_6\\y_6\\z_6\end{bmatrix}=\begin{bmatrix}x_3+(x_4-x_3)b'\\y_3+(y_4-y_3)b'\\z_3+(z_4-z_3)b'\end{bmatrix} \quad \text{(B29)}$$

The minimum distance, d, of two line segments can then be calculated as:

$$d = |P_6 - P_5| = \sqrt{(x_6 - x_5)^2 + (y_6 - y_5)^2 + (z_6 - z_5)^2} \quad \text{(B30)}$$

There are however some exceptions for the calculation. If $a' \neq a$ or $b' \neq b$ or $a' \neq a$ and $b' \neq b$, then the calculated minimum distance may be incorrect. Figure B.4 gives an example in which the point P_6' is outside of the segment P_3P_4. That means:

$$b > 1, b' = 1$$

$$P_6 = P_4$$

$$d = |P_4 - P_5|$$

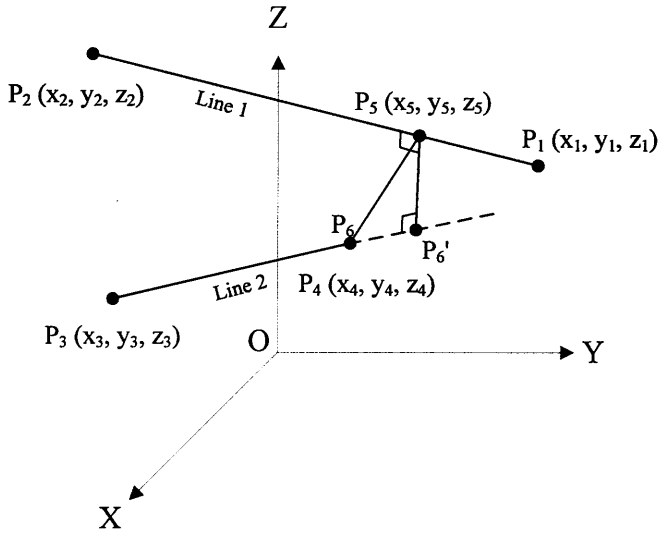


Figure B.4 A special situation for distance calculation between two line segments

d may not be the real minimum distance, however, between the two segments. The true minimum distance should be the shortest distance from each line segment endpoint to the corresponding segment.

B.2 Introduction to kinematics control for a robot manipulator

The configuration of a robot manipulator can be specified by the joints of the manipulator. The way in which the joint angles are used to specify the position and orientation of the end-

effector is called the forward kinematics. If the position and orientation of the end-effector is represented by vector $\boldsymbol{x}$ and the joint angles by vector $\boldsymbol{\theta}$, then the relation between $\boldsymbol{x}$ and $\boldsymbol{\theta}$ can be represented in the following equation (L. Zlajpah and B. Nemec 2002):

$$x = f(\theta) \tag{B31}$$

where $\boldsymbol{f}$ is the function of the manipulator forward kinematics.

There is also an equation that describes the relation between the end-effector velocity and joint velocity. The end-effector velocity is represented as a six-dimensional vector $\dot{\boldsymbol{x}}$ and the joint angular velocities are represented as n-dimensional vector $\dot{\boldsymbol{\theta}}$. Where n is the number of DOF of the manipulator. The equation is denoted as:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{B32}$$

where $\boldsymbol{J}$ is the Jacobian matrix (Paul. R., 1981).

Due to the redundancy of the manipulator the inverse of matrix $\boldsymbol{J}$ cannot be defined in this case. The joint angular velocities $\dot{\boldsymbol{\theta}}$ however, can be solved by using the pseudo-inverse (E. H. Moore 1968), which gives the best approximate solution. The pseudo-inverse of Jacobian matrix $\boldsymbol{J}$ is represented as $\boldsymbol{J}^{+}$. The generalized solution of $\dot{\boldsymbol{\theta}}$, which is solved from the equation B32, is described as equation B33 (L. Zlajpah and B. Nemec 2002):

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} + \boldsymbol{N}\boldsymbol{z} \tag{B33}$$

Where $\boldsymbol{z}$ is an arbitrary vector in $\dot{\boldsymbol{\theta}}$–space and $\boldsymbol{N}$ is a projection operator matrix which representing the projection onto the null space of $\boldsymbol{J}$, $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J}$, where $\boldsymbol{I}$ is an $n \times n$ identity matrix.

B.3 Collision avoidance strategy

A manipulator is described as redundant when it can still move its links in some directions without disturbing the aim of the end-effector to achieve its primary goal. This property of the redundant manipulators can be used for collision avoidance. The strategy of collision avoidance is to identify the point on the manipulator which is closest to an obstacle, denoted as *collision avoidance point* A_o, and then assign to it a motion component that moves the point directly away from the obstacle as shown in Figure B.5 below:

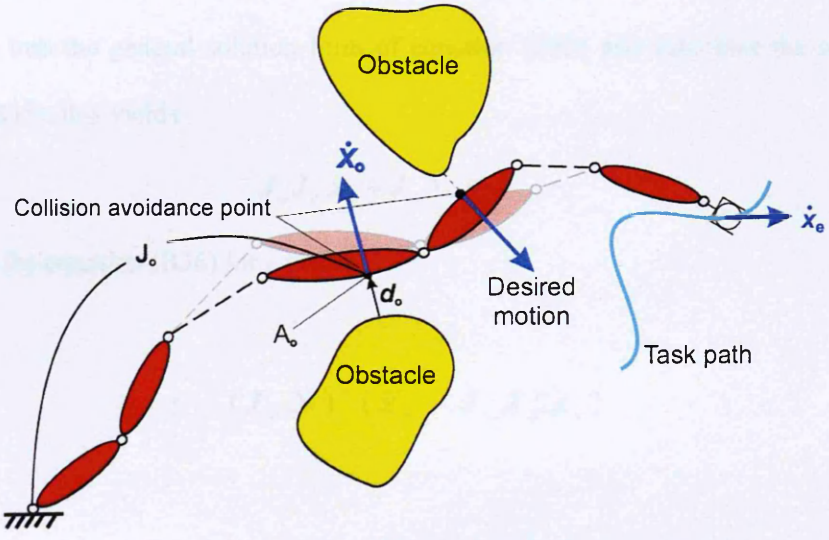


Figure B.5 Obstacles avoidance (L. Zlajpah and B. Nemec 2002)

In the case of dual-arm manipulators the minimum distances d_o between each link of the two arms can be calculated using virtual models which have the coordination information for all links of both arms. The direction of the collision avoidance point can also be obtained in the same way. The end-effector moves with a desired velocity in its specified trajectory, hence the primary goal of the specified end-effector motion and the secondary goal of collision avoidance can be described by the following equations:

$$\boldsymbol{J}_e\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{x}}_e \tag{B34}$$

$$\boldsymbol{J}_o\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{x}}_o \tag{B35}$$

Where

$\boldsymbol{J}_e$: Jacobian matrix of the end-effector

$\boldsymbol{J}_o$: Jacobian matrix of the collision avoidance point

$\dot{\boldsymbol{x}}_e$: Velocities vector of the end-effector

$\dot{\boldsymbol{x}}_o$: Velocity vector of the collision avoidance point

B.3.1 The general solution for collision avoidance

To find a common solution of both equations (B34) and (B35), it is necessary to transfer the equation (B34) into the general solution form of equation (B33) and substitute the solution into equation (B35); this yields:

$$\boldsymbol{J}_o \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e + \boldsymbol{J}_o \boldsymbol{N} \boldsymbol{z} = \dot{\boldsymbol{x}}_o \tag{B36}$$

The solution to the equation (B36) for z yields

$$\boldsymbol{z} = (\boldsymbol{J}_o \boldsymbol{N})^{+} (\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e) \tag{B37}$$

When z is substituted back into the equation (B33) it gives the final solution for $\dot{\boldsymbol{\theta}}$, which is the desired solution to achieve the two goals of the specified end-effector motion and the collision avoidance. The final solution for $\dot{\boldsymbol{\theta}}$ is described as:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e + \boldsymbol{N} (\boldsymbol{J}_o \boldsymbol{N})^{+} (\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e) \tag{B38}$$

Since the projection operator $\boldsymbol{N}$ is hermitian and also idempotent, the solution (B38) can be simplified to

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e + (\boldsymbol{J}_o \boldsymbol{N})^{+} (\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o \boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}_e) \tag{B39}$$

Each of the terms from the equation (B39) can be easily explained. When the manipulator has sufficient redundancy the term $\boldsymbol{J}_e^{+} \dot{\boldsymbol{x}}$ acts as a role which guarantees that the end-effector will

move in exactly the desired velocity with the minimum joint velocity norm. The term $(\boldsymbol{J}_o\boldsymbol{N})^+(\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o\boldsymbol{J}_e^+\dot{\boldsymbol{x}}_e)$ is a homogeneous solution which allows the manipulator to use the rest of the redundancy to satisfy the different goals. In this case it is used to achieve collision avoidance. The term $\boldsymbol{J}_o\boldsymbol{N}$ represents the degree of redundancy available to move the collision avoidance point without affecting the motion of the end-effector. Its pseudo-inverse $(\boldsymbol{J}_o\boldsymbol{N})^+$ maps the Cartesian velocity of point A_o into the joint velocity, and the term $(\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o\boldsymbol{J}_e^+\dot{\boldsymbol{x}}_e)$ is the vector which describes the desired motion of the collision avoidance point A_o. It is composed of the specified collision avoidance point velocity $\dot{\boldsymbol{x}}_o$ subtracted by a vector which represents the motion at A_o caused by the motion of end-effector.

B.3.2 Distance influence control

The collision avoidance efficiency of this algorithm relies on the volume of the influence of the minimum distance between the manipulator and the obstacle. Hence, to efficiently control the collision avoidance, the equation (B39) can be modified with some gain parameters as:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^+\dot{\boldsymbol{x}}_e + \alpha_h(\boldsymbol{J}_o\boldsymbol{N})^+(\alpha_o\boldsymbol{v}_o - \boldsymbol{J}_o\boldsymbol{J}_e^+\dot{\boldsymbol{x}}_e) \tag{B40}$$

where α_o is the collision avoidance point velocity gain. Here $\boldsymbol{v}_o$ is termed as a unit vector which represents the direction of the collision avoidance point velocity. Therefore the following equation can be used:

$$\dot{\boldsymbol{x}}_o = \alpha_o\boldsymbol{v}_o \tag{B41}$$

α_h is the gain of the homogenous solution. It indicates the volume of the homogenous solution that will be included in the whole solution. Figure B.6, given below, illustrates how the control gains α_o and α_h work when they are considered as functions versus the minimum distance.

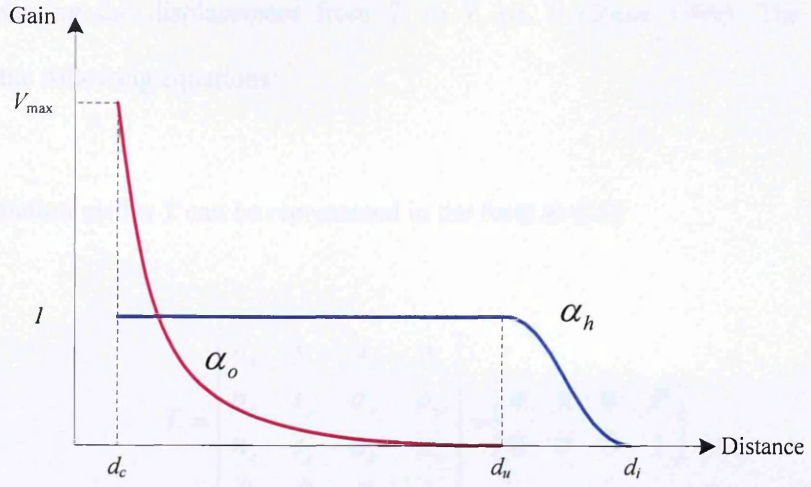


Figure B.6 Control Gains α_o and α_h Versus Minimum Distance

(Maciejewski and Klein 1985)

Figure B.6 indicates three critical distances, d_c, d_u and d_i, where d_c is the system abort distance. This is used if the manipulator is too close to the obstacle, then the task will be aborted. d_u is the unit gain distance where the homogenous gain α_h becomes '1'. This means that the complete homogenous solution is included and the velocity gain α_o starts to influence the avoidance velocity inversely against the minimum distance. d_i is the influence distance when the collision avoidance point approaches d_u from d_i. The homogenous solution is partially included in the total solution, which gives an even velocity of the collision avoidance point because of the motion of the end-effector.

B.4 Determination of the end-effector velocity

Since the end-effector trajectory is specified, the end-effector velocity $\dot{\boldsymbol{x}}_e$ can be defined as the difference between the end-effector's current position and the target position in every period of time. Therefore, the end-effector velocity can be described as:

$$\dot{\boldsymbol{x}}_e = tr2diff(\boldsymbol{T}_c, \boldsymbol{T}_t) \tag{B42}$$

where T_c is the homogenous transformation matrix for the current position of the end-effector. T_t stands for the target position and *tr2diff* is the function that returns a differential motion

vector representing the displacement from T_t to T_c (P. I. Corke 1996). The solution is described in the following equations:

The transformation matrix T can be represented in the form as [13]

$$\boldsymbol{T} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B43}$$

Hence,

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_t - \boldsymbol{P}_c \\ 1/2(\boldsymbol{n}_c \times \boldsymbol{n}_t + \boldsymbol{o}_c \times \boldsymbol{o}_t + \boldsymbol{a}_c \times \boldsymbol{a}_t) \end{bmatrix} \tag{B44}$$

Therefore the end-effector velocity $\dot{\boldsymbol{x}}_e$ is obtained.

B.5 Determination of the collision avoidance velocity

From equation (B41) it can be seen that the avoidance velocity $\dot{\boldsymbol{x}}_o$ can be determined by the velocity gain α_o and the unit vector $\boldsymbol{v}_o$; where α_o scales to adjust the volume and the unit vector $\boldsymbol{v}_o$ gives the direction. Here the velocity gain α_o is defined as a function relative to the minimum distance d_m, as shown below in equation (B45):

$$\alpha_o = \begin{cases} \left(\dfrac{d_u}{d_m}\right)^2 - 1 & d_m < d_u \\ 0 & d_m \geq d_u \end{cases} \tag{B45}$$

The curve of the relation between α_o and d_m is illustrated in Figure B.7 below.

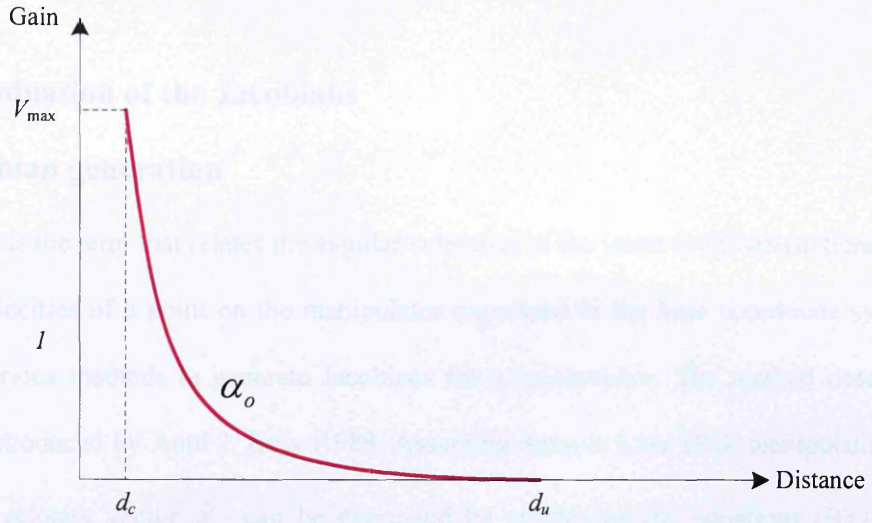
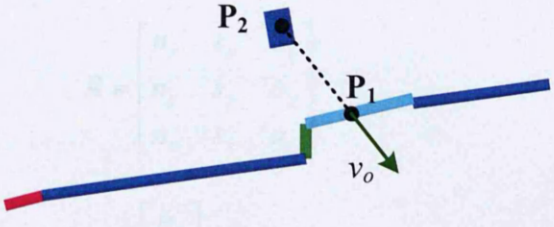


Figure B.7 The Curve of α_o Relative to the minimum distance d_m

The direction unit vector $\boldsymbol{v}_o$ is calculated by the virtual manipulators modelling module. If we know the position of the minimum distance points on both the manipulator and the obstacle, then the direction vector $\boldsymbol{v}_o$ can be calculated. For example there are two points on the manipulator P_1 and on the obstacle P_2, as shown in Figure B.8 below, where P_1 and P_2 is the position vector formed as $[x, y, z]^T$. Hence the direction of the unit vector which point the direction from P_2 to P_1 can be described as:

$$\boldsymbol{v}_o = \frac{\mathbf{P}_2 - \mathbf{P}_1}{\|\mathbf{P}_2 - \mathbf{P}_1\|} \tag{B46}$$



Slave Manipulator

Figure B.8 Slave arm with obstacle

The position of $\mathbf{P}_1$ and $\mathbf{P}_2$ can be obtained from the virtual modelling module which is discussed in section B6 below.

B.6 Determination of the Jacobians

B.6.1 Jacobian generation

The Jacobian is the term that relates the angular velocities of the joints to the translational and rotational velocities of a point on the manipulator expressed in the base coordinate system. There are various methods to generate Jacobians for a manipulator. The method described below was introduced by Antti J. Koivo1989. Assuming there is a six DOF manipulator, the end-effector velocity vector $\dot{\boldsymbol{x}}_e$ can be expressed by combining the equations (B34) and (B44) as:

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \boldsymbol{J}_e \dot{\boldsymbol{\theta}} \tag{B47}$$

The transformation matrix for each joint is represented as:

$$\boldsymbol{T} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B48}$$

The rotation matrix and position vector are represented as:

$$\boldsymbol{R} = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \tag{B49}$$

$$\boldsymbol{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{B50}$$

Hence, for all the joints (1-6) we have:

$$\boldsymbol{T}_0^1, \boldsymbol{T}_0^2, \boldsymbol{T}_0^3, \boldsymbol{T}_0^4, \boldsymbol{T}_0^5, \boldsymbol{T}_0^6 \tag{B51}$$

$$\boldsymbol{R}_0^1, \boldsymbol{R}_0^2, \boldsymbol{R}_0^3, \boldsymbol{R}_0^4, \boldsymbol{R}_0^5, \boldsymbol{R}_0^6 \tag{B52}$$

$$\boldsymbol{P}_0^1, \boldsymbol{P}_0^2, \boldsymbol{P}_0^3, \boldsymbol{P}_0^4, \boldsymbol{P}_0^5, \boldsymbol{P}_0^6 \tag{B53}$$

There is also a term $\boldsymbol{k}_z$ which is the unit vector representing the positive direction of z axis:

$$\boldsymbol{k}_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{B54}$$

For i = (0, 1, 2, 3, 4, 5) we therefore have:

$$\boldsymbol{k}_{zi} = \boldsymbol{R}_0^i \boldsymbol{k}_z, \tag{B55}$$

$$\boldsymbol{P}_i^6 = \boldsymbol{P}_0^6 - \boldsymbol{P}_0^i, \tag{B56}$$

$$\boldsymbol{J}_{zi} = \boldsymbol{k}_{zi} \times \boldsymbol{P}_i^6, \tag{B57}$$

And:

$$\boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{J}_{zi} \\ \boldsymbol{k}_{zi} \end{bmatrix} \tag{B58}$$

The Jacobian can therefore be generated in the following form as:

$$\boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{k}_{z0} \times \boldsymbol{P}_0^6 \ldots \boldsymbol{k}_{zi} \times \boldsymbol{P}_i^6 \ldots \boldsymbol{k}_{z5} \times \boldsymbol{P}_5^6 \\ \boldsymbol{k}_{z0} \quad \ldots \quad \boldsymbol{k}_{zi} \quad \ldots \quad \boldsymbol{k}_{z5} \end{bmatrix} \tag{B59}$$

Where $\boldsymbol{k}_{z0} = 0$ and $\boldsymbol{P}_0^0 = 0$, since the base coordinate system does not move.

B.6.2 End-effector Jacobian

In the case of the slave manipulator the joints configuration is known, this is represented as $\boldsymbol{\theta}$.

By applying the forward kinematics the following equation is formed:

$$\mathbf{T} = fkin(\boldsymbol{\theta}) \tag{B60}$$

where *fkin* is the forward kinematics function; T stores all the transformation matrices as (B51) and other matrices. The transformation matrices can thus be obtained from the function *fkin*. It is then necessary to apply the Jacobian generation method for the end-effector transformation matrix $\boldsymbol{T}_0^e$ which is represented below (B61) as:

$$\boldsymbol{J}_e = jacobian\_e(\boldsymbol{T}_0^e) \tag{B61}$$

where *jacobian_e* is the end-effector Jacobian generation function. The end-effector Jacobian $\boldsymbol{J}_e$ can then be determined.

For full details of the function *fkin* and the end-effector Jacobian generation function, please refer to the MATLAB code for Forward Kinematics Function at the end of this appendix.

B.6.3 Jacobian for collision avoidance point

The difference between the Jacobian generation for the end-effector and the collision avoidance point is that the Jacobians are for different locations of the manipulator. To generate the Jacobian for the collision avoidance point therefore, it is important to obtain the location of that point. The virtual manipulator modelling module provides the functionality which indicates the coordination of the collision avoidance point in a position vector as shown in Figure B10.

In Figure B.9 below $\boldsymbol{A}_o$ is the collision avoidance point on 'Link *i*', its coordinates are obtained through the virtual modelling module and are represented as $[x, y, z]^T$ which is described with respect to frame {i}. To get the Jacobian for point $\boldsymbol{A}_o$, however, it is necessary to obtain the transformation matrix $\boldsymbol{T}_0^a$ of $\boldsymbol{A}_o$, which is relative to the base frame.

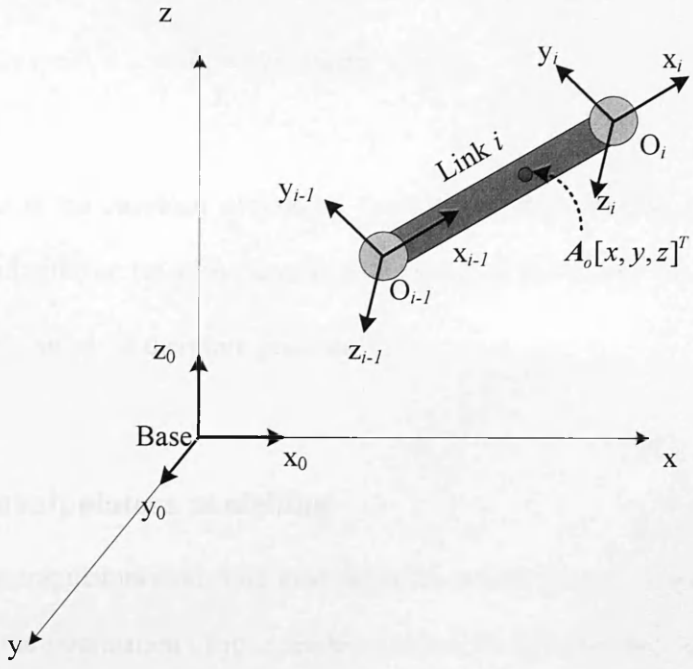


Figure B.9 Jacobian for Collision Avoidance Point A_o

Here we have the coordinates of A_o:

$$\boldsymbol{P}_i^a = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{B62}$$

The transformation matrix of A_o with respect to frame {i}:

$$\boldsymbol{T}_i^a = \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & \boldsymbol{P}_i^q \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B63}$$

Therefore, $\boldsymbol{T}_0^a$ can be obtained as:

$$\boldsymbol{T}_0^a = \boldsymbol{T}_0^i \boldsymbol{T}_i^a \tag{B64}$$

Applying the Jacobian generation method on this transformation matrix:

$$\boldsymbol{J}_o = jacobian\_o(\boldsymbol{T}_0^a) \tag{B65}$$

Where *jacobian_o* is the Jacobian generation function for the collision avoidance points (please see the End-effector Jacobian Generation Function at the end of this appendix). The Jacobian matrix of point $\boldsymbol{A}_o$ is therefore generated.

B.6.4 Virtual manipulators modelling

There is a virtual manipulators modelling module which models the two manipulators in real-time according to the information of joint angles obtained from the system. It is necessary to provide the minimum distances, the location of the collision avoidance point, and to indicate the direction of avoidance velocity. Figure B.10 is an illustration of the virtual manipulators modelling module.

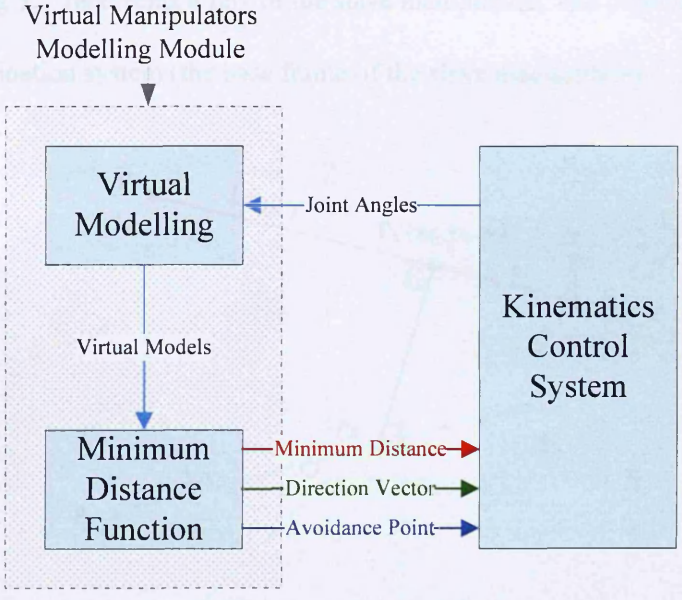


Figure B.10 Virtual manipulators modelling module

This modelling system represents the links of the two manipulators as line segments. The line segments can be represented by two points which are the two end-points. The minimum

distance between the segments can be calculated. For example if there are two line segments they can be represented as $\mathbf{L}_1$ and $\mathbf{L}_2$ as follows:

$$\mathbf{L}_1 = segment(\boldsymbol{P}_1, \boldsymbol{P}_2) \tag{B66}$$

$$\mathbf{L}_2 = segment(\boldsymbol{P}_3, \boldsymbol{P}_4) \tag{B67}$$

A segment is the function that constructs a segment object with two specified end-points. $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are the two end-points of line segment $\mathbf{L}_1$; and $\boldsymbol{P}_3$ and $\boldsymbol{P}_4$ are the two end-points of line segment $\mathbf{L}_2$. The minimum distance can thus be calculated by a distance function as:

$$d_{\min} = distance(\mathbf{L}_1, \mathbf{L}_2) \tag{B68}$$

In Figure B.11, given below, it can be seen that Link *i*, then $\boldsymbol{P}_5$ will be the collision avoidance points, assuming $\mathbf{L}_1$ represents a link of the slave manipulator. The coordinates are relative to the world coordination system (the base frame of the slave manipulator).

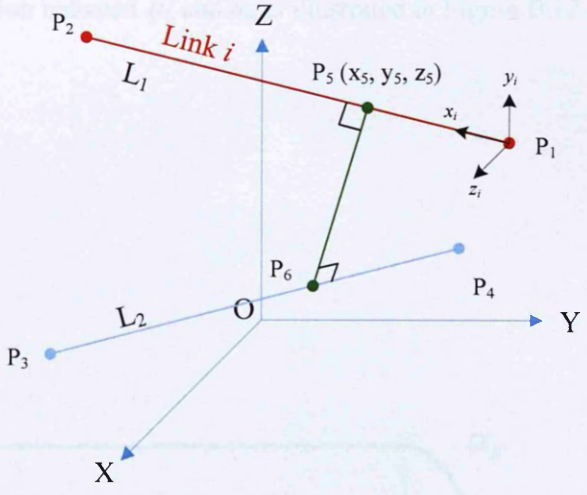


Figure B.11 Distance between two line segments

In order to get the position that relative to the frame $\{i\}$, the position vector is transformed as:

$$\boldsymbol{T}_i^{p5} = (\boldsymbol{T}_0^i)^{-1}\boldsymbol{T}_0^{p5} = \boldsymbol{T}_i^0\boldsymbol{T}_0^{p5} \tag{B69}$$

The new coordinates can thus be obtained according to the equations (B62) and (B63).

This virtual modelling module can also determine the direction of avoidance velocity. For example in Figure B.11 the direction would be from $\boldsymbol{P}_6$ to $\boldsymbol{P}_5$.

B.6.5 Algorithm flow control

We will now define α_h and then put this together with the other variables defined above in order to complete the kinematics control. With regards to α_h, the homogenous solution gain indicates the volume of the homogenous solution that will be included in the whole solution. In this case it has been selected as:

$$\alpha_h = \begin{cases} 1 & d_m \le d_u \\ \frac{1}{2}(1+\cos(\pi \frac{d_m - d_u}{d_i - d_u})) & d_u < d_m < d_i \\ 0 & d_i \le d_m \end{cases} \quad \text{(B70)}$$

The curve of the relation between α_h and d_m is illustrated in Figure B.12 below:

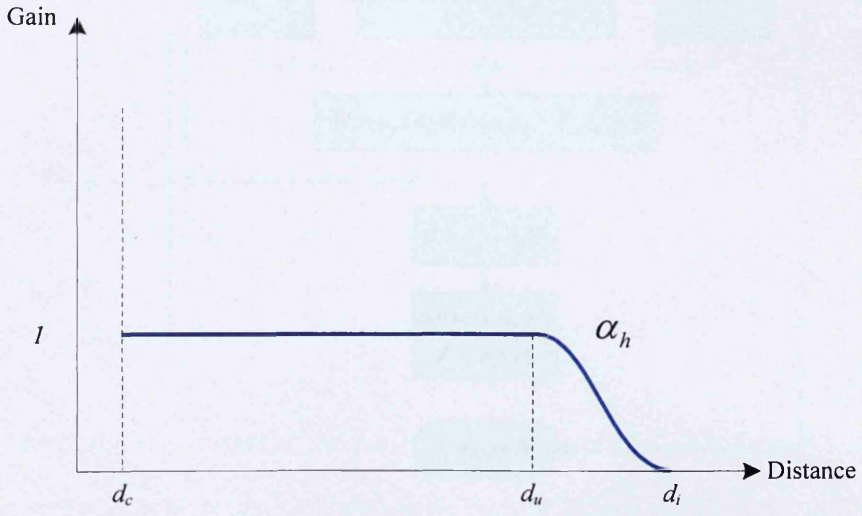


Figure B.12 The Curve of α_h Relative to Minimum Distance d_m

After all the variables are defined the kinematics control can be implemented as a whole, as illustrated in Figure B.13 below:

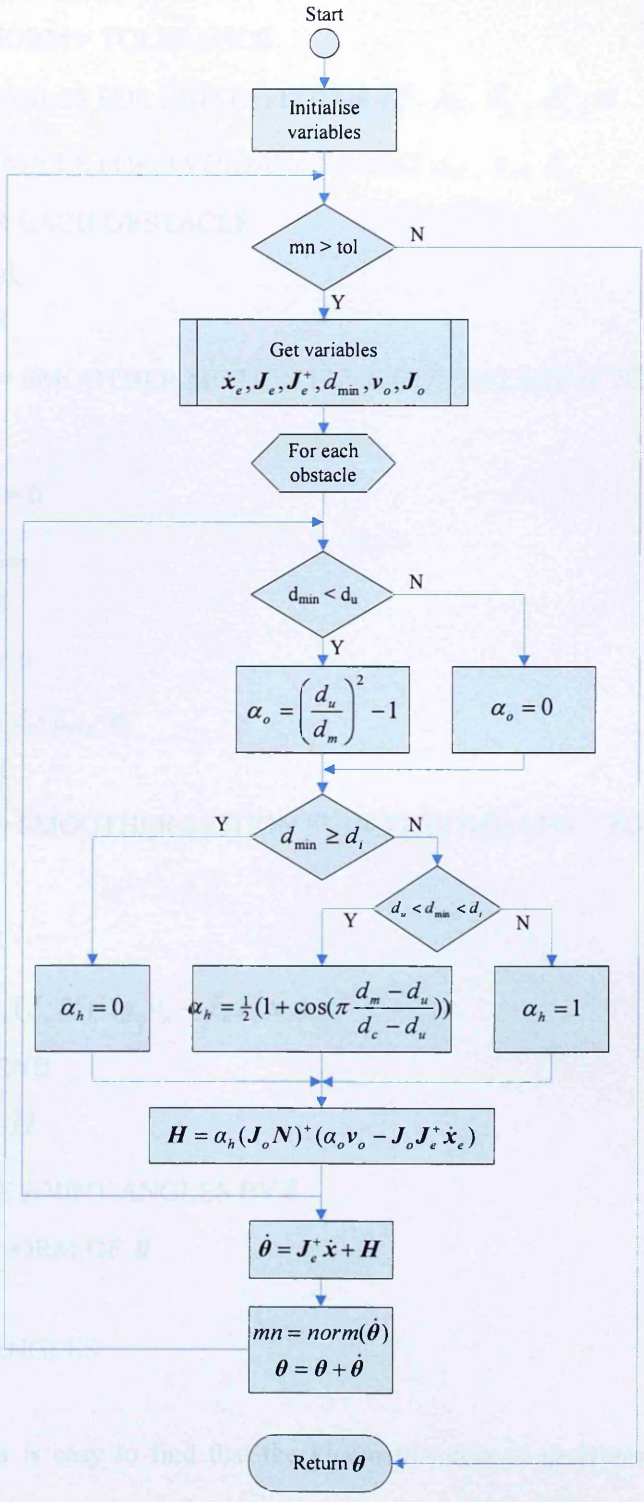


Figure B.13 Kinematics Control Flow Chart Diagram

The algorithm can be described in pseudo-code as given below:

KCONTROL

INITIALISE VARIABLES

WHILE MININORM > TOLERANCE

GET VARIABLES FOR END-EFFECTOR $\boldsymbol{T}_0^e$, $\dot{\boldsymbol{x}}_e$, $\boldsymbol{J}_e$, $\boldsymbol{J}_e^+$, $\boldsymbol{N}$

GET VARIABLES FOR AVOIDANCE POINT $d_{\min}$, $\boldsymbol{v}_o$, $\boldsymbol{J}_o$

LOOP FOR EACH OBSTACLE

IF $d_{min} < d_u$

THEN

α_o = SMOOTHER MOTION FUNCTION RELATIVE TO d_{min}

ELSE

$\alpha_o = 0$

IF $d_i \leq d_{min}$

THEN

$\alpha_h = 0$

ELSEIF $d_c < d_{min} < d_i$

THEN

α_h = SMOOTHER MOTION FUNCTION RELATIVE TO d_{min}

ELSE

$\alpha_h = 1$

H += $\alpha_h(\boldsymbol{J}_o\boldsymbol{N})^+(\alpha_o\boldsymbol{v}_o - \boldsymbol{J}_o\boldsymbol{J}_e^+\dot{\boldsymbol{x}}_e)$

LOOPEND

$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^+\dot{\boldsymbol{x}} + \boldsymbol{H}$

INCREACE JOIINT ANGLES BY $\dot{\boldsymbol{\theta}}$

GET MININORM OF $\dot{\boldsymbol{\theta}}$

WHILEEND

RETURN JOINT ANGLES

From the pseudo-code it is easy to find that the kinematics control algorithm controls the manipulator at the velocity level. It minimises the norm of joint velocity for the best solution and returns the joint angles back to the system.

CD CONTENTS

Robot Manipulator Motion Simulation MATLAB Scripts

Control Panel Function

Kinematics Control Function

Minimum Distance Function

Primary Distance Function

Animated Function

Draw Link Function

Minimum Distance Plot Function

End-Effector Cartesian Space Trajectory Generation Function

D-H Parameters Loading Function

Jacobian Generation Function

End-Effector Jacobian Generation Function

Robot Initialization Function

Forward Kinematics Function

Inverse Kinematics Function

Segment Class

APPENDIX C

ROBOT ARMS COORDINATION AND COLLISION AVOIDANCE

C.1 Introduction

In the nuclear industry, successful autonomous completion of complex, hazardous tasks by a robotic system currently relies predominantly on human input for control. The unpredictability and density of information provided by the environment surrounding a robot, combined with inaccuracies in sensor measurements make most tasks difficult for a robotic system to complete autonomously in this environment. In addition, situations may arise where the control of robotic systems becomes very complex to the point that controlling all the degrees of freedom synchronously in order to complete a task may prove difficult for the operator to direct. An example of this is Honda's ASIMO humanoid robot (Honda 2008) which requires the operator to be constantly aware of and control 26 degrees of freedom simultaneously, in order to guide the robot through tasks and avoid damage to itself or its surrounding environment. When a task requires two robot manipulators to work in a common workspace the potential for collision exists between the two moving arms. Many tasks therefore require some kind of motion coordination between the two arms for efficient operation, by planning a set of collision-free paths; for example, the motion time of the arms can be minimised.

In recent years, the cooperative motion of two robot manipulators has become an important area of research. Most of these studies consider the task to grasp or move a common object (T. Tsuji *et al.* 1997). In many cases the two-robot manipulators system has redundant DOF, but the use of a system with redundant DOF creates problems with regards to control (Hayashi *et al.* 1990; Hirose, S. and Ma, S. 1991). These problems include optimal force distribution for multiple manipulators (C. R. Carignan and D. L. Akin 1989; and F. T. Cheng and D. E. Orin 1990); and trajectory planning to move a common object along a specified path (J. E. Bobrow *et al.* 1990; and S. B. Moon and S. Ahmed 1990). The adoption of multi-arm robot systems instead of single-arm robot systems is vital for tasks such as parts mating and safe transportation of heavy or large objects. Multi-arm cooperation allows parts to be assembled and manipulated without the aid of fixtures or jigs. It is important to note that the research of multiple co-operative arms for robotic systems has a close relationship with research on dexterous robotic hands, which concerns the safe and robust grasping of an object and the dexterous manipulation of a grasped object with multiple articulated fingers (J. K. Salisbury and B. Roth 1983; J. K. Salisbury and J. J. Craig 1982; H. Kobayashi 1985).

When the dimension of joint space is greater than that of the operational space required for a particular task, arm redundancy occurs. This allows the optimal selection of joint configurations for the avoidance of singularities and collisions; balancing joint loads; and minimising required energy or time. An integral part of the development of multi-redundant-arm co-operative task execution is the automation of motion planning and control. Motion planning is concerned with the determination of Cartesian position and force trajectories of individual arms under the kinematic and kinaesthetic constraints imposed by multi-arm cooperation; and the selection of the optimal trajectory of joint configurations. Motion control is concerned with the control of kinematic and dynamic interactions between the two arms to accomplish the planned position and force trajectories. Approaches to multi-arm control include the following:

1. Master-slave control. In this approach one arm controls motions (the master) while the other arm kinematically follows and is responsible for complying with the interactive force (the slave) (T. Ishida 1977). The kinematics of the multi-arm robot system is controlled in such a way as to keep the slave arm in a symmetrical relationship to the master arm during point-to-point motion (A. Hemami 1986).
2. Object-centred control. In this approach the desired object position and force trajectories are transformed to become the control goals of the individual arms (J. H. Lim and D. H. Chung 1985; M. Uchiyama *et al.* 1987; P. Dauchez 1986; H. West and H. Asada 1985). The control of the incremental movement of the multi-arm robot system when it grasps a rigid object is based on the differential kinematic relation between the grasped object and the two robot arms (J. H. Lim and D. H. Chyung 1985).
3. Force control. This is applied to the coordination of two arms engaged in parallel and rotational object transfer. The difficulty of compensating the interactive force between the two cooperative arms is discussed by T. Ishida 1977.

The development of MARS-ND raised two important areas for consideration, the creation of motion coordination algorithms to achieve successful collaboration between the two arms when carrying out a specific task; and the development of a collision avoidance system to prevent the two arms colliding with each other or other obstacles.

C.2 Multi-Arm cooperation for task execution

A multi-arm robot system has the following advantages over single arm robot systems:

1. An object can be assembled in space without the aid of fixtures or jigs
2. It is easy to re-grasp an object for performance optimisation, if necessary, by exchanging the object between arms

3. Although a multi-arm robot system enlarges the workspace, it can work with heavy, voluminous, and non rigid objects

These advantages however, come at the expense of increased complexity with regard to control due to the additional requirements of collision avoidance, and control of kinematic and dynamic interactions.

There are two different modes of task execution by a multi-arm robot system, the distributed mode and the cooperative mode. In the distributed mode, the two arms carry out two different subtasks separately with no kinematic and dynamic interaction (J. N. Anderson 1986). In the cooperative mode, the two arms engage in one task cooperatively with kinematic and dynamic interactions between the two arms. In this mode the kinematics of one arm are constrained by those of the other arm. Sometimes a task assigned to a multi-arm robot system may require both modes of task execution.

Simple control strategies for multi-arm robot systems can be derived from control strategies for single arm systems (A. K. Ramadorai *et al.* 1994). The control architecture shown in Figure 6.1 is based on the tri-level hierarchical control of two robot arms. In this hierarchy, the low level achieves the desired motion and operates in millisecond time scale, the intermediate level determines the motion desired for the subtask in second time scale, and the high level plans the sub-task sequences in minute time scale. The high level plans the task to be performed and decomposes the task into appropriate subtasks for the right and left arms. The intermediate level transforms each subtask into a sequence of synchronous desired trajectories of end-effector motions and applied forces. The low level is concerned with the execution of the desired trajectories and employs feedback from the current status of the arms. The single-arm control strategies developed in this research can be applied at the low level of this hierarchy as shown in Figure C.1.

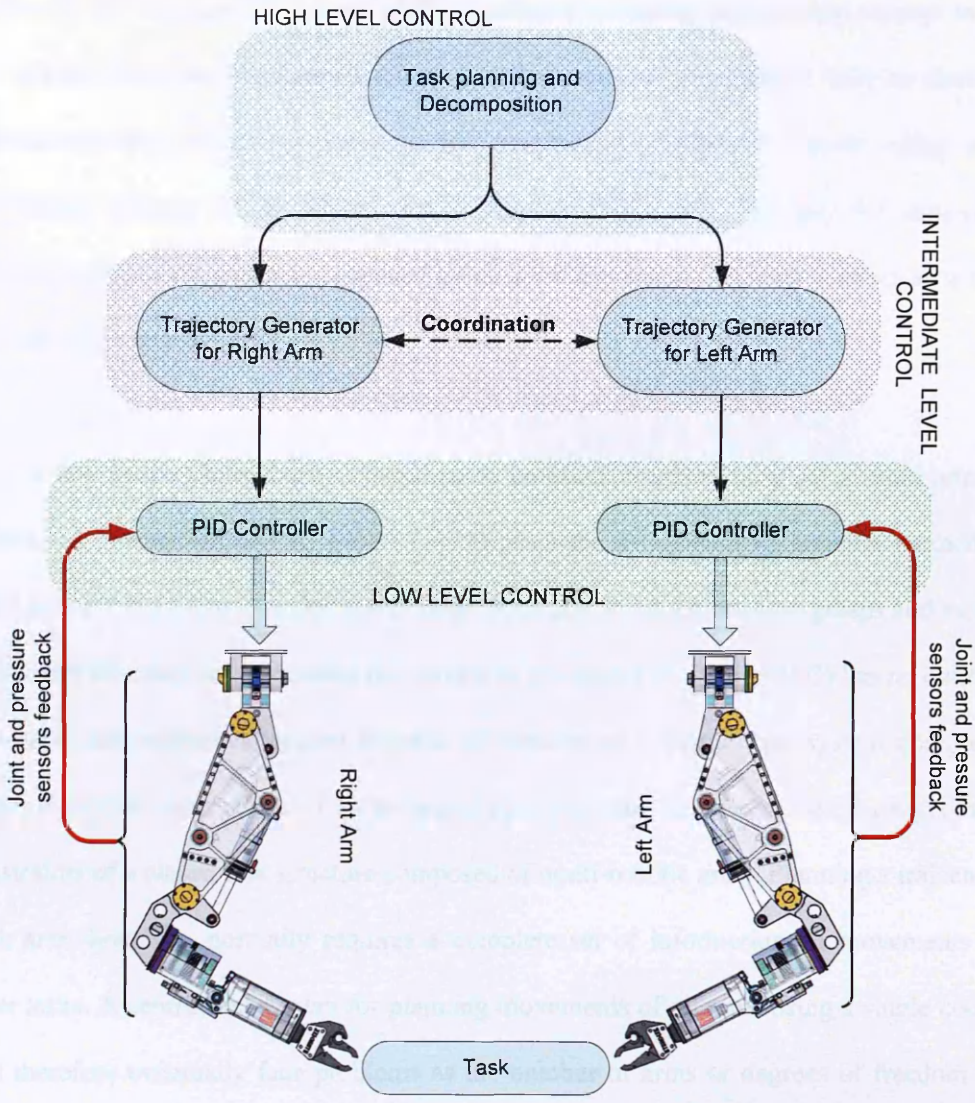


Figure C.1 Tri-level hierarchical control of multi-arm robot system

In every multi-arm strategy a suitable task-related coordinate frame of reference can be chosen for each arm, and the desired motions and applied forces of each arm can be expressed in this frame. In this way the individual arm can move as though it were carrying out the commanded motion by itself in this frame.

There are many interesting studies that have been undertaken over the last decade with regards to the generation of a trajectory for a multi-arm robot system. S. Lee (1989) proposed a method for dual-arm robots that uses a manipulability measure. O. Al-Jarrah and Y. F. Zheng (1994) suggest a method for dual-arm robots handling flexible objects using compliant motion

scheme. S. B. Moon and S. Ahmed (1991) applied a trajectory time-scaling concept to multi-arm robots. They later developed this approach to reduce computation time by developing sub-time-optimal trajectory planning for cooperative multi-arm robots using a load distribution scheme (S. B. Moon and S. Ahmed 1993). An algorithm for time-optimal trajectory generation was also proposed by F. Y. Wang and B. Pu (1993) based on a cell-to-cell mapping method.

Only a few studies have been undertaken to consider complex tasks of a multi-arm robot system. M. Yamanoto and A. Mohri (1991) proposed a trajectory generation method for a multi-arm robot system to undertake a cooperative task in which one arm grasps and moves an object and the other arm processes the surface of the object. T. Tsuji (1993) has recommended a method that utilises redundant degrees of freedom of a closed link system composed of multiple robotic arms. Both of these approaches generate trajectories based on geometrical constraints of a closed link structure composed of multi-robotic arms. Planning a trajectory for each arm, however, normally requires a complete set of information on movements of all other arms. A centralised system for planning movements of all arms using a single computer will therefore eventually face problems as the number of arms or degrees of freedom of the joints increase. These problems can include failure resistance, flexibility, and expandability. One possible solution that can be taken to overcome such problems is to construct an autonomous decentralised control system composed of a set of autonomous subsystems in a distributed manner (T. Fukuda *et al.* 1990 and R. C. Arkin *et al.* 1993).

Several other methods have also been proposed to coordinate the control of two robot arms carrying a solid object, where the object was assumed to be grasped rigidly by both arms and the relative position and orientation of the arms was fixed during the entire execution of the robot's task (Y. H. Zheng and J. Y. S. Luh 1986, J. Lim and D. H. Chyung 1985, A. K. Bejczy *et al.* 1986, N. Iwasawa *et al.* 1987). Under this setting, however, some tasks such as carrying an object along a pre-specified path cannot be accomplished due to the insufficient number of

DOFs available to each arm. An example of this is the master-slave controls coordination. In this context the generation of the motion commands for the slave arm are based on the master arm's motion. When the master arm's motion leads the slave arm to a singular region, or when the motion command generated from the master arm's motion requires the slave arm to violate its joint limits and, or to collide with obstacles, the task cannot be accomplished with the invariant grasping position unless the desired path is modified or the number of the slave arm's DOFs is increased. This difficulty can be overcome by relaxing the assumption of invariant grasping position of the slave arm. With this relaxation the slave arm becomes a redundant manipulator with no joints added physically, the slave arm's motion commands are then generated by employing the kinematic control techniques commonly used for redundant manipulators (A. Liegeois 1977, C. A. Klein and C. H. Huang 1983, W. D. Fisher 1984, P. H. Chang 1986, R. C. Gonzales *et al.* 1987, A. A. Maciejewski and C. A. Klein 1985, L. Žlajpah and B. Nemec 2002). The kinematic control algorithm of a redundant manipulator allows the joint angles and velocities to be found in such a way that the end-effector of the redundant manipulator attains the desired positions and orientations.

C.3 Coordination of two or more robot arms

In order to coordinate two or more robot arms it is necessary to make the execution of their respective movements compatible, so that they execute their tasks without colliding. This is achieved by means of adjustment of the geometric paths by fixing the velocity profiles so that the robot arms do not cross the same place at the same time. The following aspects are involved in the coordination of two or more robot arms (Todt, E., *et al* 2000):

1. Geometric Path (GP): the sequence of configurations that the robot arms follow in order to execute the task from an initial configuration to a final one
2. Trajectory (T): the geometric path plus a velocity associated with each configuration
3. Velocity Profile (VP): the description of the robot velocity as a function of the configuration

If the generation of the geometric path and the velocity profiles are determined by considering the coordination of the robots arms, then the coordination methods can be classified as coupled. If this does not occur the coordination methods are considered as decoupled. The velocity profile is independent of the geometric path; although a modification of the velocity profile implies an adjustment of space-temporal movement, the defined geometric path remains the same.

The coupled method plan the geometric path, the velocity profiles and the generation of the trajectories and their coordination of all the robot arms in one phase as inseparable processes, as shown in Figure C.2.

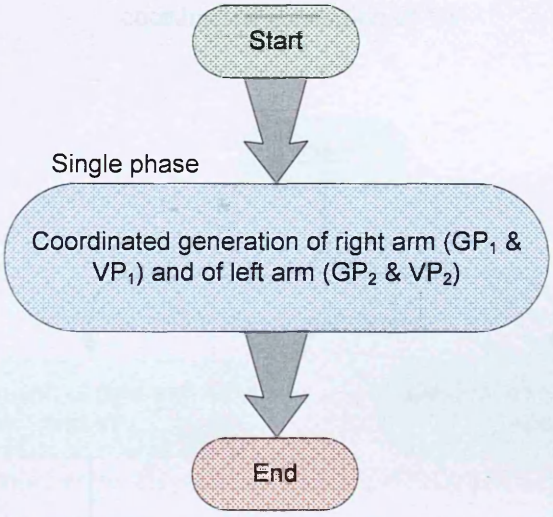


Figure C.2 Coupled coordination of trajectories

The decoupled method presents a coordination phase that is separate from the path-planning phase. The decoupled method can adjust the geometric paths, introduce pure delays in the execution of the movements, or modify the velocity profiles as shown in Figures C.3 and C.4.

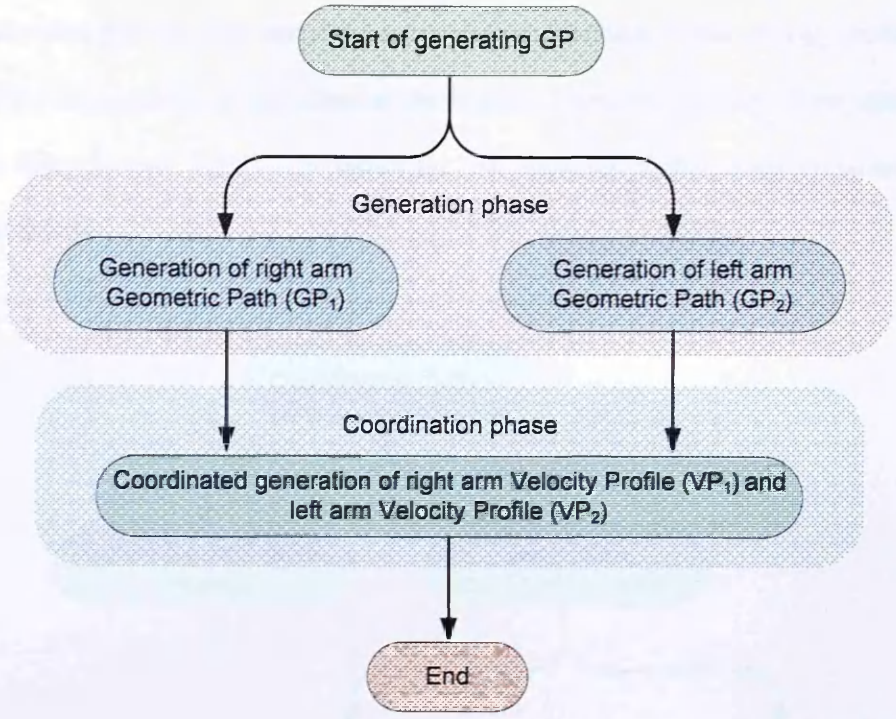


Figure C.3 Decoupled coordination of trajectories: independent generation of GP with coordinated generation of VP

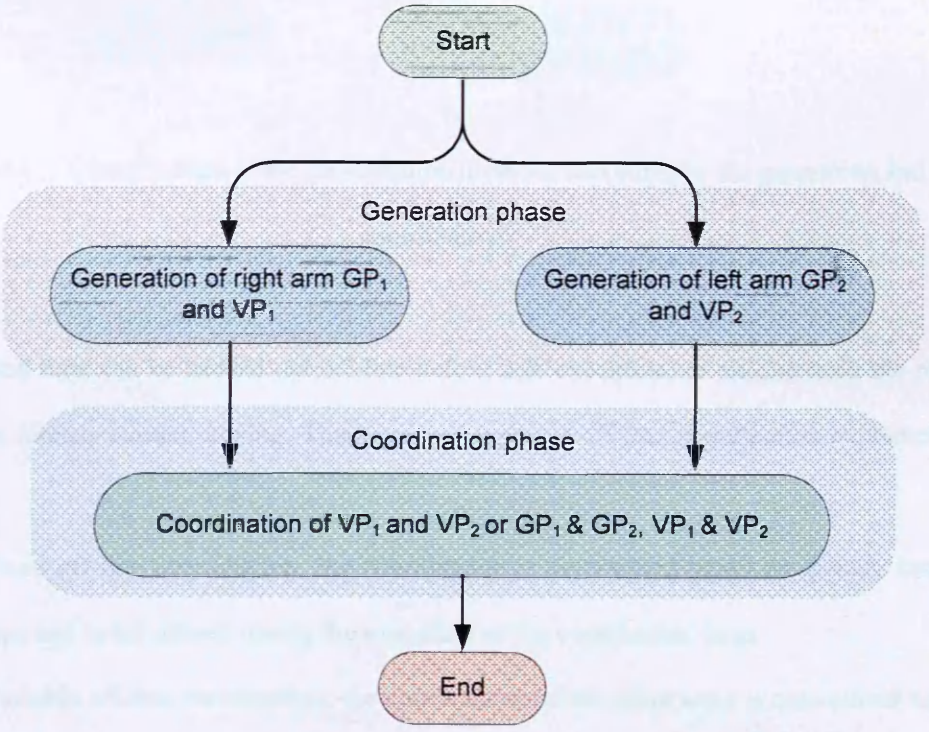


Figure C.4 Decoupled coordination of trajectories: independent generation of GP and VP with coordinated adjustment

It should be noted that the pure delay is a particular modification of the velocity profile that consists of the introduction of wait times at the instants where the velocity of the robot that suffers the delay is zero. Figure C.5 shows the classification of the coordination methods using these criteria.

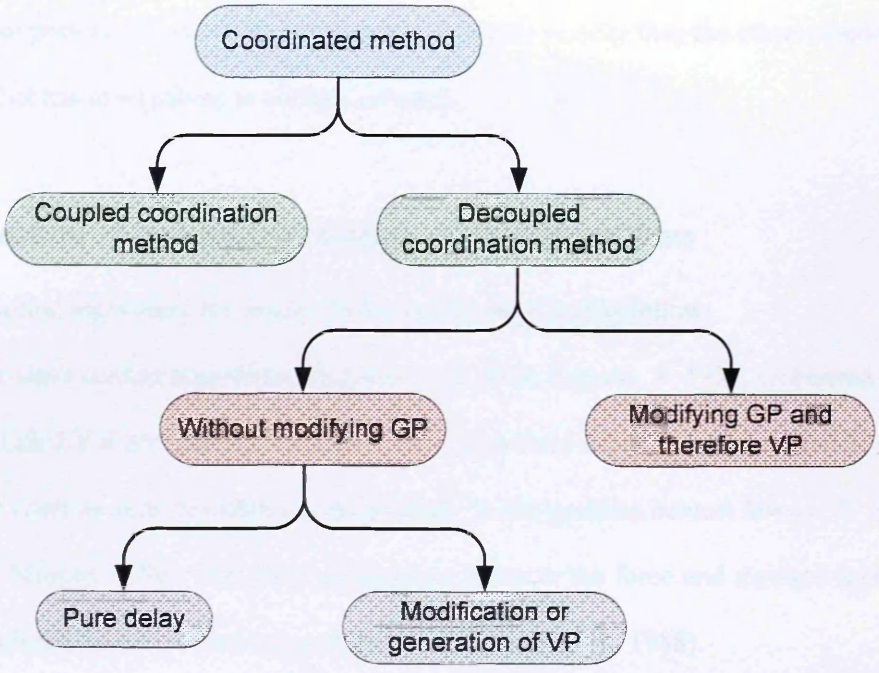


Figure C.5 Classification of the coordination methods according to the generation and coordination

Coordinated time can be carried out off-line before task execution, or on-line once the robot arms have already started moving. There are two types of off-line coordination explained as follows:

1. Fixed off-line coordination: the coordination is determined based on already known data and is not altered during the execution of the coordinated tasks
2. Variable off-line coordination: the coordination of the robot arms is determined based on already known data off-line, but there exists the possibility of choosing alternatives at certain points during the execution of the movements. For example, as a function of run-time acquired information.

Coordination priorities also exist and therefore the coordination methods can be classified as:

- With priorities: in which one of the robot arms has higher priority in the execution of its movements. In this context the other robot arm has to adapt their movements in order to avoid collisions. Different priorities can be used to define an order of priority relationship among all the robot arms of the system.
- Without priorities: none of the robot arms has higher priority than the others, therefore a conflict has to be solved to avoid a collision.

C.4 Classification of coordinated motion control algorithms

Coordinated motion algorithms for manipulators can be classified as follow:

A. Master-slave control algorithms (Nakano *et al.* 1974, Kurono, S. 1975, Uchiyama, M. 1990, Luh J.Y.S and Zheng, Y.F 1987). In this method a manipulator referred to as a master controls how the object is held based on the position control law (J. T. Wen and S. Murphy 1990). The slave manipulator controls the force and moment applied to the object, based on the force control law (T. J. Tarn et al. 1988).

B. Hybrid control algorithms (Takase, K. 1985, Hayati, S. 1986 and Tarn *et al.* 1985). In this system the robot needs to have six DOF to control the motion of the object in 3D space. It also needs six DOF to control the internal force and moment applied to the object. In this control algorithm multiple robots control the six DOF with respect to both the motion of the object and the internal force and moment applied to the object using the $6n$-DOF of n manipulators. This control algorithm is regarded as a generalisation of the master-slave algorithm.

C. Compliance based control algorithms (Hanafusa, H. and Asada, H. 1977 and Koga *et al.* 1992). In this algorithm the object is grasped compliantly through manipulator compliances or impedances realised by the hardware or the software.

D. Augmented dynamics based control algorithm (Khatib, O. 1987). This algorithm is an extension of the hybrid position and force algorithm and is based on the manipulator dynamics and the object dynamics.

The motion equation of an object supported by multiple manipulators is expressed as follows (K. Kosuge and Y. Hirata 2005):

$$m\ddot{r}_0 = F_0 + mg \tag{C1}$$

$$M\dot{w}_0 + w_0 \times (Mw_0) = N_0 \tag{C2}$$

Where

m is the mass of the object

r_0 is position vector from the origin of the absolute coordinate system to the origin of the object coordinate system

g is the acceleration gravity

w_0 is the angular velocity of the object

M is the inertia matrix of the object

F_0 and N_0 are the resultant force and the resultant moment applied to the object by all manipulators. These are expressed as follows:

$$F_0 = \sum_{i=1}^{n} F_i \tag{C3}$$

$$N_0 = \sum_{i=1}^{n} N_i \tag{C4}$$

F_i and N_i are the force and the moment applied to the object by the ith manipulator in the object coordinate system with respect to the absolute coordinate system. The motion of the object is generated based on the force F_0 and the moment N_0, which are the resultant force and the resultant moment applied to the object by the manipulators.

Putting equations (C1) and equation (C2) together, the following equation (C5) emerges:

$$L = \begin{bmatrix} mI_3 & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \ddot{r}_0 l \\ \dot{w}_0 \end{bmatrix} + \begin{bmatrix} -mg \\ w_0 \times (Mw_0) \end{bmatrix} \quad \text{(C5)}$$

Where L is the manipulator load and I is NxN identity matrix.

C.5 Recommended arm coordination for MARS-ND

The study and understanding of arm coordination issues and implementation with the multi-arm Hydro-Lek system has been an important aspect of this research. In addition if algorithms were found and formed for the coordination of the arms, then the next task would be to find a suitable collision avoidance method and implement it alongside the coordination algorithms using the motion controller developed for MARS-ND. Through the research process it became evident that the above two issues, of arm coordination and collision avoidance between two robot arms undertaking different tasks, was a new area of research that required collaboration with other research groups from the UK or internationally who were undertaking similar projects. In order to develop understanding of these issues, a six-month collaborative research project was undertaken with Sugano Laboratory at Waseda University, Tokyo, Japan. This project also involved a short visit to Kosuge and Hirata Laboratory, Tohoku University, Japan. The Sugano laboratory team and the Kosuge and Hirata team have been involved in the development of multi-arm robot systems and humanoid robot systems(Sugano 2007 and Kosuge 2007) .Within their research they encountered the same issues and developed a variety of solutions, as are discussed below.

Figure C.6a below shows a 3D CAD model of the Hydro-Lek multi-arm system. Figure C.6b shows a simplified 3D CAD model of the same system, to be used for coordination demonstration purposes.

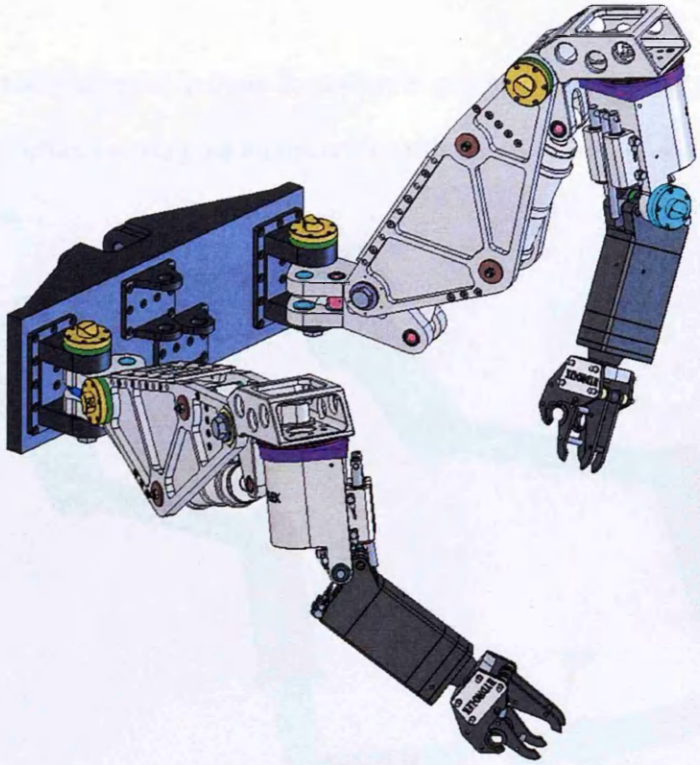


Figure C.6a 3D CAD Model of Hydro-Lek multi-arm system

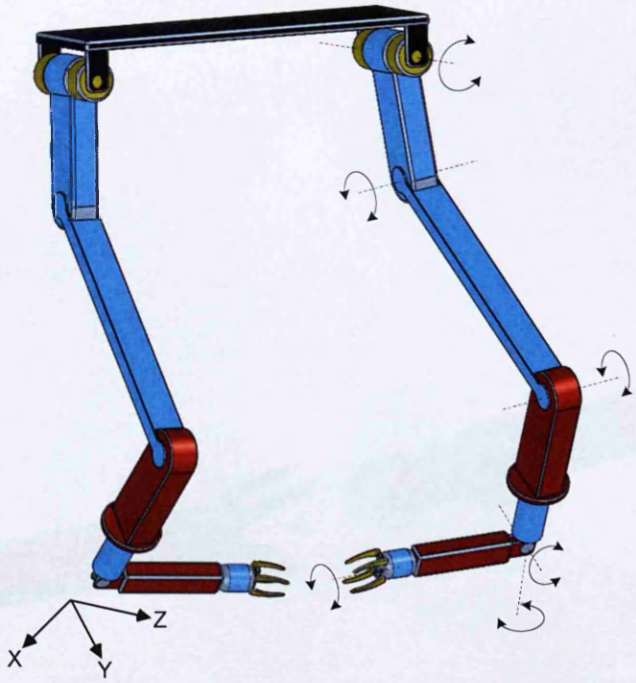


Figure C.6b Simplified 3D CAD model of Hydro-Lek multi-arm system (Bakari 2008)

After close consultation and supervision with Professor Derek Seward of the Engineering Department at Lancaster University, it was proposed that the nature of the arm coordination could be categorised in increasing degrees of complexity as follows:

1. Spatial and sequential in time: for example gripping or steadying a work-piece with one arm before carrying out an operation with the other arm, as shown in Figure C.7a and C.7b.

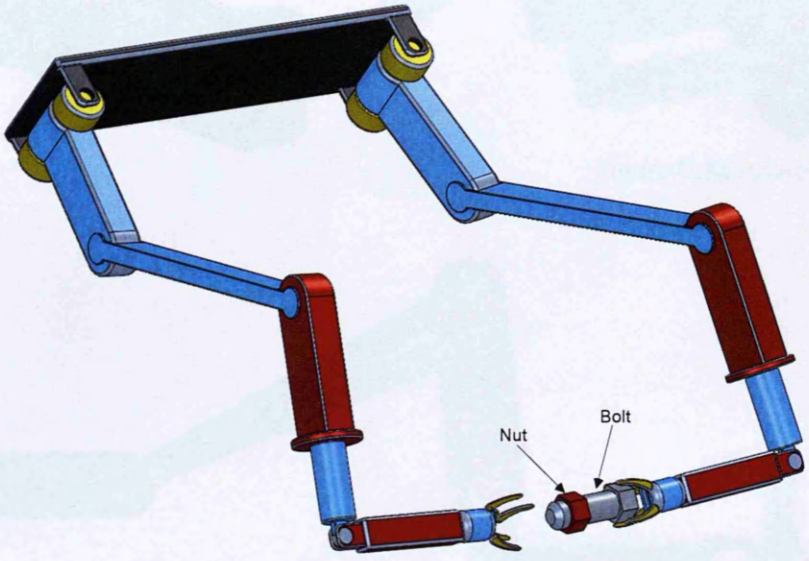


Figure C.7a

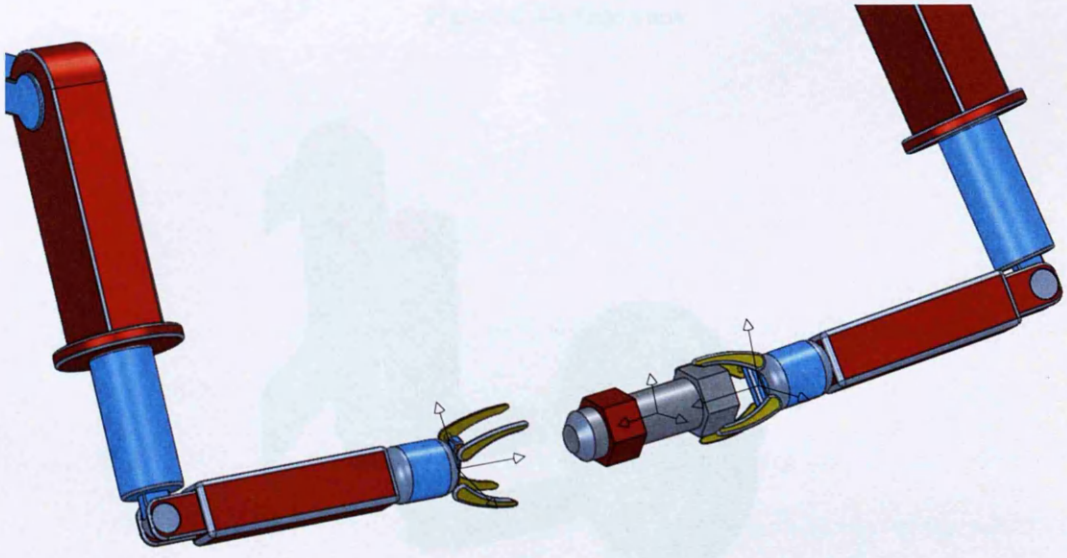


Figure C.7b Gripping and unscrewing multi-arm coordination (Bakari 2008)

2. Large parallel identical independent tasks with a possible time offset: for example undoing two wheel nuts on a car with both arms at the same time, as shown in Figure C.8a, C.8b and C.8c.

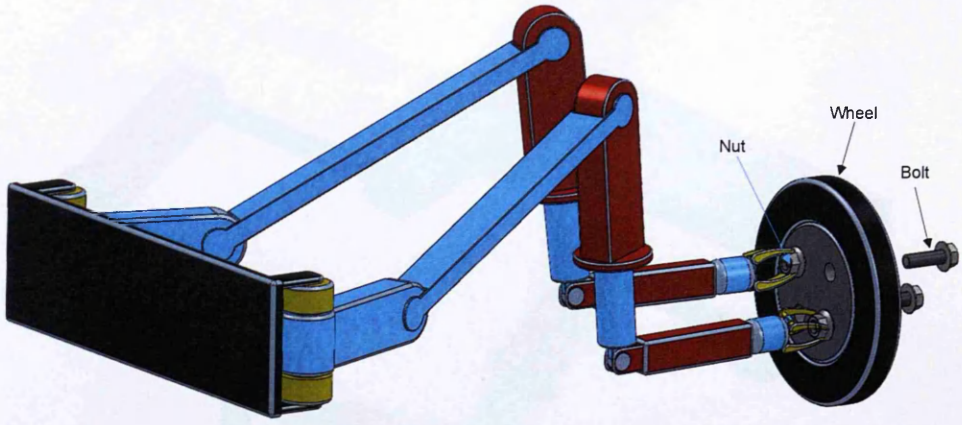


Figure C.8a Isometric view

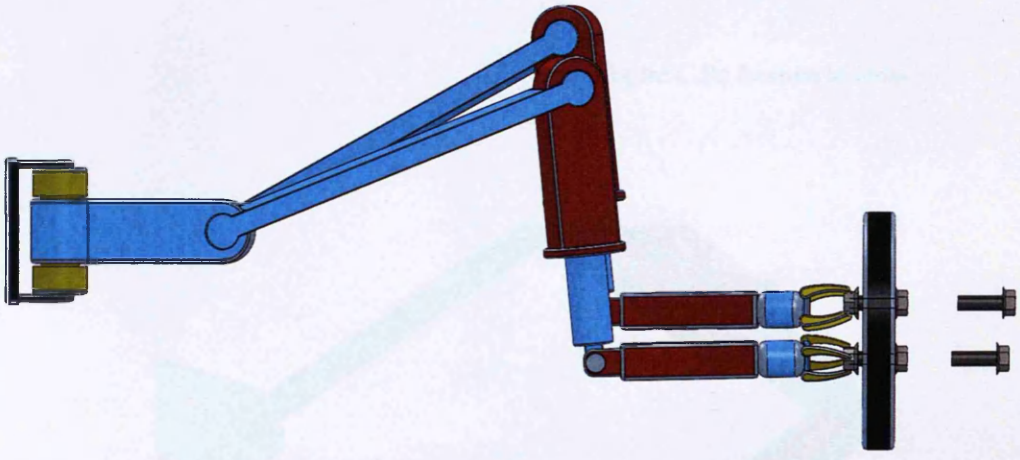


Figure C.8b Side view

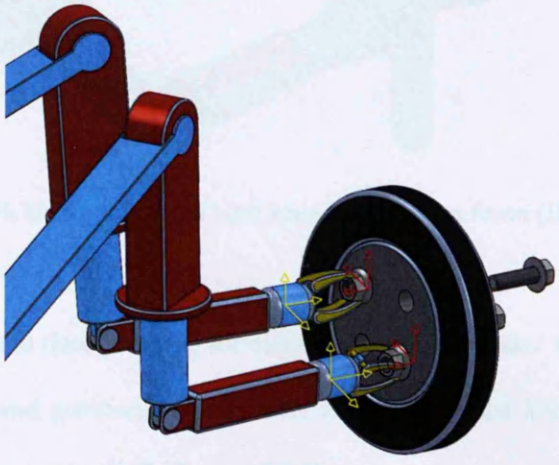


Figure C.8c Undoing two wheel nuts with both arms (Bakari 2008)

3. Close spatial and time coupling for a single task: for example using two hands on a spanner to give extra force, as shown in Figure C.9a and C.9b.

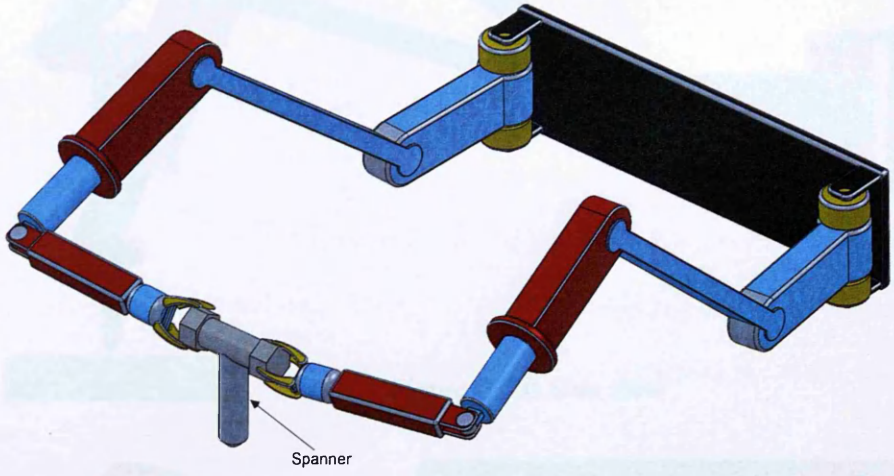


Figure C.9a Isometric view

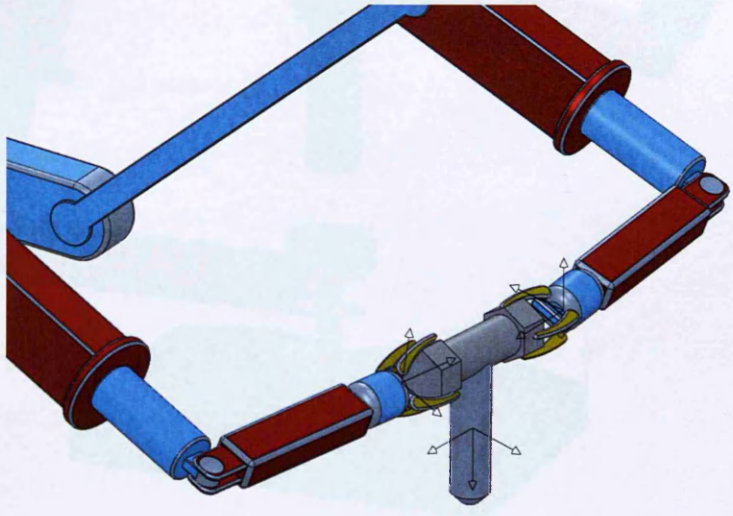


Figure C.9b Using two arms on a spanner for extra force (Bakari 2008)

4. Close spatial and time coupling for different but linked tasks: for example taking nails out of a box and positioning them with one hand whilst hammering them with the other, as shown in Figure C.10a and C.10b.

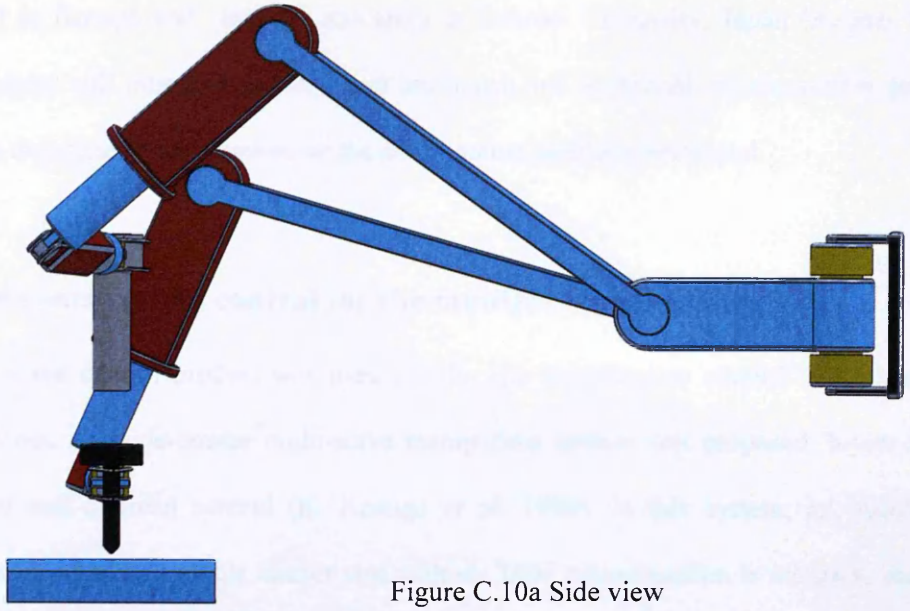


Figure C.10a Side view

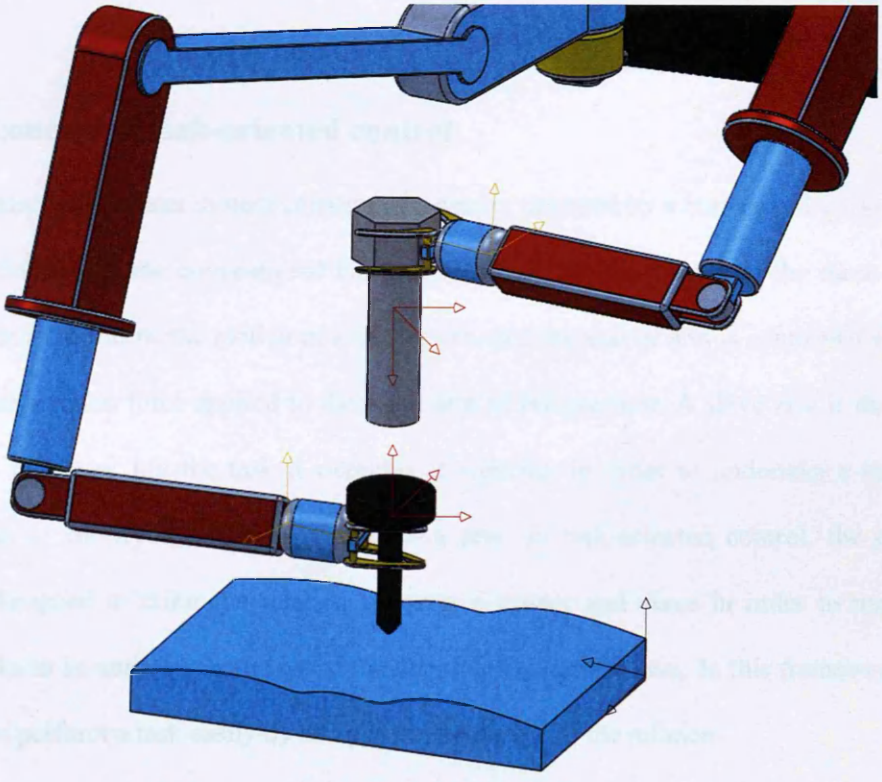


Figure C.10b Positioning a nail on a work piece with one arm whilst hammering them with the other arm (Bakari 2008)

The following sections discuss the collaborative work carried out in Japan with the Sugano group in the Department of Mechanical Engineering at the School of Science and Engineering, Waseda University, Tokyo, Japan (March-September 2008); which included a

short visit to Kosuge and Hirata Laboratory at Tohoku University, Japan (August 2008). These sections will introduce the different multi-arm and humanoid robotic system projects within the departments and summarise the coordination methods developed.

C.6 Arm coordination control for the multiple mobile robots

A master slave control method was used for the arm coordination control of the multiple mobile robots. A single-master multi-slave manipulator system was proposed, based on the concept of task-oriented control (K. Kosuge *et al.* 1990). In this system, an operator is required to manipulate a single master arm with six DOF whose motion is related to the task that the robot has to undertake.

C.7 The concept of task-oriented control

A master-salve manipulator system consists of a master operated by a human and a slave used for real operations. In the conventional bilateral feedback system, motion of the slave arm is controlled so as to follow the motion of a master arm and the master arm is controlled so as to feedback the reaction force applied to the slave arm to the operator. A slave arm is designed for general purposes, but the task it executes is specific. In order to undertake a task the operator has to specify the motion of the slave arm. In task-oriented control, the control system is designed to tailor the relation between a master and slave in order to meet the specific tasks to be undertaken and assist the operator to execute them. In this framework, the operator can perform a task easily by an appropriate choice of the relation.

The task-oriented control developed for the multiple mobile robots used control architecture for the slave arm and the master arm that controlled the task-oriented variables using Virtual Internal Model (VIM), as discussed by K. Kosuge *et al.* 1990. VIM is a reference model driven by sensory information and used to describe the desired relation between the motion of a master arm and task oriented variables.

C.8 Collision avoidance methods

The artificial potential field (APF) approach proposed by O. Khatib (1986) for obstacle avoidance was adopted at the Sugano Laboratory. The essential concept of APF is to make local decisions at each step based on the distance vectors to the goal and various obstacles that eventually leads to the goal position. This method treats the robot, represented as a point in configuration space, as a particle acting under the influence of a potential field whose local variations are expected to reflect the structure of the free space. The potential function is defined over the free space as the sum of an attractive potential which pulls the robot toward the goal configuration; and a repulsive potential which pushes the robot away from the obstacles to prevent collisions. Virtual forces are defined by negative gradients of potential function. The robot is controlled by the sum of the force moving from high potential configuration to low potential configuration.

C.9 The potential field function and its modelling

Figure C.11 below shows the processes involved in the creation of repulsive forces to prevent two robot manipulators colliding while carrying out a task in a close restricted environment.

C.9.1 Direct kinematic calculation

Direct kinematic calculation involves the identification of each point of the sphere created for every joint of the robot arm and any static or moving obstacles surrounding the robot system. Figure C.12 below shows a multi-arm robot system with a sphere attached to each joint of the arms and the obstacles.

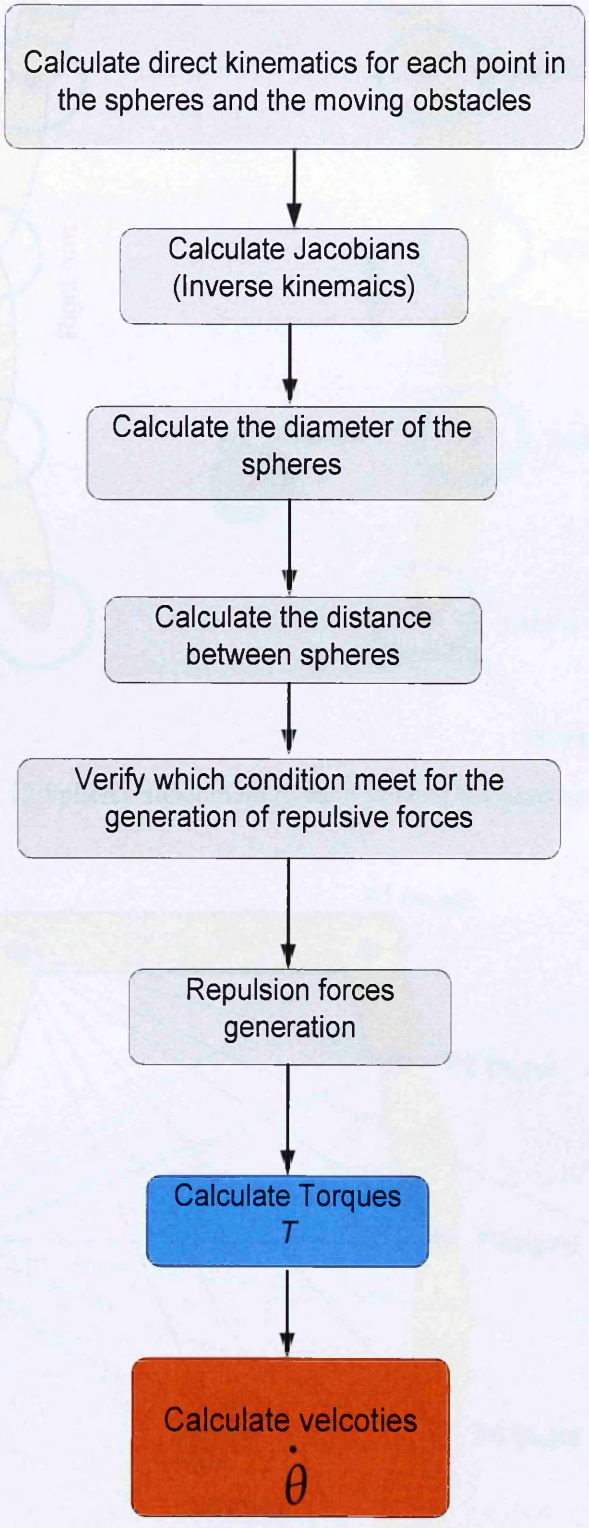


Figure C.11 Flowchart for repulsive forces generation

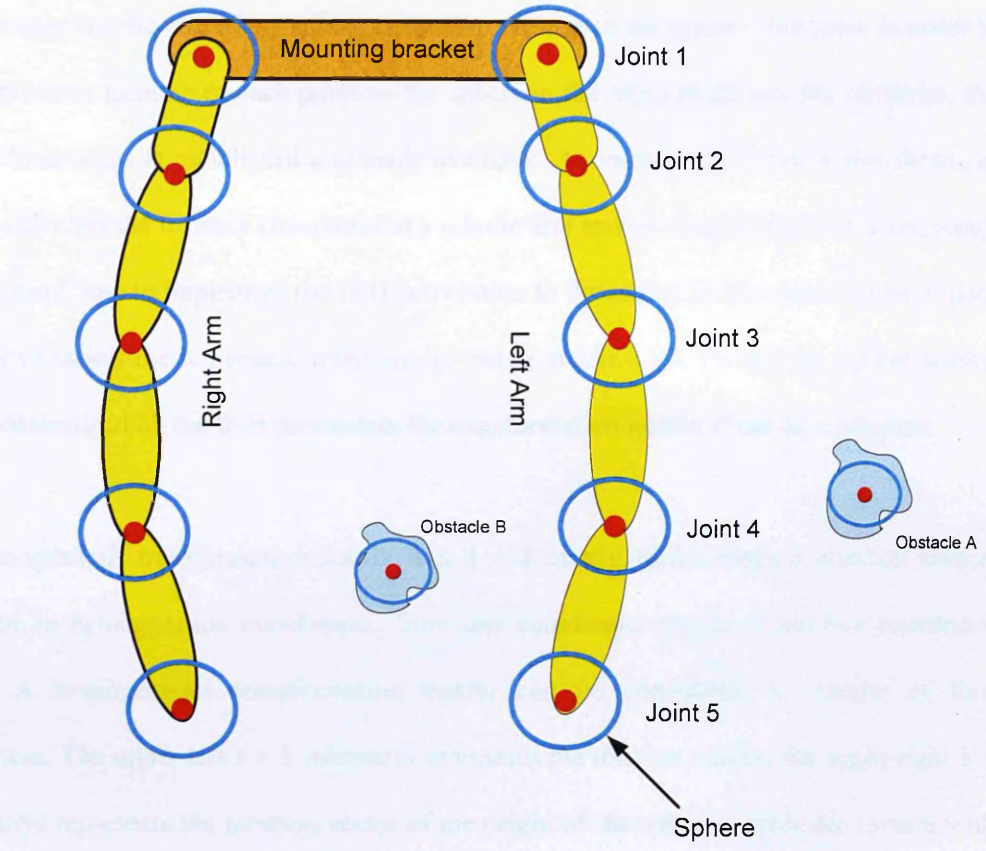


Figure C.12 Spheres attachment to each joint of the arms and obstacles

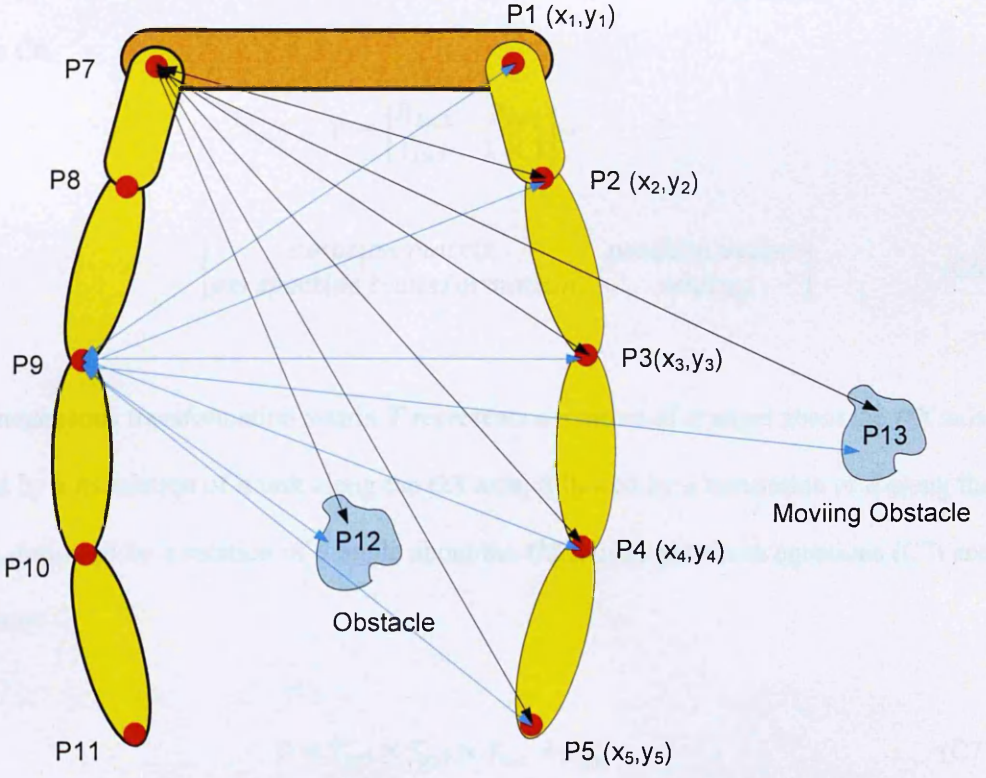


Figure C.13 Kinematic relationships between sphere points

It is desirable that the size of the sphere is equal to the size of the actual robot joint. In order to obtain the exact location of each point on the sphere in the robot joints and the obstacles, the forward kinematics is established and made available. As mentioned earlier in this thesis, in order to calculate the forward kinematics of a robotic arm and moving obstacles it is necessary to understand how to implement the D-H convention to determine its four parameters. Figure C.13 above shows the Kinematic relationships between joints P7, P9 and the sphere points. Having determined all the D-H parameters the transformation matrix $\boldsymbol{T}$ can be computed.

The homogeneous transformation matrix is a 4 × 4 matrix which maps a position vector, expressed in homogeneous coordinates, from one coordinate system to another coordinate system. A homogeneous transformation matrix can be considered to consist of four submatrices. The upper left 3 × 3 submatrix represents the rotation matrix; the upper right 3 × 1 submatrix represents the position vector of the origin of the rotated coordinate system with respect to the reference system; the lower left 1 × 3 submatrix represents the perspective transformation; and the fourth diagonal element is the global scaling factor, as shown in equation C6.

$$T = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ f_{1\times3} & 1\times1 \end{bmatrix} =$$

$$\begin{bmatrix} rotation\ matrix & position\ vector \\ perspective\ transformation & scaling \end{bmatrix} \tag{C6}$$

The homogeneous transformation matrix $\boldsymbol{T}$ represents a rotation of α angel about the $\boldsymbol{OX}$ axis, followed by a translation of a unit along the $\boldsymbol{OX}$ axis, followed by a translation of d along the $\boldsymbol{OZ}$ axis, followed by a rotation of θ angle about the $\boldsymbol{OZ}$ axis as shown in equations (C7) and (C8) below:

$$T = T_{z,\theta} \times T_{z,d} \times T_{x,a} \times T_{x,\alpha} \tag{C7}$$

$$T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\cos\alpha\sin\theta & \sin\alpha\sin\theta & a\cos\theta \\ \sin\theta & \cos\alpha\cos\theta & -\sin\alpha\cos\theta & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(C8)}$$

Performing the composition from the *n*th frame to the base frame is illustrated in Figure C.14 below:

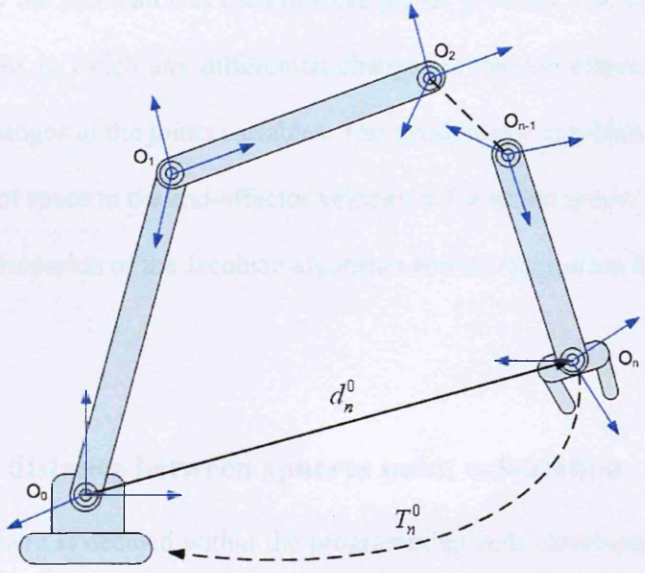


Figure C.14 Transformation from end-effector frame to base frame

. The transformation matrix can be written as:

$$T_n^0 = T_1^0 \times T_2^1 \times T_3^2 \ldots\ldots\ldots\ldots \times T_n^{n-1} = \begin{bmatrix} R_n^0 & d_n^0 \\ 0 & 1 \end{bmatrix} \quad \text{(C9)}$$

Given a set of joint angles, the forward (direct) kinematics problem is simply to compute the position and orientation of the tool-tip frame relative to the base frame of the robot arm.

C.9.2 The Jacobian calculation

The next step is the calculation of Jacobian, which is the numerical solution algorithm for the inverse kinematics of the robot arm.

The solutions of the manipulator inverse kinematics can be split into two categories:

1. Closed form solutions: In which the forward kinematics may be rewritten in a manner that leads to a set of highly structured non-linear equations that may be solved explicitly for the joint variables.
2. Numerical solutions: In which a numerical algorithm is applied that explicitly generates all solutions in a computationally feasible manner.

At the Sugano laboratory the Jacobian was used to solve the IK problem. The Jacobian matrix is a matrix of differentials in which any differential changes in the end-effector location are caused by differential changes in the joints variables. The manipulator Jacobian matrix relates the joint velocities in joint space to the end-effector velocity in Cartesian space. Please refer to Chapter Four for a full discussion of the Jacobian algorithm and its application for a multi-arm robot system.

C.9.3 Diameter and distance between spheres point calculation

The diameter of each sphere is decided within the programming code developed and used by Sugano team. As mentioned earlier, it is desirable to keep the diameter of the sphere the same as the joint diameter. If the diameter of the sphere is bigger than the actual joint diameter then this will lead to undesirable restrictions for the arms to move around. It will also reduce the flexibility of movement of the arms while carrying out a given task. If the diameter of the sphere is smaller than the actual size of the joint diameter this increases the chance of collision avoidance. There is no time for the program code to help generate repulsive forces as illustrated in Figure C.15 as the ranges tend to be smaller.

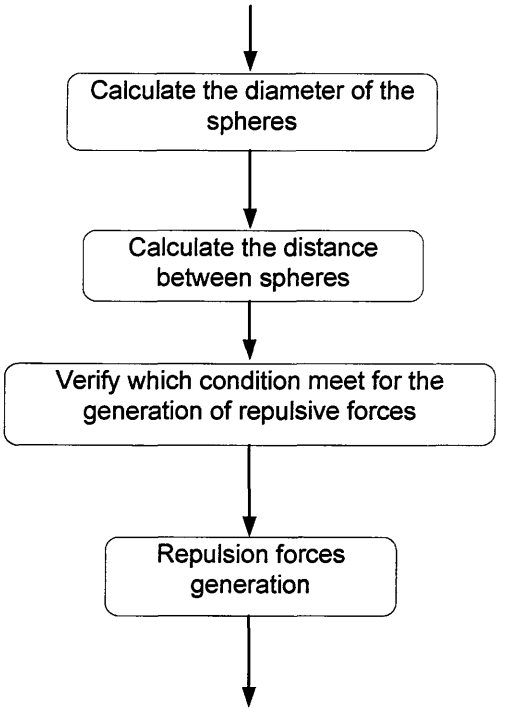


Figure C.15 Repulsive force generation process

C.9.3.1 Setting up the spheres parameter

Figure C.16 below demonstrates the parameters of the spheres for joint 4 of the multi-arm robot model used in this chapter for the demonstration.

Where: $r_{4A}(j)$ represents the radius of the sphere for joint 4 of the left arm

$r_{4A}(i)$ represents the radius of the sphere for joint 4 of the right arm

$r_F(j)$ represents the range of the repulsion force

$r_S(i)(j)$ represents the distance between the two spheres position matrix

P(i) represents the position coordinate of joint 4 of the left arm

P(j) represents the position coordinate of joint 4 of the right arm

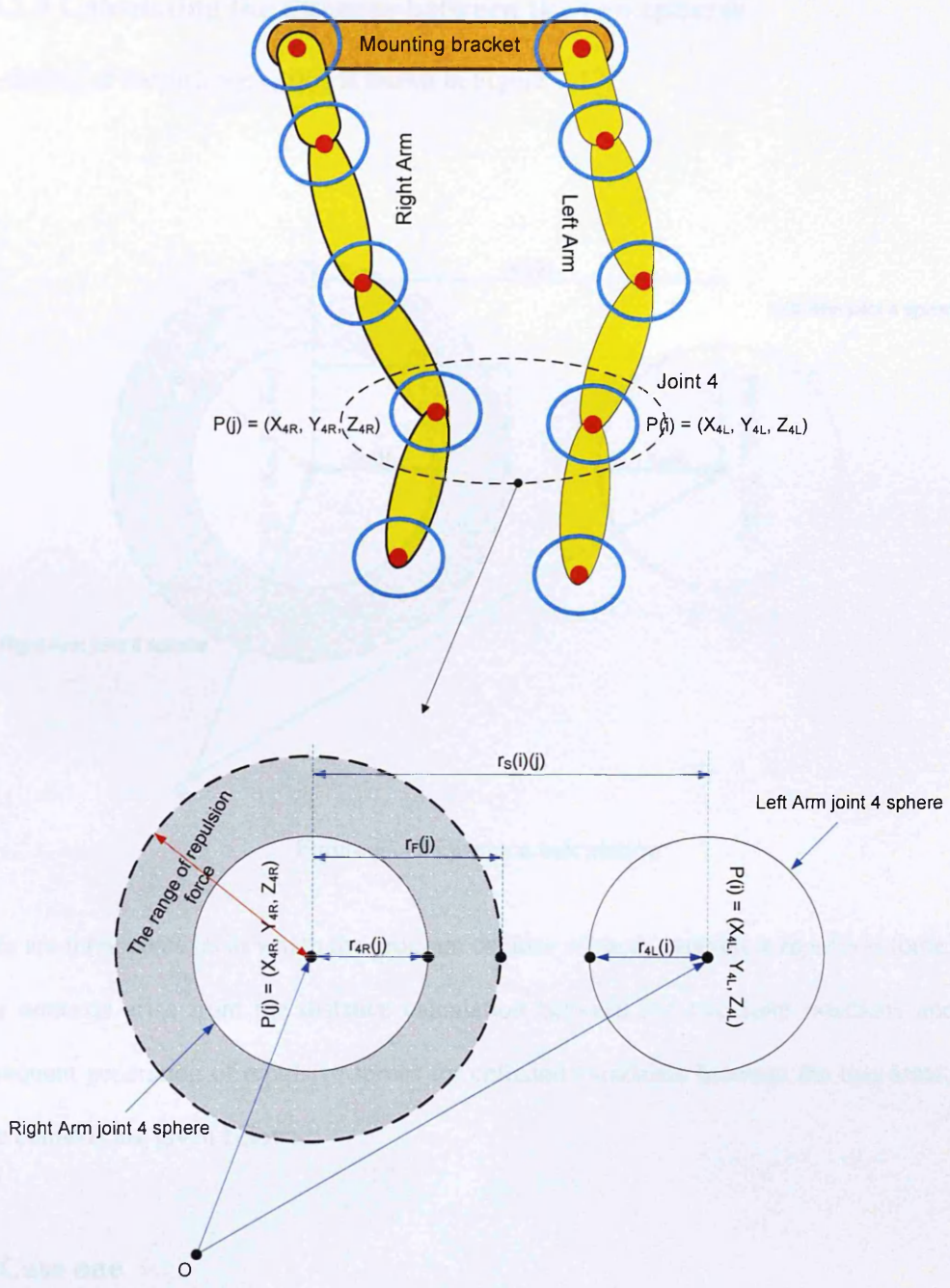


Figure C.16 Parameters of the spheres for joint 4

C.9.3.2 Deciding the repulsive force range

From Figure C.16 the repulsion force range can be decided in advance. It is desirable that the range is adequate to stop both arms colliding into each other. If the range is smaller than that required this will lead to late repulsion force generation and eventual collision.

C.9.3.3 Calculating the distance between the two spheres

The finding of the distance $r_S(i)(j)$ is shown in Figure C.17.

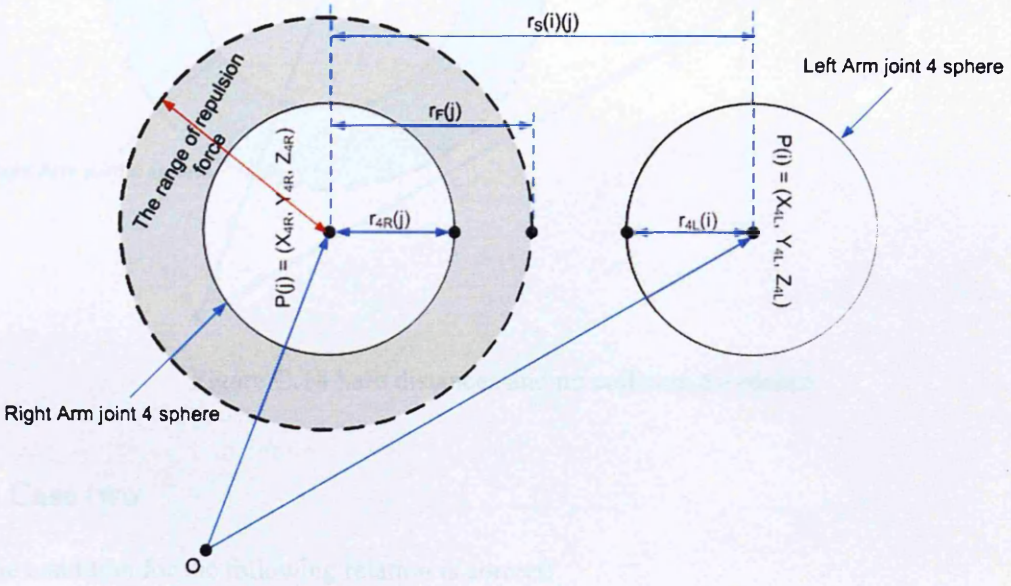


Figure C.17 Distance calculation

There are three contexts in which the program decides when to generate a repulsion force. The three contexts arise from the distance calculation between the two joint positions and the consequent generation of repulsive forces for collision avoidance between the two arms. The three contexts are given below:

(a) Case one

If the condition for the following relation is correct:

$$r_F(j) + r_{4L}(i) < r_S(i)(j) \tag{C10}$$

then there will be no repulsion force generated as the distance needed to avoid collision is safe enough for both arms to move in the workspace. This is illustrated in Figure C.18 below:

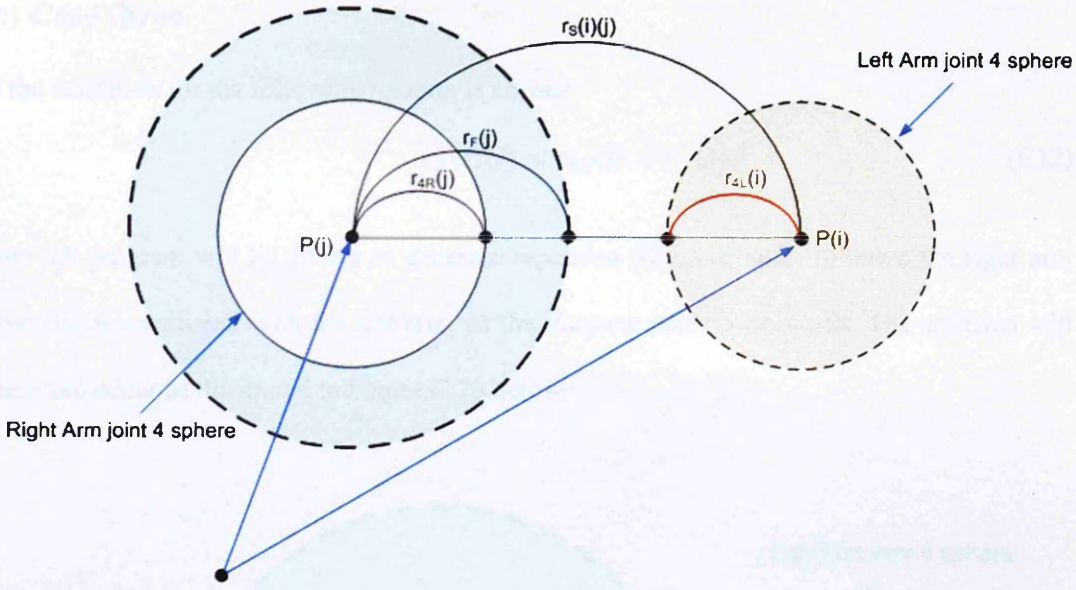


Figure C.18 Safe distances and no collision avoidance

(b) Case two

If the condition for the following relation is correct:

$$r_{4L}(i) + r_{4R}(j) < r_S(i)(j) < r_F(j) + r_{4R}(j) \qquad \text{(C11)}$$

then the program will generate repulsion forces in order to move the right arm away from a potential collision with the left arm as illustrated in Figure C.19 below:

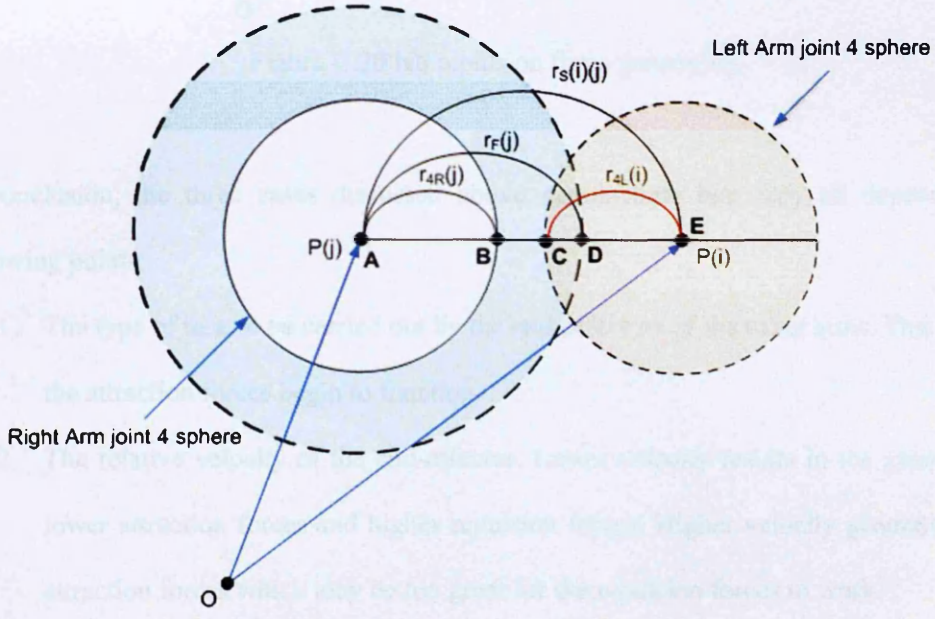


Figure C.19 Non safe distances and possible collision occur

(c) Case three

If the condition for the following relation is correct:

$$r_S(i)(j) < r_{4L}(i) + r_{4R}(j) \tag{C12}$$

then the program will be unable to generate repulsion forces in order to move the right arm away from a collision with the left arm, as the distance will be too small. The collision will therefore occur as illustrated in Figure C.20 below:

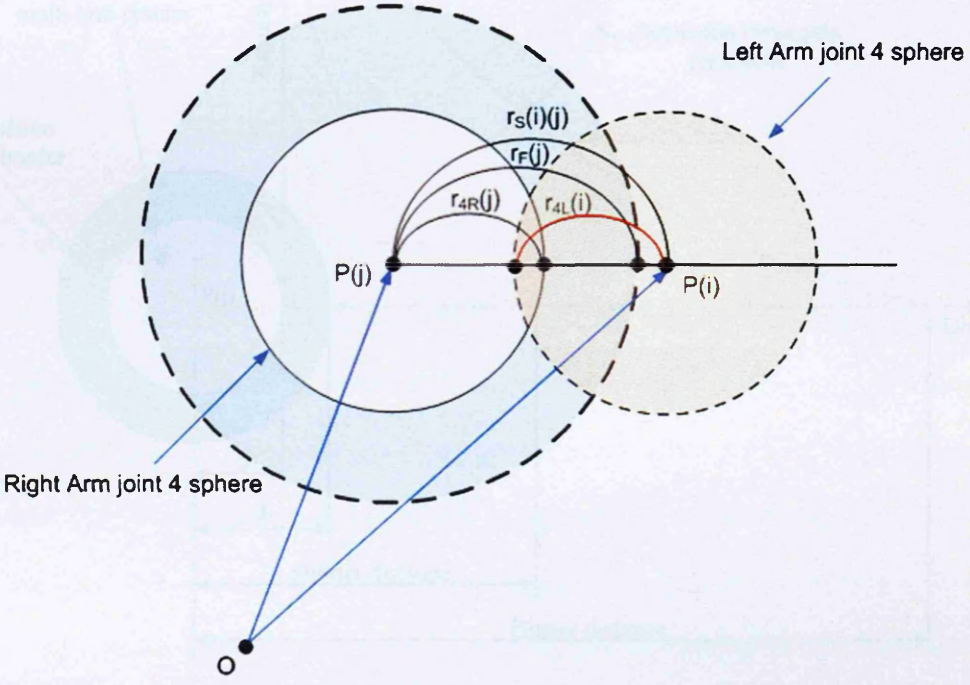


Figure C.20 No repulsion force generation

In conclusion, the three cases discussed above demonstrate that they all depend on the following points:

1. The type of task to be carried out by the end-effectors of the robot arms. This is where the attraction forces begin to function.
2. The relative velocity of the end-effector. Lower velocity results in the generation of lower attraction forces and higher repulsion forces. Higher velocity generates higher attraction forces which may be too great for the repulsion forces to work.

There are also gain parameters for the repulsion forces that need to be set up during the experimentations. These gains are set according to the type of task the robot arms will undertake. The gain parameters are illustrated below in Figure C.21:

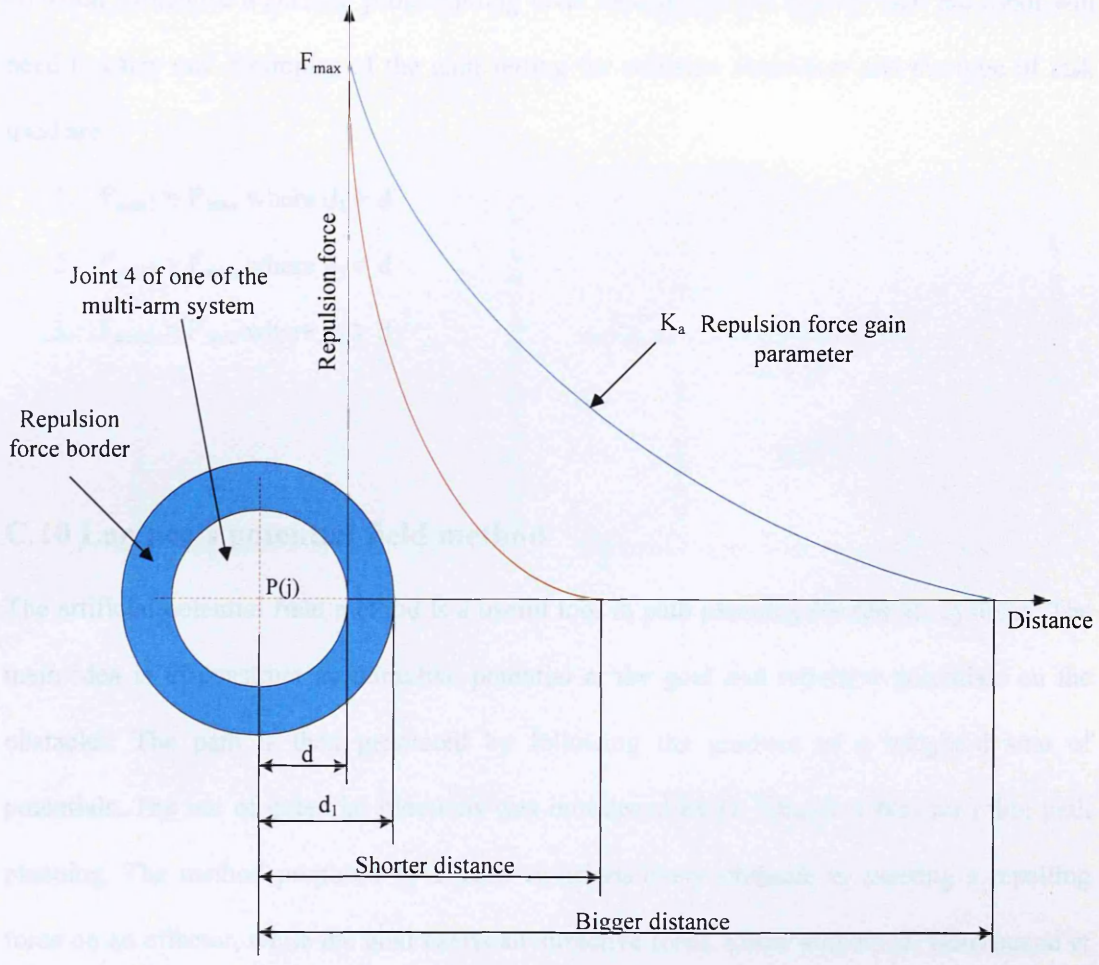


Figure C.21 Gain parameters for the generation of repulsion forces (Bakari 2008)

In Figure C.21:

d: represents $r_{4A}(j)$ which is the radius of the sphere for joint 4 of the left arm

d_1: represents $r_F(j)$ which is the range of the repulsion force (border)

Shorter and larger distance: represents $r_S(i)(j)$ which is the distance between the two sphere position matrix

Figure C.21 shows that if the distance between the two links of the multi-arm system are too close then the collision avoidance algorithm will quickly generate a maximum repulsion force

in a short time in order to prevent collision occurring; if the distance between the two links is big enough but can get closer then the algorithm will generate a repulsion force with plenty of time. As mentioned earlier, the repulsion force gain parameter that needs to be set in the collision avoidance algorithm programming code depends on the type of task the robot will need to carry out. Examples of the gain setting for collision avoidance and the type of task used are:

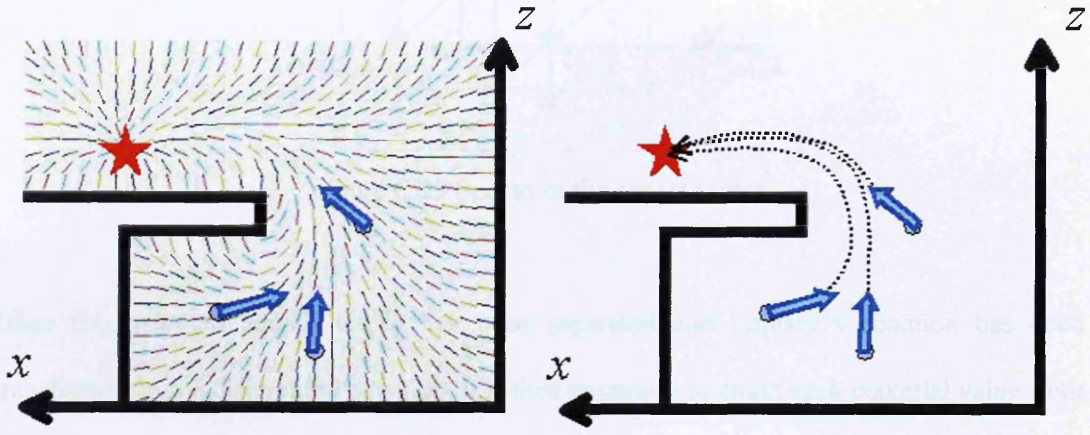
1. $F_{max1} = F_{max}$ where $d_1 > d$
2. $F_{max2} > F_{max}$ where $d_1 = d$
3. $F_{max3} > F_{max}$ where $d_1 > d$

C.10 Laplace's potential field method

The artificial potential field method is a useful tool in path planning for robotic systems. The main idea is to construct an attractive potential at the goal and repulsive potentials on the obstacles. The path is then generated by following the gradient of a weighted sum of potentials. The use of potential functions was introduced by O. Khatib (1985) for robot path planning. The method proposed by Khatib envisions every obstacle as exerting a repelling force on an effector, while the goal exerts an attractive force. Other authors (J. Barraquand et al. 1989; Bruce H. Krogh 1984; W. S. Newman and N. Hogan 1986; Damian Lyons 1986; R. C. Arkin 1987; J. K. Myers 1985; D. E. Koditschek 1987) have used a variety of potential functions all based on this underlying approach. The speed and facility of this method make it a useful tool for constructing paths for robots. The usual formulation of potential fields for path construction does not preclude the spontaneous creation of local minima and the achievement of a stable configuration short of the goal. Several authors (J. Barraquand *et al.* 1989; O. Khatib 1985; R. C. Arkin 1987) have mentioned this problem. The following section discusses the application of this method by the Sugano Laboratory for Twendy-One Robotic system.

C.10.1 Summary of Laplace's potential field method used by Sugano group

The Sugano group stated that they used Laplace's potential field method as illustrated in Figure (C.22a) for simple reaching and handling. This method was applied in conjunction with a roadmap method (RTT method) as shown in Figure (C.22b) for accurate trajectory following.



(a) (b)

Figure C.22 (a) Potential field method, (b) Roadmap method (Sugano Lab. 2008)

This method was applied for TWENDY-ONE Robot because this human symbolic robot needs to be safe at all times when operating in an environment full of humans and when following a human. In addition Laplace's potential field method has advantages over other approaches because there are no deadlocking problems and the robot can reach its designated destination. The disadvantage of the Laplace's potential method is that its implementation in the PC controller requires a very large amount of calculation. The Laplace's potential method involves the following steps:

1. Separating the space to the square lattice and transforming Laplace's equation to a differential equation as explained in Figure C.23 and equation (C13)
2. Setting the lattices with the initial value and boundary condition (see Figure C.24)
3. Attaining continuous potential field by interpolation as illustrated in Figure C.25 and equation (C14)

4. Obtaining the forces by the steepest descent method from the potential field as demonstrated in equation (C15)

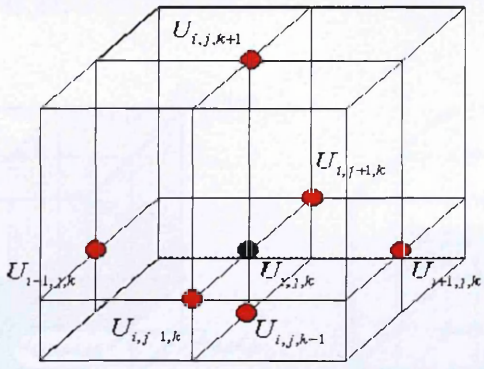


Figure C.23 Spaces in the square lattice

Once the space to square lattice has been separated and Laplace's equation has been transformed into a differential equation, it is then necessary to attain each potential value from the 3-dimensional version of Laplace's equation by averaging the nearest six potential values.

Laplace's equation:

(C13)

For TWENDY-ONE robot, the initial value and boundary condition set for the lattice were based on the conditions that the potential value of the destination had to be minimised; and the potential values of the obstacles and the destination after the interpolation did not change.

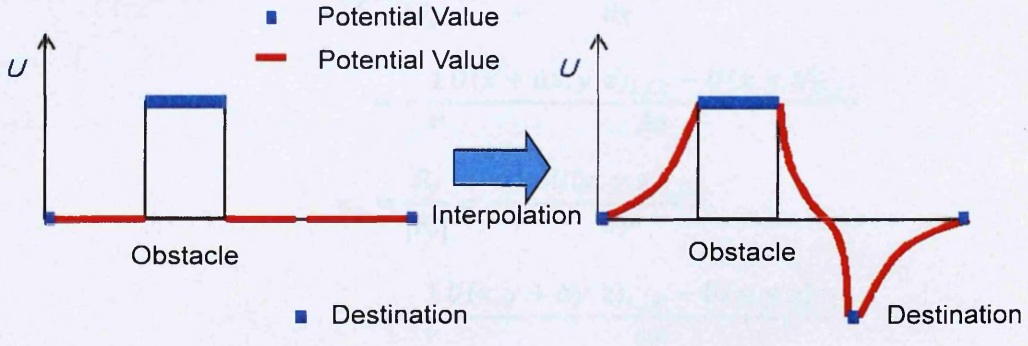


Figure C.24 Setting initial value and boundary condition

In order to obtain the continuous potential field by interpolation, it was necessary to add together the eight potential values multiplied by the volume of the opposing corners as illustrated below in Figure C.25:

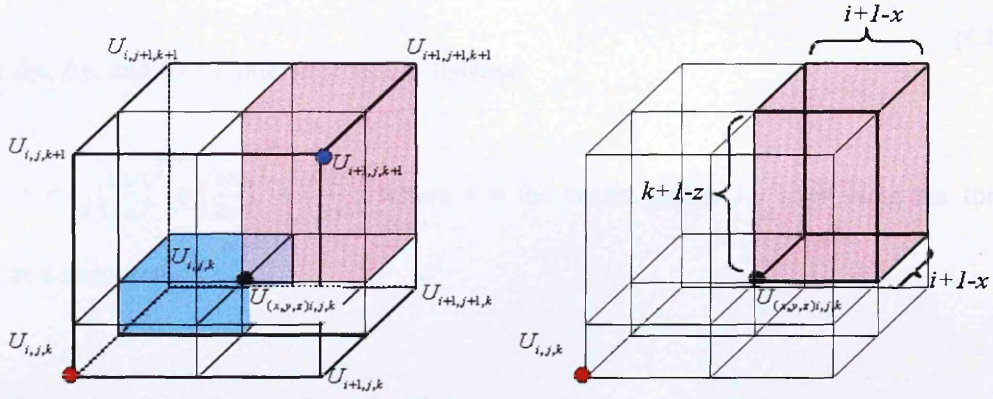


Figure C.25 Attaining continuous potential fields by interpolation

$$\begin{aligned} U(x,y,z)_{i.j.k} = {} & U_{i,j,k}(i+1-x)(j+1-y)(k+1-z) \\ & +U_{i+1,j,k}(-i+x)(j+1-y)(k+1-z) \\ & +U_{i,j,k+1}(i+1-x)(j+1-y)(-k+z) \\ & +U_{i+1,j,k+1}(-i+x)(j+1-y)(-k+z) \\ & +U_{i,j+1,k}(i+1-x)(-j+y)(k+1-z) \\ & +U_{i+1,j+1,k}(-i+x)(-j+y)(k+1-z) \\ & +U_{i,j+1,k+1}(i+1-x)(-j+y)(-k+z) \\ & +U_{i+1,j+1,k+1}(-i+x)(-j+y)(-k+z) \end{aligned} \tag{C14}$$

Obtaining the forces by the steepest descent method from the potential field:

$$e_x = \frac{F_x}{|F_x|} = -\frac{1}{r}\frac{\partial U(x,y,z)_{i,j,k}}{\partial x}$$

$$= -\frac{1}{r}\frac{U(x+\Delta x,y,z)_{i,j,k} - U(x,y,z)_{i,j,k}}{\Delta x}$$

$$e_y = \frac{F_y}{|F_y|} = -\frac{1}{r}\frac{\partial U(x,y,z)_{i,j,k}}{\partial y}$$

$$= -\frac{1}{r}\frac{U(x,y+\Delta y,z)_{i,j,k} - U(x,y,z)_{i,j,k}}{\Delta y}$$

$$e_z = \frac{F_z}{|F_z|} = -\frac{1}{r}\frac{\partial U(x,y,z)_{i,j,k}}{\partial z}$$

$$= -\frac{1}{r}\frac{U(x,y,z+\Delta z)_{i,j,k} - U(x,y,z)_{i,j,k}}{\Delta z} \quad \text{(C15)}$$

Where Δx, Δy, and Δz << interstice lattice distance

and $r = \sqrt{\left(\frac{\partial U}{\partial x}\right)^2 + \left(\frac{\partial U}{\partial y}\right)^2 + \left(\frac{\partial U}{\partial z}\right)^2}$ Where r = the vector length for expressing the force vector as a unit vector.

C.11 Collision avoidance for Mr Helper

There are a number of studies that have examined the development of mobile robot assistants. These include B. Graf *et al.* (2002) who developed mobile robot assistants called Care-O.Bot to assist humans in the home and in the production environment. O. Khatib *et al.* (1999) who proposed the concept of the "robot assistant" using a mobile manipulator. H. Hashinuma *et al.* (2002) who have studied the applications of humanoid robots including tele-operations of construction machines by the humanoid robot "HRP" and construction works in the open air. This research has found that humanoid robots and mobile manipulators, both of which have redundant DOF, are very useful for realising a variety of tasks in cooperation with a human. These robot systems are however likely to have self-collisions because of their redundancy. This can include collisions between the robot's body and its arms, and collision between two legs. When self-collision occurs the robot can be damaged or lose its balance and possibly harm humans working around the robot. The possibility of self-collision increases when the robot cooperates with humans because the motion of the robot can be affected by interaction between the robot and the human. It is therefore essential to solve this problem when a robot works in an environment where humans are also working.

K. Kosuge *et al.* (2000) have developed a mobile robot with dual arms referred to as Mobile Robot Helper (Mr Helper). The task of Mr. Helper is to handle an object in cooperation with a

human based on impedance control. The Kosuge and Hirata laboratory have developed a real-time self-collision avoidance system for robots based on cooperation with humans. They have proposed a method of representing the robot's body by elastic elements; this approach is referred to as RoBE (Representation of Body by Elastic elements). Within their research, they have also proposed control algorithms to avoid self-collisions in real-time using RoBE.

APPENDIX D

CLOSED FORM AND NUMERICAL SOLUTIONS OF INVERSE KINEMATICS FOR HYDRO-LEK ARM

D.1 Exact solution

$$T_0^1 = \left[\begin{array}{ccc:c} C_1 & 0 & S_1 & a_1C_1 \\ S_1 & 0 & -C_1 & a_1S_1 \\ 0 & 1 & 0 & d_1 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad T_1^2 = \left[\begin{array}{ccc:c} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad T_2^3 = \left[\begin{array}{ccc:c} S_3 & 0 & -C_3 & -a_3S_3 \\ -C_3 & 0 & -S_3 & a_3C_3 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad T_3^4 = \left[\begin{array}{ccc:c} C_4 & 0 & S_4 & a_4C_4 \\ S_4 & 0 & -C_4 & a_4S_4 \\ 0 & 1 & 0 & -d_4 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_4^5 = \left[\begin{array}{ccc:c} C_5 & 0 & S_5 & -a_5C_5 \\ S_5 & 0 & -C_5 & -a_5S_5 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad T_5^6 = \left[\begin{array}{ccc:c} =-S_6 & C_6 & 0 & 0 \\ C_6 & S_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]\left[\begin{array}{ccc:c} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} 0 & -S_6 & -C_6 & 0 \\ 0 & C_6 & -S_6 & 0 \\ 1 & 0 & 0 & d_6 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

(This to adjust the tool-tip frame to be parallel to the reference frame)

$$T_1^3 = T_1^2 T_2^3 = \left[\begin{array}{ccc|c} S_{23} & 0 & -C_{23} & a_2C_2 - a_3S_{23} \\ -C_{23} & 0 & -S_{23} & a_2S_2 + a_3C_{23} \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad T_4^6 = T_4^5 T_5^6 = \left[\begin{array}{ccc|c} S_5 & -C_5S_6 & -C_5C_6 & -a_5C_5 + d_6S_5 \\ -C_5 & -S_5S_6 & -S_5C_6 & -a_5S_5 - d_6C_5 \\ 0 & C_6 & -S_6 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_3^6 = T_3^4 T_4^6 = \left[\begin{array}{ccc|c} C_4S_5 & -C_4C_5S_6 + S_4C_6 & -C_4C_5C_6 - S_4S_6 & C_4(a_4 - a_5C_5 + d_6S_5) \\ S_4S_5 & -S_4C_5S_6 - C_4C_6 & -S_4C_5C_6 + C_4S_6 & S_4(a_4 - a_5C_5 + d_6S_5) \\ -C_5 & -S_5S_6 & -S_5C_6 & -d_4 - a_5S_5 - d_6C_5 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Problem statement

$$\left[\begin{array}{ccc|c} C_1S_{23}C_4S_5 + S_1S_4S_5 + C_1C_{23}C_5 & C_1S_{23}(S_4C_6 - C_4C_5S_6) - S_1(S_4C_5S_6 + C_4C_6) + C_1C_{23}S_5S_6 & -C_1S_{23}(S_4S_6 + C_4C_5C_6) - S_1(S_4C_5C_6 - C_4S_6) + C_1C_{23}S_5C_6 & C_1[S_{23}(-a_3 + a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2] + S_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\ \hline +S_1S_{23}C_4S_5 - C_1S_4S_5 + S_1C_{23}C_5 & S_1S_{23}(S_4C_6 - C_4C_5S_6) + C_1(S_4C_5S_6 + C_4C_6) + S_1C_{23}S_5S_6 & -S_1S_{23}(C_4C_5C_6 + S_4S_6) + C_1(S_4C_5C_6 - C_4S_6) + S_1C_{23}S_5C_6 & S_1[S_{23}(-a_3 + a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2] - C_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\ \hline -C_{23}C_4S_5 + S_{23}C_5 & -C_{23}(S_4C_6 - C_4C_5S_6) + S_{23}S_5S_6 & C_{23}(C_4C_5C_6 + S_4S_6) + S_{23}S_5C_6 & -C_{23}(-a_3 + a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + S_{23}(d_4 + a_5S_5 + d_6C_5) + d_1 + a_2S_2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & X \\ R_{21} & R_{22} & R_{23} & Y \\ R_{31} & R_{32} & R_{33} & Z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Some reduction to the main problem statement

$$
\left[\begin{array}{c|c|c|c}
\begin{array}{l} C_1S_{23}C_4S_5+S_1S_4S_5 \\ +C_1C_{23}C_5 \end{array} & \begin{array}{l} C_1S_{23}(S_4C_6-C_4C_5S_6) \\ -S_1(S_4C_5S_6+C_4C_6) \\ +C_1C_{23}S_5S_6 \end{array} & \begin{array}{l} -C_1S_{23}(S_4S_6+C_4C_5C_6) \\ -S_1(S_4C_5C_6-C_4S_6) \\ +C_1C_{23}S_5C_6 \end{array} & \begin{array}{l} C_1[-S_{23}(a_3-a_4C_4+a_5C_4C_5) \\ +C_{23}(d_4+a_5S_5)+a_2C_2+a_1] \\ +S_1(a_4S_4-a_5S_4C_5) \end{array} \\
\hline
\begin{array}{l} S_1S_{23}C_4S_5-C_1S_4S_5 \\ +S_1C_{23}C_5 \end{array} & \begin{array}{l} S_1S_{23}(S_4C_6-C_4C_5S_6) \\ +C_1(S_4C_5S_6+C_4C_6) \\ +S_1C_{23}S_5S_6 \end{array} & \begin{array}{l} -S_1S_{23}(C_4C_5C_6+S_4S_6) \\ +C_1(S_4C_5C_6-C_4S_6) \\ +S_1C_{23}S_5C_6 \end{array} & \begin{array}{l} S_1[-S_{23}(a_3-a_4C_4+a_5C_4C_5) \\ +C_{23}(d_4+a_5S_5)+a_2C_2+a_1] \\ -C_1(a_4S_4-a_5S_4C_5) \end{array} \\
\hline
\begin{array}{l} -C_{23}C_4S_5+S_{23}C_5 \end{array} & \begin{array}{l} -C_{23}(S_4C_6-C_4C_5S_6) \\ +S_{23}S_5S_6 \end{array} & \begin{array}{l} C_{23}(C_4C_5C_6+S_4S_6) \\ +S_{23}S_5C_6 \end{array} & \begin{array}{l} C_{23}(a_3-a_4C_4+a_5C_4C_5) \\ +S_{23}(d_4+a_5S_5)+a_2S_2+d_1 \end{array} \\
\hline
0 & 0 & 0 & 1
\end{array}\right]
=
\left[\begin{array}{ccc|c}
R_{11} & R_{12} & R_{13} & X-d_6R_{11} \\
R_{21} & R_{22} & R_{23} & Y-d_6R_{21} \\
R_{31} & R_{32} & R_{33} & Z-d_6R_{31} \\
\hline
0 & 0 & 0 & 1
\end{array}\right]
$$

$$
=
\left[\begin{array}{ccc|c}
R_{11} & R_{12} & R_{13} & \bar{X} \\
R_{21} & R_{22} & R_{23} & \bar{Y} \\
R_{31} & R_{32} & R_{33} & \bar{Z} \\
\hline
0 & 0 & 0 & 1
\end{array}\right]
$$

$$
\begin{bmatrix}
S_4S_5 & -(S_4C_5S_6+C_4C_6) & -(S_4C_5C_6-C_4S_6) & (a_4S_4-a_5S_4C_5) \\
S_1S_{23}C_4S_5-C_1S_4S_5+S_1C_{23}C_5 & S_1S_{23}(S_4C_6-C_4C_5S_6)+C_1(S_4C_5S_6+C_4C_6)+S_1C_{23}S_5S_6 & -S_1S_{23}(C_4C_5C_6+S_4S_6)+C_1(S_4C_5C_6-C_4S_6)+S_1C_{23}S_5C_6 & S_1S_{23}(a_4C_4-a_5C_4C_5)-C_1(a_4S_4-a_5S_4C_5)+S_1C_{23}(d_4+a_5S_5)+a_1S_1+a_2S_1C_2-a_3S_1S_{23} \\
-C_{23}C_4S_5+S_{23}C_5 & -C_{23}(S_4C_6-C_4C_5S_6)+S_{23}S_5S_6 & C_{23}(C_4C_5C_6+S_4S_6)+S_{23}S_5C_6 & -C_{23}(a_4C_4-a_5C_4C_5)+S_{23}(d_4+a_5S_5)+d_1+a_2S_2+a_3C_{23} \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
= \begin{bmatrix}
S_1R_{11}-C_1R_{21} & S_1R_{12}-C_1R_{22} & S_1R_{13}-C_1R_{23} & S_1\bar{X}-C_1\bar{Y} \\
R_{21} & R_{22} & R_{23} & \bar{Y} \\
R_{31} & R_{32} & R_{33} & \bar{Z} \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

For which

$\bar{X} = X - d_6R_{11}$ $\bar{Y} = Y - d_6R_{21}$ $\bar{Z} = Z - d_6R_{31}$

First:

Let us assume that $\theta_4 = 0$, so we can obtain θ_1 approximately as follows,

$$\tilde{\theta}_1 = \tan^{-1}\frac{\bar{Y}}{\bar{X}}$$

This gives estimation for the slew θ_1

Second:

From the above matrices, we have

$$S_4(a_4 - a_5C_5) = S_1\overline{X} - C_1\overline{Y}$$

$$S_4S_5 = S_1R_{11} - C_1R_{21}$$

By dividing to get rid of S_4 and rearranging

$$(S_1\overline{X} - C_1\overline{Y})S_5 + a_5(S_1R_{11} - C_1R_{21})C_5 = a_4(S_1R_{11} - C_1R_{21})$$

This can be rewritten as

$$\alpha S_5 + a_5\beta C_5 = a_4\beta \qquad \text{OR} \qquad \alpha S_5 = \beta(a_4 - a_5C_5)$$

where,

$$\alpha = S_1\overline{X} - C_1\overline{Y}$$

$$\beta = S_1R_{11} - C_1R_{21}$$

By squaring both sides and rearranging

$$(\alpha^2 + a_5^2\beta^2)C_5^2 - 2a_4a_5\beta^2 C_5 + (a_4^2\beta^2 - \alpha^2) = 0$$

OR (two solutions) (I will use the upper equation)

$$(\alpha^2 + a_5^2\beta^2)S_5^2 - 2a_4\alpha\beta S_5 + \beta^2(a_4^2 - a_5^2) = 0$$

The above equation will give two answers for C_5, then

$$S_5 = \frac{\beta(a_4 - a_5C_5)}{\alpha} \qquad \text{(two solutions)}$$

Then the two values of θ_5 can be obtained as follows,

$$\theta_5 = \tan^{-1}\frac{S_5}{C_5}$$ (two solutions)

Third:

From the above matrices, we have

$$-S_4C_5C_6 + C_4S_6 = S_1R_{13} - C_1R_{23}$$

$$-S_4C_5S_6 - C_4C_6 = S_1R_{12} - C_1R_{22}$$

By squaring and adding we can get rid of C_6 and S_6 as follows,

$$\boxed{S_4^2C_5^2 + C_4^2 = (S_1R_{13} - C_1R_{23})^2 + (S_1R_{12} - C_1R_{22})^2} \quad \text{(D1)}$$

Also from the above matrices, we have

$$\boxed{a_4S_4 - a_5S_4C_5 = S_1\bar{X} - C_1\bar{Y}} \quad \text{(D2)}$$

By rearranging equation (D2) as follows,

$$a_5S_4C_5 = a_4S_4 - (S_1\bar{X} - C_1\bar{Y})$$

$$= a_4S_4 - \alpha$$

giving that $\alpha = S_1\bar{X} - C_1\bar{Y}$

Therefore,

$$S_4^2C_5^2 = \frac{a_4^2S_4^2 + \alpha^2 - 2a_4\alpha S_4}{a_5^2}$$

By substituting in (D1)

$$\frac{a_4 S_4^2 + \alpha^2 - 2 a_4 \alpha S_4}{a_5^2} + C_4^2 = (S_1 R_{13} - C_1 R_{23})^2 + (S_1 R_{12} - C_1 R_{22})^2$$

giving that

$$= \gamma$$

$$\gamma = (S_1 R_{13} - C_1 R_{23})^2 + (S_1 R_{12} - C_1 R_{22})^2$$

Rearranging gives

$$a_4^2 S_4^2 + \alpha^2 - 2 a_4 \alpha S_4 + a_5^2 (1 - S_4^2) = a_5^2 \gamma$$

OR

$$(a_4^2 - a_5^2) S_4^2 - (2 a_4 \alpha) S_4 + (\alpha^2 + a_5^2 - a_5^2 \gamma) = 0$$

This is can be used to get θ_4 if θ_1 is known

Very important notice:

Since θ_5 has <u>two</u> different solutions and θ_4 has <u>four</u> different solutions; therefore the whole solutions has <u>four</u> different solutions. In other words:

The 1st solution in θ_5 should be combined with the 1st two solutions of θ_4 (They are θ_4 and $180 - \theta_4$).

<u>Fourth:</u>

From the above matrices, we have

$$(S_4 C_5) S_6 + (C_4) C_6 = C_1 R_{22} - S_1 R_{12}$$

$$(S_4 C_5) C_6 - (C_4) S_6 = C_1 R_{23} - S_1 R_{13}$$

Solving these two equation simultaneously gives

$$C_6 = \frac{C_4\phi + S_4C_5\psi}{S_4^2C_5^2 + C_4^2}$$

$$S_6 = \frac{S_4C_5\phi - C_4\psi}{S_4^2C_5^2 + C_4^2}$$

giving that

$$\phi = C_1R_{22} - S_1R_{12}$$

$$\psi = C_1R_{23} - S_1R_{13}$$

Fifth:

From the above matrices, we have

$$(S_5C_6)S_{23} + (C_4C_5C_6 + S_4S_6)C_{23} = R_{33}$$

$$(S_5S_6)S_{23} - (S_4C_6 - C_4C_5S_6)C_{23} = R_{32}$$

Solving these two equations simultaneously gives

$$C_{23} = \frac{(S_5S_6)R_{33} - (S_5C_6)R_{32}}{(C_4C_5C_6 + S_4S_6)(S_5S_6) + (S_4C_6 - C_4C_5S_6)(S_5C_6)}$$

$$S_{23} = \frac{(S_4C_6 - C_4C_5S_6)R_{33} + (C_4C_5C_6 + S_4S_6)R_{32}}{(C_4C_5C_6 + S_4S_6)(S_5S_6) + (S_4C_6 - C_4C_5S_6)(S_5C_6)}$$

$$S_1S_{23}(a_4C_4 - a_5C_4C_5) - C_1(a_4S_4 - a_5S_4C_5) + S_1C_{23}(d_4 + a_5S_5) + a_1S_1 + a_2S_1C_2 - a_3S_1S_{23} = \bar{Y}$$

$$-C_{23}(a_4C_4 - a_5C_4C_5) + S_{23}(d_4 + a_5S_5) + d_1 + a_2S_2 + a_3C_{23} = \bar{Z}$$

This can be rewritten as

$$a_2S_1C_2 = \bar{Y} - S_1[S_{23}(-a_3 + a_4C_4 - a_5C_4C_5) + C_{23}(d_4 + a_5S_5) + a_1] + C_1(a_4S_4 - a_5S_4C_5)$$

$$a_2S_2 = \bar{Z} + C_{23}(-a_3 + a_4C_4 - a_5C_4C_5) - S_{23}(d_4 + a_5S_5) - d_1$$

D.2 Numerical solution

D.2.1 Forward kinematics

$$T_0^1 = \left[\begin{array}{ccc:c} C_1 & 0 & S_1 & a_1C_1 \\ S_1 & 0 & -C_1 & a_1S_1 \\ 0 & 1 & 0 & d_1 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \quad T_1^2 = \left[\begin{array}{ccc:c} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \quad T_2^3 = \left[\begin{array}{ccc:c} S_3 & 0 & -C_3 & -a_3S_3 \\ -C_3 & 0 & -S_3 & a_3C_3 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_3^4 = \left[\begin{array}{ccc:c} C_4 & 0 & S_4 & a_4C_4 \\ S_4 & 0 & -C_4 & a_4S_4 \\ 0 & 1 & 0 & -d_4 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \quad T_4^5 = \left[\begin{array}{ccc:c} C_5 & 0 & S_5 & -a_5C_5 \\ S_5 & 0 & -C_5 & -a_5S_5 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_5^6 = \left[\begin{array}{ccc:c} =-S_6 & C_6 & 0 & 0 \\ C_6 & S_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]\left[\begin{array}{ccc:c} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} 0 & -S_6 & -C_6 & 0 \\ 0 & C_6 & -S_6 & 0 \\ 1 & 0 & 0 & d_6 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

The product of All six-link transforms takes the following form

$$T_0^6 = \left[\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad (1)$$

For which $r_{ij} \quad \forall 1 \le i, j \le 3$ represent the elements of the orientation matrix $\boldsymbol{R}_0^6$ and the vector $\boldsymbol{p}^T = \left[p_x \; p_y \; p_z\right]$ represents the position of the end-effector with respect to the base coordinate frame. In this regard,

$$
\begin{aligned}
r_{11} &= C_1S_{23}C_4S_5 + C_1C_{23}C_5 + S_1S_4S_5 \\
r_{21} &= S_1S_{23}C_4S_5 + S_1C_{23}C_5 - C_1S_4S_5 \\
r_{31} &= S_{23}C_5 - C_{23}C_4S_5 \\
r_{12} &= C_1S_{23}(S_4C_6 - C_4C_5S_6) - S_1(S_4C_5S_6 + C_4C_6) + C_1C_{23}S_5S_6 \\
r_{22} &= S_1S_{23}(S_4C_6 - C_4C_5S_6) + C_1(S_4C_5S_6 + C_4C_6) + S_1C_{23}S_5S_6 \\
r_{32} &= S_{23}S_5S_6 - C_{23}(S_4C_6 - C_4C_5S_6) \\
r_{13} &= -C_1S_{23}(S_4S_6 + C_4C_5C_6) - S_1(S_4C_5C_6 - C_4S_6) + C_1C_{23}S_5C_6 \\
r_{23} &= -S_1S_{23}(C_4C_5C_6 + S_4S_6) + C_1(S_4C_5C_6 - C_4S_6) + S_1C_{23}S_5C_6 \\
r_{33} &= C_{23}(C_4C_5C_6 + S_4S_6) + S_{23}S_5C_6 \\
p_x &= C_1[S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2 - a_3S_{23}] \\
&\quad + S_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\
p_y &= S_1[S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2 - a_3S_{23}] \\
&\quad - C_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\
p_z &= -C_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + S_{23}(d_4 + a_5S_5 + d_6C_5) + d_1 + a_2S_2 + a_3C_{23}
\end{aligned}
$$

The above equations specify how to compute the position and orientation of frame $\boldsymbol{O}_6$ at the end-effector with relative to frame $\boldsymbol{O}_o$ at the base of the arm

D.2.2 The general inverse kinematics problem

The inverse kinematics of a serial arm is more complex than its forward kinematics. However, many industrial applications don't need inverse kinematics algorithms, since the desired positions and orientations of their end-effectors are *manually taught*: a human operator steers the robot to its desired pose, by means of control signal to each individual actuator; the operator stores the sequence of corresponding *joint* positions into

the robot's memory; during subsequent task execution, the robot controller moves the robot to this set of the taught joint coordinates. However, the current trend towards on-line programming does require IK algorithms.

The general problem of inverse kinematics can be stated as: Given the numerical 4×4 homogeneous transformation matrix T_0^6, find the corresponding joint variables vector $\boldsymbol{q} = [q_1\ q_2\ q_3\ q_4\ q_5\ q_6]^T$ for which $q_i = \theta_i$ for revolute joints and $q_i = d_i$ for prismatic joints.

The inverse kinematic problem is not as simple as the forward kinematics. As shown, the equations, of course, much too difficult to solve directly in closed form. This is the case for most 6 dof robot arms. Therefore, it is required to develop efficient and systematic techniques that exploit the particular kinematic structure of the manipulator. Whereas the forward kinematics always has a unique solution that can be obtained simply by evaluating the forward equations, the inverse kinematics problem may or may not have a solution. Even if a solution exists, it may or may not be unique. Furthermore, because these forward kinematic equations are in general complicated nonlinear functions of the joints variables, the solution may be difficult to obtain even when they exist.

The solving of IK problem using closed form solution of the equations is preferable more than a numerical solution for the following reasons:

1- Finding a closed-form solution means finding an explicit relationship
$q_i = f(r_{ij}, p_x, p_y, p_z)\ \forall\quad 1 \le i, j \le 3$
2- In certain applications, such as tracking an object whose location is provided by a vision system, the inverse kinematic equations must be solved in a rapid rate, and having closed form expressions rather than an iterative search is a practical necessity.
3- The kinematic equations in general have multiple solutions. Having closed form solutions allows one to develop rules for choosing a particular solution among several.

However, sometimes it is difficult to achieve this analytical solution; therefore the alternative is using one of the available numerical techniques. There are several numerical methods for solving IK problems, coming originally from robotics applications, such as pseudo-inverse methods[1], Jacobian transpose methods[2], the Levenberg-Marquardt damped least squares methods[3]. Since the jacobian matrix of the arm under study is in

[1] D.E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Transaction on Man-Machine Systems, 10, (1969), pp. 47-53.
[2] W.A. Wolovich and H. Elliot, A computational technique for inverse kinematics, in Proc. 23rd IEEE Conference on Decision and Control, 1984, pp. 1359-1363.
[3] J. Zhao and N.I. Badler, Inverse kinematics positioning using nonlinear programming for highly articulated figures, ACM Transcations on Graphics, 13 (1994), pp. 313-336.

need for velocity control, the same matrix will be used for IK problem. However, the singularities arise from the approach is now being studied by the authors to prove the validity of the approach throughout the whole working space of the arm.

D.2.3 Jacobian matrix for Hydro-Lek manipulator

Jacobian matrix $\boldsymbol{J}(q)$ is the transformation from end-effector velocity vector $\left[\dot{\boldsymbol{p}}^T \;\; \boldsymbol{\omega}^T\right]^T$ to joints velocity vector $\dot{\boldsymbol{q}}$. Since, the generalized Cartesian velocity vector of the end-effector is composed of two sub-vectors $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$, the jacobian matrix may be partitioned into linear and orientation part by writing

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{J}(q)\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_p(q) \\ \boldsymbol{J}_\omega(q) \end{bmatrix} \dot{\boldsymbol{q}} \tag{D3}$$

For which $\dot{\boldsymbol{p}} = \left[\dot{p}_x \;\; \dot{p}_y \;\; \dot{p}_z\right]^T$ represents the resolved linear velocity of the end-effector and $\boldsymbol{\omega} = \left[\omega_x \;\; \omega_y \;\; \omega_z\right]^T$ represents the angular velocity. Here, the linear position jacobian $\boldsymbol{J}_p(q)$ represents the first three rows of $\boldsymbol{J}(q)$ and the angular jacobian $\boldsymbol{J}_\omega(q)$ its last three rows. Thus, the arm jacobian $\boldsymbol{J}(q)$ is $6 \times n$ matrix, with n the number of joints in the manipulator. In this case, $n = 6$, therefore the jacobian of the Hydro-Lek manipulator is square.

The computation of the linear position jacobian can be obtained using the following relation,

$$\boldsymbol{J}_p(q) = \left[\frac{\partial \boldsymbol{p}}{\partial q_1} \;\; \frac{\partial \boldsymbol{p}}{\partial q_2} \;\; \cdots \;\; \frac{\partial \boldsymbol{p}}{\partial q_n} \right] \tag{D4}$$

In order to evaluate the orientation part of the arm jacobian, consider the following chaining operation,

$$T_0^i = T_0^1 T_1^2 \ldots T_{i-1}^i = \left[\begin{array}{c|c} R_0^i & p_0^i \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \tag{D5}$$

Here R_0^i denotes the rotation matrix of frame i with respect to the base frame, for which

$$R_0^i = [x_i \ \ y_i \ \ z_i] \tag{D6}$$

The vectors x_i, y_i and z_i represent the x, y and z-axis of frame i in base coordinate.

In order to construct the orientation part of the jacobian part, this needs add the angular velocities of the joints, of course this requires representing them in the same coordinate frame, taking into account that the prismatic joints do not contribute to the angular velocity of the end-effector.

For a revolute joint, the joint rotation $q_i = \theta_i$ occurs about joint axis z_{i-1}. The angular velocity for joint variable i is therefore given by $z_{i-1}\dot{q}$. To add the effect of all the links, it is necessary to express z_{i-1} in a common frame which is the base frame. Therefore, the orientation part of the jacobian matrix may be written as

$$J_\omega(q) = [k_1 z_0 \ \ k_2 z_1 \ \ \ldots \ \ k_n z_{n-1}] \tag{D7}$$

where $k_i = 0$ if q_i is prismatic and 1 if q_i is revolute. Also the vector $z_0 = [0 \ \ 0 \ \ 1]^T$

Therefore, in the case of the Hydro-Lek ($q_i = \theta_i$) and $k_i = 1 \ \ \forall 1 \le i \le 6$, therefore the complete jacobian matrix can be obtained by stacking $J_p(\theta)$ and $J_\omega(\theta)$ as follows,

$$J(\theta) = \left[\begin{array}{cccccc} \frac{\partial p}{\partial \theta_1} & \frac{\partial p}{\partial \theta_2} & \frac{\partial p}{\partial \theta_3} & \frac{\partial p}{\partial \theta_4} & \frac{\partial p}{\partial \theta_5} & \frac{\partial p}{\partial \theta_6} \\ z_0 & z_1 & z_2 & z_3 & z_4 & z_5 \end{array}\right] \tag{D8}$$

where,

$$\frac{\partial \boldsymbol{p}}{\partial \theta_1} = \begin{bmatrix} C_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\ -S_1[S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2 - a_3S_{23}] \\ S_1(a_4S_4 - a_5S_4C_5 + d_6S_4S_5) \\ +C_1[S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_1 + a_2C_2 - a_3S_{23}] \\ 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_2} = \begin{bmatrix} C_1[C_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) - S_{23}(d_4 + a_5S_5 + d_6C_5) - a_2S_2 - a_3C_{23}] \\ S_1[C_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) - S_{23}(d_4 + a_5S_5 + d_6C_5) - a_2S_2 - a_3C_{23}] \\ S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) + a_2C_2 - a_3S_{23} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_3} = \begin{bmatrix} C_1[C_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) - S_{23}(d_4 + a_5S_5 + d_6C_5) - a_3C_{23}] \\ S_1[C_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) - S_{23}(d_4 + a_5S_5 + d_6C_5) - a_3C_{23}] \\ S_{23}(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) + C_{23}(d_4 + a_5S_5 + d_6C_5) - a_3S_{23} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_4} = \begin{bmatrix} C_1[S_{23}(-a_4S_4 + a_5S_4C_5 - d_6S_4S_5)] + S_1(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) \\ S_1[S_{23}(-a_4S_4 + a_5S_4C_5 - d_6S_4S_5)] - C_1(a_4C_4 - a_5C_4C_5 + d_6C_4S_5) \\ -C_{23}(-a_4S_4 + a_5S_4C_5 - d_6S_4S_5) \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_5} = \begin{bmatrix} C_1[S_{23}(a_5C_4S_5 + d_6C_4C_5) + C_{23}(a_5C_5 - d_6S_5)] + S_1(a_5S_4S_5 + d_6S_4C_5) \\ S_1[S_{23}(a_5C_4S_5 + d_6C_4C_5) + C_{23}(a_5C_5 - d_6S_5)] - C_1(a_5S_4S_5 + d_6S_4C_5) \\ -C_{23}(a_5C_4S_5 + d_6C_4C_5) + S_{23}(a_5C_5 - d_6S_5) \end{bmatrix}$$

$$\frac{\partial \boldsymbol{p}}{\partial \theta_6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Also

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad z_1 = \begin{bmatrix} S_1 \\ -C_1 \\ 0 \end{bmatrix} \qquad z_2 = \begin{bmatrix} S_1 \\ -C_1 \\ 0 \end{bmatrix} \qquad z_3 = \begin{bmatrix} -C_1C_{23} \\ -S_1C_{23} \\ -S_{23} \end{bmatrix}$$

$$z_4 = \begin{bmatrix} C_1S_{23}S_4 - S_1C_4 \\ S_1S_{23}S_4 + C_1C_4 \\ -C_{23}S_4 \end{bmatrix} \qquad z_5 = \begin{bmatrix} S_5(C_1S_{23}C_4 + S_1S_4) + C_1C_{23}C_5 \\ S_5(S_1S_{23}C_4 - C_1S_4) + S_1C_{23}C_5 \\ -C_{23}C_4S_5 + S_{23}C_5 \end{bmatrix}$$