Development and Implementation of a Telerobotic System with Video and Haptic Feedback

by

Johnathan Pretorius

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering Management at Stellenbosch University

Department of Industrial Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisor: Dr. A.F. van der Merwe

March 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: . . . .

Abstract

Development and Implementation of a Telerobotic System with Video and Haptic Feedback

J. Pretorius

Department of Industrial Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MScEng (Management)

March 2012

Telerobotics is a field of robotics interested in controlling robots from a distance. Incorporating visual and haptic (touch) feedback allows the operator greater accuracy in manipulating objects in a remote environment. This project endeavours to develop a telerobotic system by using two similar industrial robotic manipulators, one acting as a haptic input device (known as the Master), the other as the telerobot (known as the Slave).

This project makes use of a diverse range of technologies to achieve a telerobotic system. As such it makes use of Systems Engineering and Technology Management principles to guide the development process of converting standard industrial robots into a telerobotic system that allows the operator to "see" and "feel" in the remote environment.

The ability of the robot to perceive touch is accomplished by incorporating a multi-axis force and torque sensor to both industrial robots to measure any applied force or torque acting on each robot's end-effector.

A communication system is developed that receives the force and torque readings from each sensor as well as transmit the robot instructions to both Master and Slave.

Haptic feedback is achieved by combining the force and torque readings from both sensors within a control system before manipulating the data into a robot instruction that is used to control both Master and Slave.

Testing of the developed telerobotic system indicates that a functional system has been developed. However, several issues were identified that limit the control capabilities the operator has over the Slave and as such could lead to safety concerns.

Opsomming

Ontwikkeling en Implementering van 'n Telerobotiese Stelsel met Video en Haptiese Terugvoer

("Development and Implementation of a Telerobotic System with Video and Haptic Feedback")

J. Pretorius

Departement Bedryfs Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MScIng (Bestuur)

Maart 2012

Telerobotika is 'n veld van robotika wat belangstel in die beheer van robotte oor 'n afstand. Die insluiting van visuele en hapties (voel) terugvoer stel die operateur instaat om voorwerpe in 'n afgeleë omgewing met hoër akkuraatheid te manipuleer. Hierdie projek streef daarna om 'n telerobotiese stelsel te ontwikkel deur gebruik te maak van twee soortgelyke industriële robot manipuleerders, een wat optree as 'n meganiese invoer toestelle (bekend as die Meester), die ander as die telerobot (bekend as die Slaaf).

Hierdie projek maak gebruik van 'n verskeidenheid van tegnologieë om 'n telerobotiese stelsel te ontwikkel. Sodanig word tegnologie bestuur en stelsels ingenieurswese beginsels deur die loop van die projek geïmplementeer om die ontwikkelings proses te lei - van standaard industriële robotte tot 'n telerobotiese stelsel wat dit moontlik maak vir die operateur om te "sien" en "voel" in 'n afgeleë omgewing.

Die robot se vermoë om fisiese kontak te ervaar word bereik deur middel van 'n multi-as krag en torsie sensor wat aan beide industriële robotte toegepas word.

'n Kommunikasie stelsel is ontwikkel wat die krag en torsie lesings van beide sensors kan ontvang sowel as die robot instruksies na elke robot (Meester en Slaaf) kan stuur.

Haptiese terugvoer word bereik deur die krag en torsie lesings van beide sensors te kombineer in 'n beheer stelsel voordat dit verder gemanipuleer word in die form van 'n robot instruksie wat gebruik word vir die beheer van beide Meester en Slaaf.

Toetsing van die ontwikkelde telerobotiese stelsel dui aan dat 'n funksionele stelsel ontwikkel is. Daar is egter verskeie kwessies geïdentifiseer wat die operateur se beheer vermoeëns oor die Slaaf beperk wat sodoende na veiligheids kwessies kan lei.

Contents

Declaration **i**

Abstract **ii**

Opsomming **iii**

Contents **iv**

List of Figures **vii**

List of Tables **ix**

Glossary **x**

1 Introduction **1**
1.1 Motivation and Domain Application 1
1.2 Problem Statement 2
1.3 Research Objectives 3
1.4 Research Restrictions 3
1.5 Outline of this Thesis 4

2 Literature Study **5**
2.1 Industrial Robots 5
2.1.1 History of Industrial Robots 5
2.1.2 Industrial Robot Components 6
2.2 Telerobotic Systems 8
2.2.1 History of Telerobotics 8
2.2.2 Telerobotic Control Architectures 9
2.2.3 Current Applications 10
2.3 Haptics 11
2.3.1 Human Haptic Perception 12
2.3.2 Fundamentals of Haptic Interfaces 13
2.3.3 Haptic Implementations 14
2.4 Available Resources 14
2.4.1 Yaskawa Motoman Robots 15
2.4.2 Barix Barionet 17
2.4.3 Vivotek Network Camera 17

3 Systems Engineering Approach **18**
3.1 The Design Process 19

3.1.1 Project Definition and Planning 19
3.1.2 Specification Definition 20
3.1.3 Conceptual Design 21
3.1.4 Product Development 21
3.2 The Role of Technology Management 21

4 Development of a Telerobotic System 23
4.1 System Overview and Requirements 23
4.2 Incorporating Haptics 24
4.2.1 Sensor Requirements 25
4.2.2 Choosing the Appropriate Haptic Sensor 25
4.2.3 Description of the Net F/T Sensor 26
4.3 Communication System Development 27
4.3.1 Overview of the Communication System 27
4.3.2 Communication Architecture 28
4.3.3 Introducing the Interpreter 29
4.3.4 Barionet Communication Set-up 30
4.3.5 Java Communication Set-up 34
4.4 Control System Design 35
4.4.1 Command Input 35
4.4.2 Manipulated Variable 36
4.4.3 Designing the Controller 39
4.4.4 The Developed Control System 43
4.5 Combining the Interfacing Subsystems 43
4.5.1 The Physical Interface 44
4.5.2 The Software Interface 45
4.6 System Limitations and Considerations 47
4.6.1 Jittery Motion 47
4.6.2 Large Magnitude Inputs 48

5 Evaluation Methods 50
5.1 Accuracy, Repeatability and Equitability 50
5.2 Evaluation Standards 51
5.3 Adapting the Standards for Haptic Control 52
5.4 Pose Evaluation Procedures 52
5.4.1 Operator-Master Accuracy 54
5.4.2 Master-Slave Accuracy 55
5.4.3 Positional Repeatability 55
5.5 Haptic Evaluation Procedures 55
5.5.1 Steady State Procedure 56
5.5.2 Dynamic Procedure 56
5.5.3 Haptic Accuracy 57
5.5.4 Haptic Repeatability 58

6 Experimentation and Results 59
6.1 Pose Experimentation 59
6.1.1 Defining the Command Points 60
6.1.2 Pose Evaluation Equitability 60
6.1.3 Pose Accuracy and Repeatability 60

6.2 Force Experimentation . . . 63
6.2.1 Force Evaluation Equitability . . . 64
6.2.2 Steady State Results . . . 64
6.2.3 Dynamic Results . . . 65
6.2.4 Dynamic Limitations . . . 67

7 Conclusion and Recommendations 71

List of References 73

Appendices 79

A Motion Derivation 80
A.1 Equation of Linear Motion . . . 80
A.2 Equation of Angular Motion . . . 81

B Experimental Data for the Pose Evaluation 83

C Experimental Data for the Steady State Force Evaluation 85

D Experimental Data for the Dynamic Force Evaluation 87

E Force Component Results 89

F Technical Drawings 93

G t-Distribution Probability Table 97

H Motoman IMOV Command 99

I Device Manuals 102

List of Figures

1.1 Illustration of Problem Statement . . . 2

2.1 Industrial Robot Components . . . 7
2.2 The Zeus Telesurgery System . . . 10
2.3 The Da Vinci Telesurgery System . . . 11
2.4 Commercial Haptic Devices . . . 14
2.5 Motoman Industrial Robots with Controllers . . . 15
2.6 INFORM Syntax . . . 16
2.7 Barionet 100 . . . 17
2.8 Vivotek PZ715 . . . 17

3.1 Product Life Cycle . . . 18
3.2 Design Process Overview . . . 20

4.1 Telerobotic System Overview . . . 23
4.2 Net F/T Sensor . . . 26
4.3 Sensor Reference Frame . . . 26
4.4 Device Communication Overview . . . 28
4.5 Initial Communication Architecture . . . 28
4.6 Java Based Communication Architecture . . . 29
4.7 BCL Communication Example . . . 31
4.8 Motoman Handshaking Protocol . . . 32
4.9 BCL Handshaking protocol . . . 34
4.10 Generic Feedback Control System . . . 35
4.11 Sensor Orientation Effect . . . 37
4.12 Coordinate Systems . . . 38
4.13 FBD Illustration . . . 41
4.14 Position-based Haptic Control System for a Master-Slave Telerobotic System . 43
4.15 Mounting Plates . . . 44
4.16 Fully Assembled Haptic Interface . . . 44
4.17 Telerobotic System Process Flow . . . 45
4.18 Graphical User Interface . . . 46
4.19 Updated Control System . . . 49

5.1 Accuracy and Repeatability . . . 51
5.2 Physical Test Plane with Command Points and Paths . . . 53
5.3 Programming Pendant Displays . . . 54

6.1 Diagonal Test Plane Structure . . . 59
6.2 Operator Moving Input Manipulator . . . 61

6.3 Haptic Evaluation using Latex bands 63
6.4 Steady State Force Vector Results Graph 66
6.5 Dynamic Experimental Results Graph 67
6.6 Dynamic Experimental Results: Shifted Response 68
6.7 Limitation Example A 69
6.8 Limitation Example B 70

A.1 Force Torque Free Body Diagram 80

E.1 Steady State Fx Results Graph 89
E.2 Steady State Fy Results Graph 90
E.3 Steady State Fz Results Graph 90
E.4 Dynamic Fx Results Graph 91
E.5 Dynamic Fy Results Graph 91
E.6 Dynamic Fz Results Graph 92

F.1 Pencil Mounting Plate 94
F.2 UP6 Mounting Plate 95
F.3 SDA10D Mounting Plate 96

G.1 t-Distribution 98

List of Tables

2.1 Summary of Finger Force Strength and Variability . . . 13

4.1 Human Hand Perception . . . 25
4.2 Net F/T Sensing System Specifications . . . 27
4.3 Motoman Communication Requirements . . . 30
4.4 Transmission Control Characters . . . 33
4.5 Control System Terminology . . . 36
4.6 Motion Command Comparison . . . 37
4.7 Difference Between Active and Passive Control . . . 42
4.8 Combining Sensory Readings . . . 49

5.1 GUI Data Collection Format Example . . . 57

6.1 Command Point Coordinates . . . 60
6.2 Pose Accuracy Results: Operator-Master . . . 61
6.3 Pose Accuracy Results: Master-Slave . . . 62
6.4 Pose Repeatability Results - Master . . . 62
6.5 Pose Repeatability Results - Slave . . . 63
6.6 Minimum Number of Samples for Force Evaluation . . . 64
6.7 Steady State Force Results . . . 65
6.8 Dynamic Force Results . . . 66
6.9 Dynamic Results: Shifted Response . . . 68

B.1 Pose 1 Data . . . 83
B.2 Pose 2 Data . . . 83
B.3 Pose 3 Data . . . 84
B.4 Pose 4 Data . . . 84
B.5 Pose 5 Data . . . 84

C.1 Steady State Force Error Summary . . . 85
C.2 Steady State Force and Time Data Sample . . . 86

D.1 Dynamic Force Vector Error Summary . . . 87
D.2 Dynamic Force and Time Data Sample . . . 88

Glossary

BASIC	Beginner's All-purpose Symbolic Instruction Code. It is a general-purpose, easy to use, high-level programming languages.
BCL	Barix Control Language. The programming language used by the Barix Bariont device range. It is a high-level programming language with added support for networking applications
CAN	Controller Area Network. Communication protocol designed to allow micro-controllers and devices to communicate with each other without a host computer.
CGI	Common Gateway Interface. It is a internet based communication protocol used to send data to a server, where calculations are done, and to receive data, making the web content dynamic.
DCI	Data Communication through Instruction. It is a control function for saving and loading jobs to the Motoman controller from an external host controller.
DOF	Degrees Of Freedom. The number of independent parameters that define the configuration of a mechanical system.
FBD	Free Body Diagram. A graphical representation of forces acting on a body of interest.
GUI	Graphical User Interface. It is a type of user interface that allows users to interact with electronic devices with images rather than text commands.
HTML	Hyper text Markup Language. It is an annotation-based programming language for developing specifically web pages.
HTTP	Hyper Text Transfer Protocol. It is an application-level protocol for distributed, collaborative, hypermedia information systems.
IDE	Integrated Development Environment. It is a software application that provides comprehensive facilities to computer programmers for software development. It usually includes a source code editor, compiler or interpreter and a debugger.
IP	Internet Protocol. It is a protocol used for communicating data across a packet-switched network, such as the internet, using the Internet Protocol Suite.
ISO	International Organization for Standardization. It is an standard-setting body composed of representatives from various national standards organizations.
JND	Just Noticeable Difference. It is the smallest detectable difference between a starting and secondary level of a particular sensory stimulus, such as force.

JPEG	Joint Photographic Experts Group. It is a commonly used method of compression for digital photography (image).
LAN	Local Area Network. It is a computer network that interconnects computers in a limited area such as a home, school, computer laboratory, or office building.
MJPEG	Motion Joint Photographic Experts Group. It is a class of video formats where each video frame is separated as a JPEG image.
PoE	Power over Ethernet. It is a technology that enables a system to pass electrical power safely, along with data, on Ethernet cabling.
RS-232	Recommended Standard 232. It is a serial communication standard.
SDK	Software Development Kit. It is a set of software development tools that allows for the creation of applications for a certain software package.
SNMP	Simple Network Management Protocol. It is an Internet-standard protocol for managing devices on IP networks.
TCP	Transmission Control Protocol. It is a core members of the Internet Protocol Suite. With TCP computer applications can send reliable, ordered messages to a specific host on an IP network.
UDP	User Datagram Packet. It is a core members of the Internet Protocol Suite. With UDP computer applications can send messages to other hosts on an IP network without requiring prior communications to set up special transmission channels or data paths.

Chapter 1

Introduction

Telerobotics is a field of robotics interested in controlling robots from a distance. The ability to manipulate and inspect objects in a remote environment is extremely valuable, especially for tasks that pose a health risk to the human operator or require specific skills and knowledge to perform correctly [1]. The main issue in telerobotics, though, is preventing the loss of human perception when performing tasks in a remote environment. In the medical environment, the ability of a surgeon to see and feel when using a surgical tool such as a scalpel is extremely important to successfully perform surgery. Introducing a remote environment removes the surgeons ability to perceive the sense of sight and sense of touch. The challenge of telerobotic systems is, thus, to extend the human capabilities so they can perform a task in a remote environment without limiting their perception of the remote environment.

This thesis endeavours to develop a telerobotic system that allows an operator to use his or her hand in an intuitive manner to guide a telerobot in a remote environment while being able to see how the telerobot is moving as well as feel what the telerobot is "feeling". A typical procedure will involve that the operator controls a haptic input device (known as the Master) for manipulating a tool, such as a pen or scalpel, attached to the end-effector of the telerobot (known as the Slave) whilst monitoring the remote environment visually via video feedback on a computer screen.

1.1 Motivation and Domain Application

The automotive industry is currently one of the most influential customers for industrial robots [2]. These robots perform very well in tasks like spot-welding, press tending, spray painting, coating and other tasks where predictable environments are present. It is also known that their performance degrades significantly for tasks such as complex part assemblies, laser cutting, riveting and glueing due to a dynamic environment such as the variability in the product geometries etc. These tasks still require human labour to cope with the dynamic "random" environment [3].

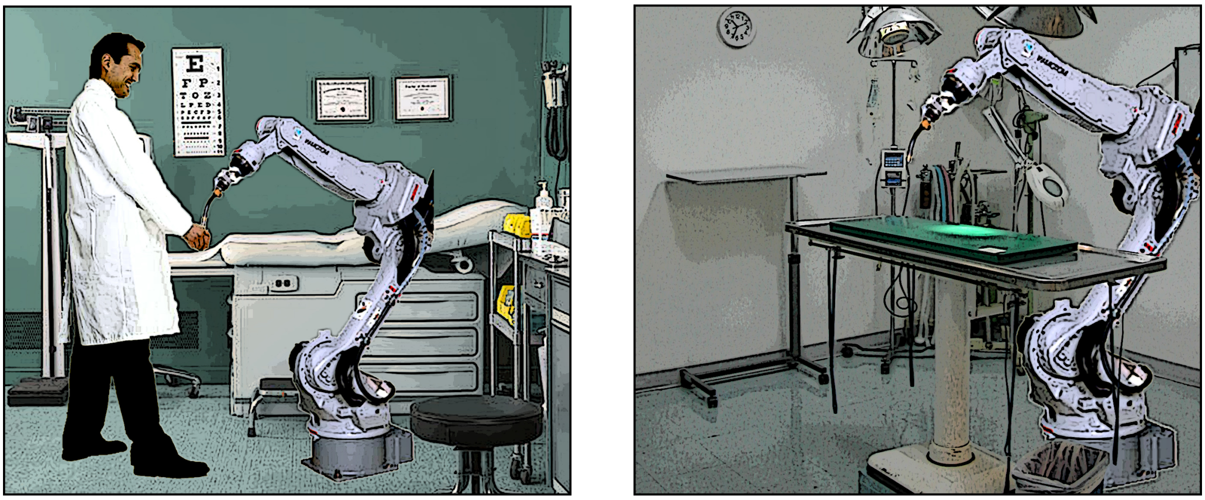
Telerobotic systems allow a human operator to control a telerobot in a remote environment. The inclusion of video and haptic feedback provide numerous advantages to both the operator and the operations being performed. The human operator obtains a better perception of the remote environment, allowing a greater dexterity in manipulating objects in the environment while introducing a computer into the control loop, allows for computer-assisted modifications [4]. The medical industry is a popular field where telerobotic research is actively being applied. Highly sophisticated, computer aided haptic devices (Master and Slave), have been developed, such as the da Vinci and Zeus surgical systems to aid surgeons in performing live surgery on a patient [5].

The first paragraph illustrates a very mature and well understood technology, industrial robots, with a clear limitation for applications requiring any change to working environment. The second paragraph illustrates a technology that is capable of high flexibility in dynamic environments but requires sophisticated devices to achieve.

What if industrial robots could replace the custom designed haptic technology to function as a telerobotic system? In doing so, these standard robot's capabilities would be extended into dynamic environmental applications, greatly increasing the available applications the robots can be used for while simultaneously providing telerobotic systems with alternate options for haptic devices.

1.2 Problem Statement

Consider the scenario illustrated in Figure 1.1. Two industrial robots are situated in different dynamic environments that do not typically involve repetitive tasks. An examples of a non repetitive task is inspecting random objects or performing a cutting operation on a random surface, such as making an incision with a scalpel in the medical industry. These dynamic environments prevent the operation of these robots from using the standard, preprogrammed instructions as currently utilised in industry.



(a) Operator Environment (b) Remote Environment

Figure 1.1: Illustration of Problem Statement

An operator, wishes to use these robots as a telerobotic system, allowing him to manually perform the tasks as he normally would in person without the need to travel from an to the remote environment to do so. The telerobotic system therefore needs to be intuitive and interactive to extend the operator abilities to the remote environment.

The operator would typically use his hands to perform the above mentioned tasks, he therefore needs a haptic input interface (Master) that can "capture" the action he is performing with his hand. These actions then need to be communicated to the industrial robot in the remote environment (Slave) that will perform the actual task. The operator is completely separate from the remote environment. Therefore, before he can perform a task, such as making an incision with scalpel, the operator needs to see where he wants to make the incision (visual feedback) as well as feel whether more or less effort is required to perform the cutting operation (haptic feedback). Note that such a cutting operation is time-sensitive, i.e. a long delay between when the operator receives the feedback and when it actually occurred could mean that he overshoots the target position or applied force. The delay therefore needs to be kept to a minimum to allow the operator to perform such tasks successfully.

This project aims to turn two standard industrial robots into a Master-Slave telerobotic system by fitting each with the necessary sensors and devices to allow an operator to intuitively perform a task in a dynamic remote environment.

1.3 Research Objectives

The objectives required for the completion of this research project will include the following:

- Transform two industrial robots into haptic devices. One that can be used as a operator input device (Master), the other as the telerobot (Slave).
- Develop a bilateral communication system between the operator and remote environment. This includes the protocols required for data transmission between the various devices in each environment.
- Design a feedback control system that can control both Master and Slave in an intuitive manner based on the operators hand movements.
- Develop a human interface system which combines visual and haptic devices into an operator friendly graphical interface.
- Analysing the accuracy and repeatability of the developed telerobotic system with regards to positioning and haptic control.

1.4 Research Restrictions

The research project includes specific resource restrictions that limit various aspects of the telerobotic system. These restrictions include the following:

- The project will make use of two standard industrial robots for the roles of the Master and Slave devices respectively. These robots are limited to a Motoman UP6 with XRC controller and a Motoman SDA10D with DX100 controller.
- The Motoman SDA10 with DX100 controller will only be available for use in the latter stages of the project.

- Communication between the operator and remote environment will follow an Ethernet communication protocol within the intra-network of the University of Stellenbosch.
- A Vivotek PZ7151 network enabled surveillance camera will provide the video feedback to the operator.

1.5 Outline of this Thesis

The core of the thesis is divided into seven chapters. This section provides a brief explanation on each of the chapters to follow.

Chapter 2. This chapter provides an overview of the literature followed in this thesis. It gives information on the topics of industrial robots, telerobotics and haptics. It also provides background information on the resources available for this thesis.

Chapter 3. This chapter describes the Systems Engineering approach that will be applied to guide the entire process as well as the role of technology management during the development, implementation and documentation process.

Chapter 4. This chapter discusses the development of the proposed telerobotic system and is divided into five sections. First it provides an overview of the requirements of a telerobotic system before discussing how to incorporate haptic control into two industrial robot manipulators. The chapter continues by developing a communication system, which allows the various devices to communicate between the two environments, via Ethernet and serial interfacing. The 4th section discusses the different effects that input and output requirements place on the control system before a control algorithm is formulated. The final section combines all these technologies into a single system by making use of concurrent processes to achieve real-time control and feedback.

Chapter 5. In this chapter the telerobotic evaluation procedures are discussed. A statistical analyses method is identified to determine the equitability of the experimental data to be used in the evaluations. Furthermore, a clear methodology for assessing the accuracy and repeatability of the system with specific regards to the positioning and haptic control capabilities of both the Master and Slave manipulators are presented.

Chapter 6. The chapter details the experimental procedure along with the achieved accuracy and repeatability results obtained by following the methodological approach provided in Chapter 4. These results are analysed and discussed in detail to provide a greater understanding of the capabilities and limitations of the developed telerobotic system.

Chapter 7. The final chapter summarizes the conclusions that can be drawn from the developed and tested telerobotic system.

Chapter 2

Literature Study

This chapter provides an overview of the literature followed in this thesis. It gives information on the topics of industrial robots, telerobotics and haptics. It also provides background information on the resources available for this thesis.

2.1 Industrial Robots

An industrial robot is an automatically controlled, reprogrammable, multi-purpose, manipulative machine with several degrees of freedom, which may be either fixed in place or mobile for use in industrial automation applications [6]. This definition, formulated by the International Organization for Standardization (ISO), can be interpreted according to Wallén [7] as follows:

- An industrial robot shall easily be reprogrammable without physically rebuilding the machine.
- It shall also have memory and logic to be able to work independently and automatically.
- Its mechanical structure shall be able to be used in several working tasks, without any larger mechanical operations of the structure.

The most commonly used industrial robots are articulated robots, SCARA robots, Cartesian coordinate robots and delta robots. An articulated robots features rotary joints and can range from simple two joint structures to 10 or more joints. The links in the arm are connected by rotary joints. Each joint is called an axis and provides an additional degree of freedom, or range of motion. Note that the robots available for this project fall under the articulated robots category [8]. Thus, the rest of this section will focus on articulated robots.

2.1.1 History of Industrial Robots

George Devol invented the first industrial robot, known as the Unimate, in 1954. After forming the first robot company, Unimation (in collaboration with Joseph Engelberger) in 1960, the Unimate was receiving interest from the automobile industry, specifically from General Motors. A year later, in 1961, the Unimate was incorporated into General Motors' assembly line for spot welding and extracting die casting [9].

Industrial robotic arms continued to evolve in the 1960's and 70's. The six-jointed Rancho Arm was developed to assist the handicapped in 1963. This was followed by the tentacle arm in 1968, designed by Marvin Minsky, which consisted of 12 joints and was able to lift a person. The year of 1969 brought on the Stanford Arm that eventually led to commercial arm production. The Stanford Arm was one of the first electronically powered, computer-controlled arms. By 1974, it reached a level of sophistication where it could assemble a Model T water pump [10].

In 1974, the first microcomputer controlled robot was introduced by Cincinnati Milacron, known as the T3 - The Tommorow Tool. This was closely followed by the Silver Arm in the same year. The Silver Arm was created by MIT's David Silver to perform precise assembly using touch and pressure sensors and a microcomputer. These two microcomputer controlled arms lead to Victor Scheinman, the inventor of the Stanford Arm, to focus on the creation of the PUMA (programmable universal manipulator for assembly)[7].

During the 1970's, the automotive and metal industry were the main customers when it came to industrial robots and as such the research focus was on improving material handling and arc welding capabilities. This meant improving the robots load capacity as well as control systems to manage path control. Early 80's the focus shifted to assembly, thereby industrial robots with greater repeatability, acceleration and velocity were needed in order to shorten the production cycle times[7].

Early in the 1980's industrial robots performed very well in limited areas of industrial tasks, specifically tasks that mainly involved synchronization and no task interaction, like arc-welding, paint spraying etc. It, however, became apparent that the current structure and control methods resulted in ineffective performance of tasks requiring interaction with the environment, such as product assembly [11] Research focus up to the current day therefore shifted to improving flexibility by incorporating external devices, such as computers and advanced sensors.

In special applications machine vision, laser scanners or force sensors were incorporated [12] and lead to the development innovative control systems such as Cartesian force control, impedance control and advanced motion planning to name but a few [13].

2.1.2 Industrial Robot Components

A typical industrial robot consists of an end-effector, an industrial robotic arm (manipulator) that consists of multiple joints, a controller and a programming pendant as shown in Figure 2.1.

An end-effector is a device designed for a specific task that attaches to the end of the robotic arm (manipulator) and acts as a “hand”. The end-effector comes in direct contact with the material or part being manipulated. Typical examples of end-effectors are various grippers for part handling, a welding torch for arc-welding applications, paint spray gun for spraying applications, machine tools for drilling (or milling) or a scalpel for surgical applications.

The robotic arm (manipulator) is the part which moves the end-effector to a specific position and/or orientation within the workspace of the robot. The robot arm can vary in size and shape depending on its designed application and load requirements. The arm consists of multiple joints each giving the industrial robot a degree of freedom (DOF). A simple robot with three degrees of freedom (3DOF) can move the end-effector in three ways - up & down (z-direction), left & right (y-direction) and forward & backward (x-direction) [8]. A 6DOF industrial robot, like the one shown in Figure 2.1, can move the end-effector in six ways - which consists of the same movements as the simple 3DOF robot with an additional 3 types to specify the orientation of the end-effector around the x, y or z axis respectively. The controller is the "brain" of the robot, allowing all the parts to operate

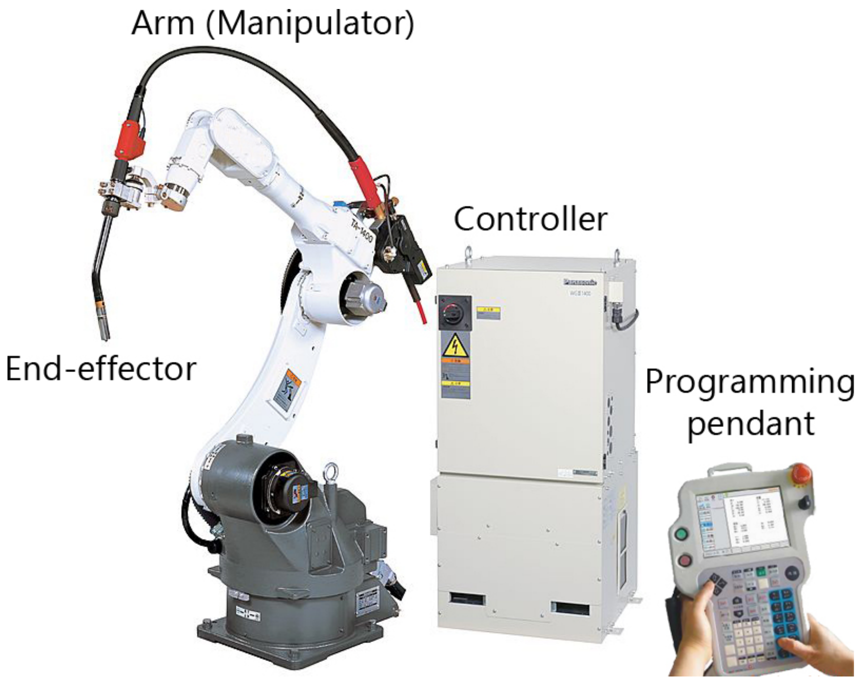


Figure 2.1: Industrial Robot Components

together. It contains a logic unit, processor, memory and communication devices. The logic unit and processor enables the controller to execute various instructions saved to the memory, known as "jobs" for controlling the robotic arm while the communication capabilities enable the robot to be connected to other systems [8]. A robot jobs can typically be programmed via the programming pendant (also known as a teach pendant) or by a method known as off-line programming (OLP).

The programming pendant is a hand-held controller which provides a means of manually manipulating the end-effector position by controlling the individual joints on the robotic arm. The joint orientations can be recorded within the teach pendant and by combining various joint coordinates into a sequential list, a robot instruction or job can be formulated that specifies the trajectory and speed with which the robot should move to each coordinate [14].

In OLP, the robot is mapped graphically on a computer program using an OLP software package such as RobotStudio, MotoSim, KUKA-Sim or 3D Studio [15]. The robot can be moved on a screen and the process simulated, without working on the actual robot.

These software packages create a program that contains a path for a particular simulated task. This program can then be downloaded into the controller.

2.2 Telerobotic Systems

Telerobotics systems allow human operators to accurately perceive a remote environment, enabling them to properly interact with a telerobot to manipulate objects located in that remote environment. It is a combination of two main sub fields, teleoperation and telepresence.

A general definition of teleoperation is performing work in a remote location without actually being in that location [16]. In this definition "work" refers to controlling a robot accurately, whilst "remote location" can be associated with either physical distance, whereby the operator is separated from the telerobot over a large distance, or a change in scale, where large adjustments on the operator side can be translated into minor adjustments on the telerobot side and vice versa (e.g. micro surgery).

The second major sub field, telepresence is concerned with creating a sense of physical presence at a remote location using the necessary multimedia such as sound, vision and touch [17]. The aim of a telepresence system is to excite the operator's senses by generating sensory stimuli so that the operator has the illusion of being present at the remote location.

The purpose, therefore, of telerobotics is to extend human actions to remote locations whilst providing the necessary perceptual feedback. Thus, allowing the operator to perform tasks in the remote environment with a similar dexterity as would be possible in person, whilst avoiding physical risk to the human operator [18].

2.2.1 History of Telerobotics

Telerobotics systems date back to the mid 1940's, where a means of safely handling radioactive materials in World War II were necessary. Raymond Goertz, known today as the pioneer in telerobotics, was the first person to successfully develop a human-robot telerobotic system for this purpose [19]. This telerobotic system originally made use of mechanical pantograph mechanisms but was later replaced by an electrical based servomechanism with force reflecting capabilities to achieve mechanical separation of the Master and Slave.

The telerobotic field, up to this stage, was mainly focused on the teleoperation aspects (performing work over a distance) and thus, still required the operator to be in eye contact with the hazardous material. Realizing this shortcoming, Goertz later introduced a telepresence system, in the form of closed-circuit television screens, which allowed the operator to be an arbitrary distance away [20]. Since then, the field of telerobotics has broadened with several noticeable advancements being made up to the early 90's.

These advancements include experiments in the early 1960's that provided insight into the effect of time delay in teleoperation [21]. Supervisory control was subsequently developed to address delay related issues [22]. This sparked a new field of research to develop teleoperation-oriented control languages [23; 24; 25] during the 1980's along with visual enhanced predictive displays [26; 27] to minimize overhead system's communication. Sheridan conducted an extensive survey in the late 1980's, documenting all of of the telerobotic advancements since 1950 in a single article for the journal of Automatika [28].

In 1994, teleoperation over a network became popular when a group of researchers of the University of Cambridge decided to monitor the local coffee filter machine, via a camera, to avoid lengthy and possibly fruitless trips to an empty coffee pot. The camera captured an image of the coffee pot every 20 seconds, which allowed the researchers to access an up-to-date image across the local network and determine the state of the pot without walking a considerable distance [29].

Despite various challenging issues introduced by packet switched networks (and later internet based telerobotics) such as random, time-varying transmission delays and delay jitter - resulting in a potential loss of data [19; 30] the field has seen tremendous growth with new applications in handling radioactive material [31], operating unmanned underwater vehicles [32], space robotics [33], telesurgery [34] and mobile robotics [35].

2.2.2 Telerobotic Control Architectures

The control system can make use of several different control structures, depending on the intended application, of which position, force, hybrid, impedance and admittance are the most popular [36].

The position control architecture [37] refers to controlling the positions between the robotic mechanism (Master) and the telerobot(Slave) to minimizing the positional error between the two. The control system, thus, compares the position of the Slave to that of the Master (with regard to their respective coordinate systems). If the positions differ, the position of the telerobot is adjusted to minimize the positional error.

The force control [38] architecture looks to minimize the resulting force error between the Master and Slave devices. The control system, thus, compares the force on the Slave to the force on the Master. If the forces differ, the position of the telerobot is adjusted to minimize the force error.

The hybrid control architecture [39] combines the previous two methods by trying to accomplish both but specifying a priority to one of the two in case of a conflict. This proposed method, therefore, requires the communication of both position and force information to the control system. The priority between position and force control depends on the applications and it is not uncommon for it to change several times during a specific operation.

Impedance control [40], aims to control neither position nor force but their dynamic relationship, known as the mechanical impedance of the manipulator. Impedance control makes use of a "motion" input, such as position and/or velocity to calculate the corresponding forces specific to a given system model. The telerobot is then controlled using the dynamic relationship to ensure that the end-effector "motion" error is minimized [41].

The last approach mentioned, admittance control [42] is the opposite of impedance control. Forces are measured, usually with a load cell, and are then sent to a computer. Calculations are performed to find the corresponding motion of the end-effector of the telerobot according to the equations of motion, and position control approaches are used to move the robot accordingly [41].

2.2.3 Current Applications

Telerobotics is a field that can be utilized in various applications, ranging from microscopic surgery to exploring the vast and infinite space. This section gives a brief overview on two applications utilised to great effect in the medical industry.

Telesurgery is a relatively new direction of telerobotics. It allows a physician to exchange medical expertise anywhere in the world without the need to travel. This effectively brings the remote surgery room to the surgeon's fingertips, thereby reducing effort, time and costs [19]. Two telesurgery systems very popular in the medical field are the "ZEUS" and "Da Vinci" surgical systems. Both systems are capable of performing general thoracic procedures and laparoscopic surgery, however, they follow different approaches to performing robotic surgery [5].

The ZEUS surgical system (shown in Figure 2.2) consists of three robotic manipulators attached to a surgical table in the remote environment, two of which are controlled by the operator's/surgeon's hands. The third robotic manipulator is fitted with a camera and is controlled via voice commands.

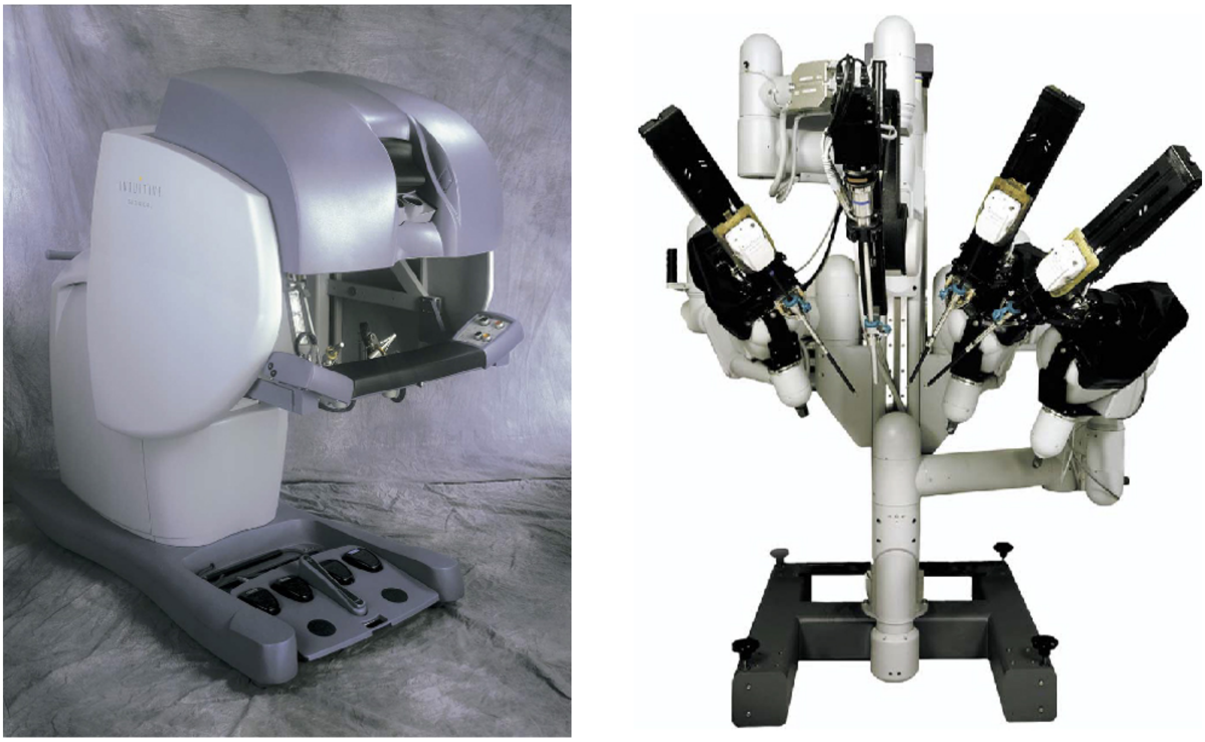
The Da Vinci surgical system (shown in Figure 2.3) comprises of essentially two main components - an operator console where the surgeon sits and a mobile cart that contains

(a) Operator interface (Master)　　(b) Telerobot (Slave)

Figure 2.2: The Zeus Telesurgery System [43]

four articulated robotic manipulators. One of the manipulators is fitted with a camera while the other three manipulators are used for performing surgery. The operator console consists of an image processing computer that generates a true 3-dimensional image with depth of field, the view port where the surgeon views the image, foot pedals to control electrocautery, camera focus, instrument and camera arm clutches, and master control grips that drive the articulated robotic arms at the patient's side.[43; 44].

The end effector of each telerobotic system allows for up to seven degrees of freedom of motion and since they are fundamentally managed by computers, several benefits are conferred to the systems to improve dexterity, including manipulating image quality to include 3D and stereoscopic capabilities, hand tremor filtering and motion scaling.



(a) Operator interface (Master) (b) Telerobot (Slave)

Figure 2.3: The Da Vinci Telesurgery System [43]

2.3 Haptics

Haptics refer to the human sense of touch and can be subdivided into two primary components, cutaneous and kinaesthetic touch [45; 46]. Cutaneous touch refers to the human tactile perception specifically regarding pressure experienced by the skin. It enables humans to detect vibration, surface roughness, skin stretch, skin curvature, etc. A typical example of tactile perception is the ability to distinguish between different surface textures. Kinaesthesia on the other hand, refers to the sense of force in the muscles and tendons. It provides awareness of the position and motion of the human body (static and dynamic) as well as larger scale details, such as basic object shape and mechanical properties, such as hardness of materials [47]. The sense of touch is, thus, invaluable when a sense of presence in a remote/virtual environment is required as it is one of the most informative senses that humans possess.

Unlike the other human senses (sight, sound and smell), touch is the only human sense that is a true bilateral sensation as touch receptors and activators share the same pathways - allowing humans to directly influence the amount of sensation being perceived. As such haptic input systems need actuators to provide haptic feedback as well. Haptic feedback, which is essentially force or cutaneous feedback in a man-machine interface, allows the remote/virtual environment to relay realistic, tangible sensations to an operator. With the incorporation of haptic feedback into virtual or remote environments, users have the ability to push, pull, feel, and manipulate objects in a remote environment via manual dexterity, rather than just seeing a representation on a video screen [41].

2.3.1 Human Haptic Perception

Human haptic perception is the process of acquiring, interpreting, selecting, and organizing haptic sensory information [41]. Acquiring haptic information is achieved via the cutaneous and kinaesthetic receptors on the human body. This information is sent via neural pathways to the brain for interpretation after which the appropriate response is sent to the human sensory motor control - which is responsible for all movement. Acquiring haptic sensory data and being able to respond to such information governs the design of a haptic device and as such requires knowledge of the human sensory system. Knowledge of the human haptic sensory system can be subdivided into four categories, sensing bandwidth, control bandwidth, force resolution and exertion capabilities. The rest of this section provides a quick overview on each of these categories, specifically with regard to human hand sensing and control capabilities.

The human sensing bandwidth refers to a range of frequencies within which humans can detect a change to haptic stimuli. The upper boundary or limit is the highest frequency at which a human can sense stimuli. Stimulants at a frequency above and beyond the upper boundary cannot be detected. The lower boundary, on the other hand is the lowest frequency at which stimuli can be sensed, which is logically fixed at 0Hz, indicating no stimulants. As such it is usually only required to specify the upper boundary to fully quantify haptic bandwidths. Literature regarding hand and wrist sensing bandwidths indicate that the maximum bandwidths for kinaesthetic sensing is in the order of 20 to 30Hz while cutaneous (tactile) sensing is as high as 320 to 400Hz [48; 49].

Similarly the control bandwidths refer to the frequency with which humans can respond to sensed stimuli (via the sensory motor control system). Telerobotic literature on control bandwidths provide contradictory results, however, a current consensus as to the significant difference in the results is that the human bandwidths are dependant on the situation [50]. Specifically, a human operator has different maximum bandwidths for unexpected signals (1 to 2Hz), versus periodic signals (2 to 5Hz), versus internally-generated or learned trajectories (up to 5Hz), versus reflexive actions (up to 10Hz).

Unlike the sensing and control bandwidths, force resolution and exertion capabilities concerns the magnitude of forces that can be applied. Where force resolution focuses on the smallest incremental force that can be distinguished by a human hand/wrist, whilst exertion capabilities aim to measure the maximum forces that can be applied by the human hand [51]. Force resolution is measured at a value ranging between 0.025 to 0.05N (Newtons)[52], however while this is the smallest force value that the average person can detect, it does not mean that a human can discriminate between two forces varying by 0.025N. This has led to research regarding JND (just noticeable difference) for various input forces [53].

The human hand can perform various tasks such as pulling, pushing, pinching, grasping, etc. The maximum force exertion, therefore differs depending on the action being performed as well as the orientation of the hand while performing the action. As such, several studies have been conducted in literature, each with specific set-ups and under certain conditions to determine various hand and finger characteristics [54; 49; 55]. Astin [51] investigated finger force capabilities for a number of different hand intensive tasks such as pushing buttons, sliding levers, inserting fasterners etc. Experiments were conducted on various subjects that simulate these tasks, including poking, puling, pressing, grasping and pinching. Table 2.1 provides a summary on his findings relating to motion induced tasks i.e. not including pinching and grasping.

Table 2.1: Summary of Finger Force strength and variability [51]

Gender	Measure	Poke	Press	Pull
Male	Mean (N)	52.58	50.90	70.84
	Standard Deviation (N)	18.01	18.37	27.16
	Coeff. Of Variation (%)	34.25	36.08	38.34
Female	Mean (N)	39.31	35.20	49.33
	Standard Deviation (N)	14.94	14.93	17.71
	Coeff. Of Variation (%)	38.00	42.42	35.91

2.3.2 Fundamentals of Haptic Interfaces

A haptic interface consists of a robotic mechanism along with sensors to determine the human operator's motion and actuators to apply a force to the operator. This physical mechanism couples the operator to the remote (or virtual) environment and can take the form of a common computer gaming joystick [56], a multiple-degree-of-freedom stylus [57], a wearable exoskeleton device [58], or an array of tactors that directly stimulate the skin surface [59].

The robotic mechanism defines the motion capabilities of the human operator when interacting with the device; the sensors allow the operators motions to be tracked and conveyed to the remote/virtual environment as well as collect perceptual data from the remote/virtual environment. The sensory data is then fed into a control system, governing the motion capabilities of the various actuators on the robotic mechanism which in turn provides haptic (force or tactile) feedback to the operator.

Haptic feedback can take one of two forms (active or passive) and can be differentiated by whether or not energy is added to the system [60]. Active force feedback controllers apply a force directly to the operator by adding energy into the haptic input interface (Master). This means that active feedback can control the motion on the Master manipulator and in doing so relay the force experienced in the remote environment and apply it to the operator. An advantage of using active force feedback control is that it is inherently general and any force sensation can be produced.

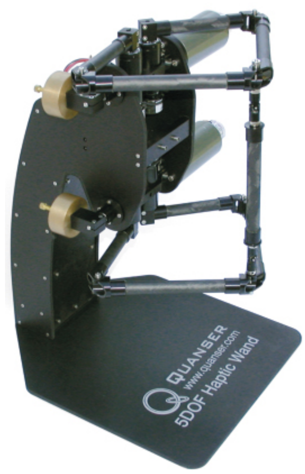
Passive force feedback, on the other hand, applies forces to the operator by removing energy from the system, using energy dissipation techniques (such as a brake). The passive based approach does not apply a force to the operator but rather a resistance to the operator's motion. The advantage of using passive force feedback control is that it is more stable and inherently safe for the operator, since energy dissipation elements only resist motion and not induce motion [61].

2.3.3 Haptic Implementations

Haptic devices enable the human sense of touch to be digitalized. It opens up the possibility of manipulating the data into any shape or form on a computer allowing it to to be implemented on a wide variety of applications, ranging from simple single-DOF devices for research [62] to complex, multi-DOF wearable devices [58]. Haptic devices, such as the Phantom desktop [57], Delta interface [63] and Haptic wand [64], shown in Figure 2.4, are some of the more popular haptic devices used for haptic implementations. For instance, these devices are used as an operator interface for telerobotic systems [65], education and training, particularly surgical training [66], visual arts [67], augmentation of graphical user interfaces [68] and enhancement of nano-manipulation systems [69].

(a) Phantom Haptic [57]

(b) Delta Haptic [70]



(c) Haptic Wand [64]

Figure 2.4: Commercial Haptic Devices

2.4 Available Resources

This section provides an overview of the resources available for use in this project. These resources form the basis of the telerobotic system, as stated in Section 1.4, with all subsequent device considerations made aimed at transforming the resources described in the rest of this section into a telerobotic system.

2.4.1 Yaskawa Motoman Robots

The Motoman UP6, Figure 2.5a and Motoman SDA10D, Figure 2.5d are commercially available industrial robotic manipulators developed by Yaskawa Electric Corporation. Each robot consists of the basic robot components described in Section 2.1.2 with the Motoman SDA10D fitted with two robotic arms (manipulators).

(a) Motoman UP6 (b) XRC controller (c) DX100 Controller (d) Motoman SDA10D

Figure 2.5: Motoman Industrial Robots with Controllers - adapted from [71; 72; 73; 74]

The Motoman UP6 is a six degree of freedom robot (6DOF), has a repetitive positional accuracy (precision) of ±0.08mm and is rated to handle a 6kg payload [71]. Each arm of the Motoman SDA10D allows for seven degrees of freedom (7DOF), has a repetitive positional accuracy (precision) of ±0.1mm and is rated to handle a 10kg payload [72]. The rated maximum payload is the amount of weight, attached to the end-effector, that the robot can carry while maintaining the rated precision.

Control of the robotic manipulators is accomplished via their respective controllers. The controllers are based on a robot control language known as INFORM II (XRC controller) and INFORM III (DX100 controller) for programming the robot to perform specific jobs. INFORM II and INFORM III uses the same command syntax and control functions, with INFORM III providing additional functionalities [74].. The INFORM syntax is illustrated in Figure 2.6 and is composed of the instruction and an additional item (tag and numeric data):

Instruction: It is used to execute the operation and processing. In the case of a move instruction, when a position is taught, the move instruction is automatically displayed according to the interpolation method.

Additional item: The speed, time, etc. are set according to the type of instruction. Numeric data and character data are added to the tag that specifies the condition as necessary.

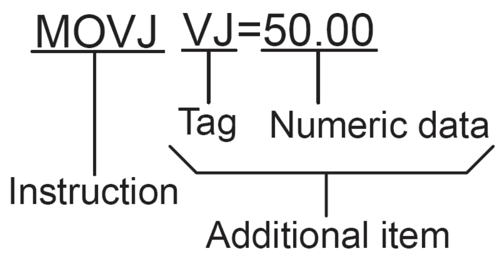


Figure 2.6: INFORM Syntax [74]

Both controllers provide the following control modes:

- Stand alone function
- DCI function
- Host control function

The stand alone function makes use of the programming pendant (hand held controller). The operator controls the Motoman by controlling each servo motor independently via buttons on the pendant. A robot job can be created by specifying the instruction, tag and numeric data on the programming pendant.

The data communication by instruction (DCI) function enables an external host controller (personal computer) to save and load jobs to the controller. This function is used by off-line programming (OLP) software packages such as Motosim for creating or modifying a job on a personal computer and then transferring it to the Motoman controller[15]. The DCI method requires that the jobs be saved on the Motoman controller before it can be loaded to execute a task.

The final method, host control functions, enable a host controller (personal computer) to send a temporary job to the Motoman controller. Unlike the DCI function, the job is not saved on the Motoman controller, it is immediately processed and executed and therefore cannot be recalled at a later stage [73].

The XRC and DX100 controllers both provide a serial (RS232) interface for communicating with an external host controller. In addition, the DX100 controller is also equipped with an Ethernet interface, for communicating with a external host controller over a network [74]. For detailed specifications regarding the Motoman robotic systems, refer to the product data sheets provided in Appendix I.

2.4.2 Barix Barionet

The Barionet (Figure 2.7) is a fully programmable network-enabled controller for interfacing various devices to IP-based networks. With the Barionet, most devices can be network-enabled to monitor and control via a web browser as well as other standard-based automation systems, such as SNMP and Modbus [75]. The Barionet 100 range is available for this project and is equipped with various modular interfacing options, including digital, analogue and serial interfaces and can therefore be used in a large variety of applications.

The Barionet 100 is equipped with a web server capable of communicating with a host (or Internet) via a 10/100MBit Ethernet connection. The web server hosts a website for easy access to the Barionet's configuration. The website is fully customizable by uploading user-created Hyper Text Markup Language (HTML) documents to the barionet. The web server can change variables within the barionet's memory by executing hyper text transfer protocol (HTTP) packages that use common gateway interface (CGI) commands.

The Barionet has its own high level, interpreted control language used to program the barionet, called Barix Control Language (BCL). The BCL syntax is very similar to the well known BASIC language, with various enhancements specifically for network access (such as UDP, TCP and CGI) as well as Input/Output control [76]. Customized BCL applications can be programmed to use any or all of the interfacing options available on the Barionet.

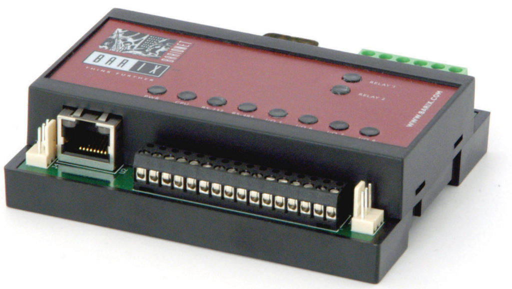


Figure 2.7: Barionet 100 [75]

Figure 2.8: Vivotek PZ715 [77]

2.4.3 Vivotek Network Camera

The Vivotek PZ7151 (Figure 2.8) is a high-performance network camera for indoor surveillance applications such as retail stores, offices or bank security [77]. Built into the camera is a 2.6x motorized pan-focus zoom module and along with the 350-degree horizontal and 125-degree vertical range of capture, it effectively gives users a wide-area birds's view. Like the Barionet, it is also equipped with a web server hosting a website from which the video feedback can be viewed. Other functions such as camera set-up and configuration can also be altered via the website.

Chapter 3

Systems Engineering Approach

To design a quality product, its entire life needs to be taken into consideration. Figure 3.1 depicts a generic life history of a product consisting of 4 phases.

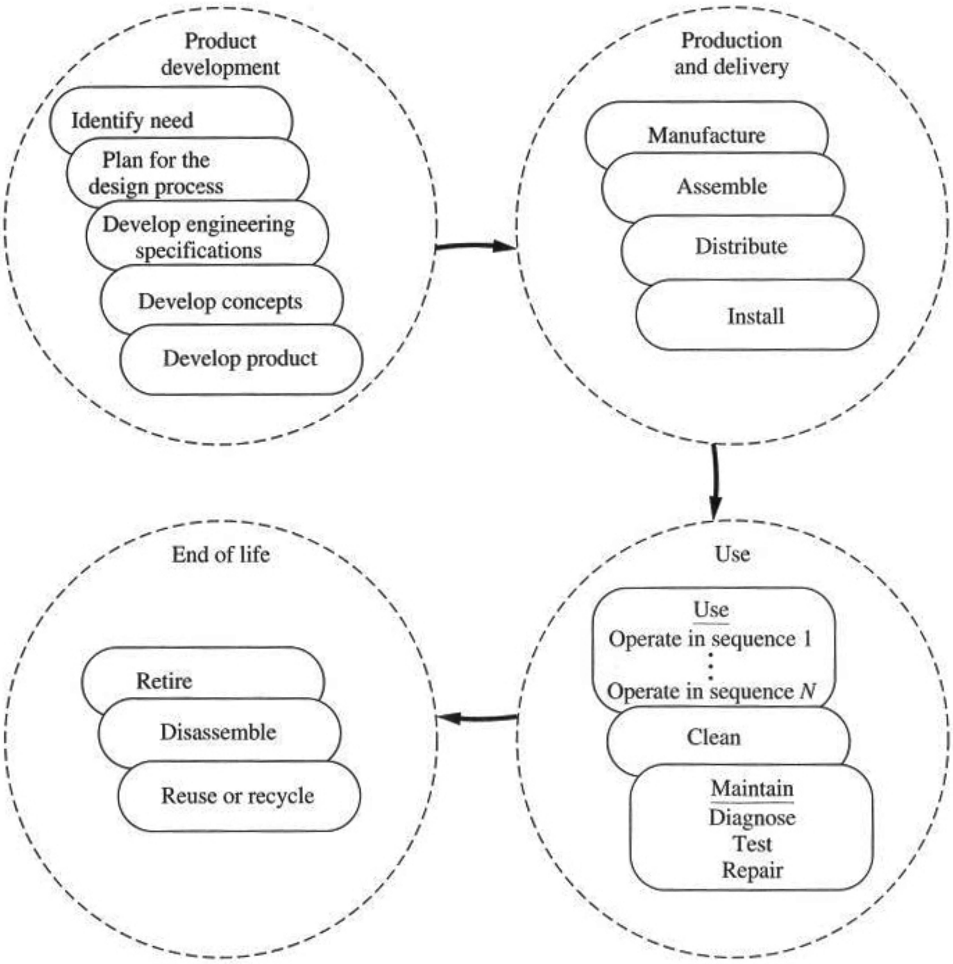


Figure 3.1: Product Life Cycle [78]

The first phase looks at identifying a need as to why such a system or product is beneficial, planning the design process and executing the plan to develop a product or system. This product development phase is crucial as is needs to take into account what will happen in the remainder of the product's lifetime.

The second phase is concerned with production and delivery - some products are just assemblies of existing components and do not require manufacturing of new unique components while distribution and installation can influence the final design of a product to allow for more productive packaging or easier installation.

The third phase is where a project gets used, which involves providing the operator with enough information on how to operate, maintain and repair the system.

The final phase of the products life is concerned with what happens to the product at the end of its life [78].

The work done in this thesis forms the first attempt at applied telerobotic research at the University of Stellenbosch and due to time and resource constraints, discussed in Section 1.4, it is not feasible to include a full life cycle design in this report. As such, this thesis will focus on the initial stages of the life cycle to develop a functional system and evaluate its performance, which will provide valuable insight that will aid future telerobotic research projects at the university.

3.1 The Design Process

Figure 3.2 illustrates a generic mechanical design process. The methodology described in this process is an ideal flow chart of activities and will be used as a framework for developing a telerobotic system with visual and haptic feedback. Note that the process includes aspects from the 2nd, 3rd and 4th product life cycle which does not form part of the scope of this research project and is therefore not covered in this report. The rest of this section describes the methodology that will be followed in developing a telerobotic system.

3.1.1 Project Definition and Planning

The first phase in the mechanical design process is to define the project. As with any new undertaking, the process is driven by an idea to accomplish a task that will eventually add value by introducing a new product or process or improving an existing process. This project started with an idea to control two physically separate industrial robots in an intuitive manner by hand where the remote robot would mimic the input applied by the operator on the other robot.

The project definition and planning phase therefore takes this idea and quantifies it into a more structured set of deliverables or objectives that creates a clearer picture of what resources will be required to realise the idea. This phase is documented in Section 1.2 to Section 1.4 by describing the idea as a problem that needs to be solved. The problem statement subsequently identifies the resource constraints along with the objectives that need to be met.

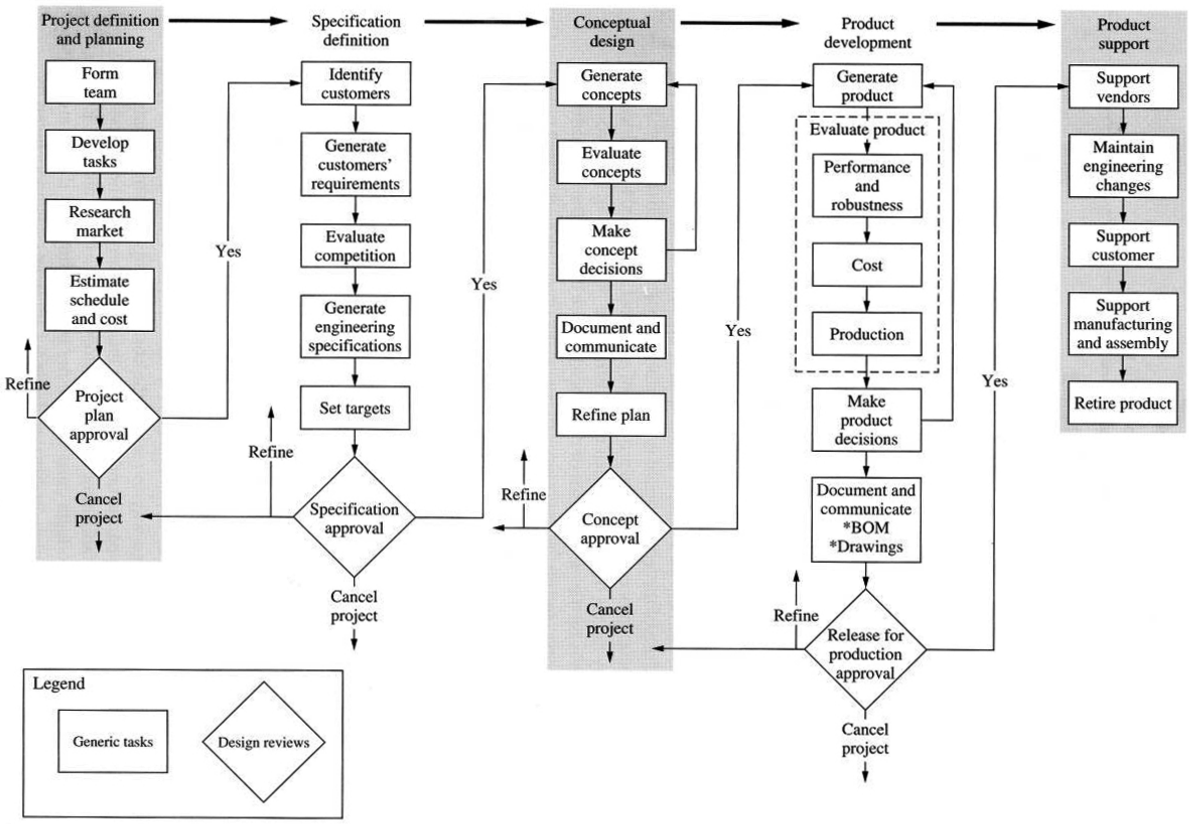


Figure 3.2: Design Process Overview [78]

3.1.2 Specification Definition

The goal of the engineering specification definition phase is to understand the problem further to lay a solid foundation for the development of the system. This phase involves generating the engineering requirements based on the "customer's" needs and the competition capabilities [78].

A literature study is conducted in Section 2 to provide insight into industrial robots, telerobotics, haptics and how such systems currently function in industry. It creates clarity on how similar systems operate as well as identifies the available resources capabilities from which to develop a functional telerobotic system. As a research project, the customer requirements are not fixed on specific performance requirements that need to be met, such as controlling a robot within a certain degree of accuracy or speed, but rather laid out as a set of guidelines that the developed system needs to adhere to. These guidelines are:

1. The telerobotic system is aimed towards basic telesurgical procedures. Meaning that an operator needs to control a robot in a remote environment dexterously by hand.
2. The motion (magnitude and direction) applied by the operator must be transferred and executed by the telerobot as closely as possible.
3. The perceived motion (magnitude and direction) due to the remote environment acting on the telerobot must be transferred to the operator as closely as possible

4. Points 2 and 3 above must occur as close to possible in real-time.

These guidelines need to be followed by incorporating the resource restrictions mentioned in Section 1.4 while the findings and performance of the developed telerobotic system must be documented to aid future research within the field.

3.1.3 Conceptual Design

Generating concepts for a product or system before development starts leads to higher quality products or systems being developed [78]. It ensures that the best alternative is chosen that needs the least expenditure of time and other resources to develop a functional model of the system, and that adheres to the specifications or guidelines of the previous phase. The time and resource restrictions discussed in Section 1.4, however, limit the number of concepts that can be developed substantially. Therefore, only a few concepts are considered and evaluated in this report to obtain the final design.

Note that this project will initially follow a linear development process - meaning that once a concept is decided on for the final design, all subsequent concept considerations are based on it. This approach is used primarily to ensure that the development and evaluation of a telerobotic system can be completed within the enforced time-constraints of the project. The initial concepts along with the improvements will therefore be discussed in this report in Section 4.

3.1.4 Product Development

After generating and evaluating the various concepts, the next phase is to combine all the concepts into a telerobotic system for evaluation. Unlike the design process of Figure 3.2, the evaluation in this project will be limited to performance. Performance measures the degree to which the initial requirements have been met. Section 5 will focus on the methods used to evaluate performance. The evaluation results and performance findings will be discussed in Section 6.

Note that the scope of the project is up to (and including) a performance evaluation of the developed system. This is, however, not the end of the design life cycle. The system that will be developed therefore does not denote the final "product". This research project will therefore serve as a foundation for future research project aimed at improving the work done in this project while extending the scope to other aspects in a design process.

3.2 The Role of Technology Management

This project aims to achieve an end objective of a functional telerobotic system using two standard industrial robots, however, the conceptualization, design, development, and the creation process of turning this idea into a working product can be loaded with obstacles [79]. Especially when considering that this idea requires a diverse range of objectives (as noted in Section 1.3) from various technological fields, including engineering science, modelling, statistics, database technology, networking and communication. It is therefore important to manage these technologies to achieve the end goal of a functional telerobotic system.

Technology management can be defined as a process, which includes planning, directing, control and coordination of the development and implementation of technological capabilities to shape and accomplish the objectives of an organization [80]. This definition can be applied to this research project by planning the development phase with the capabilities of the available resources in mind. This ensures that the technology can be shaped to accomplish the individual goals of each sub system. As obstacles arise, new technological advancements will be identified to manage these limitations. Technology management therefore provides a means of directing the development, implementation and documentation process to ensure that all research objectives can be achieved, resulting in a functioning telerobotic system.

Chapter 4

Development of a Telerobotic System

This chapter details the process of developing a fully functional telerobotic system, starting with a broad view of what constitutes a telerobotic system before delving deeper into each of its core components. These components are then developed individually before they are combined to construct a telerobotic system capable of haptic and visual feedback.

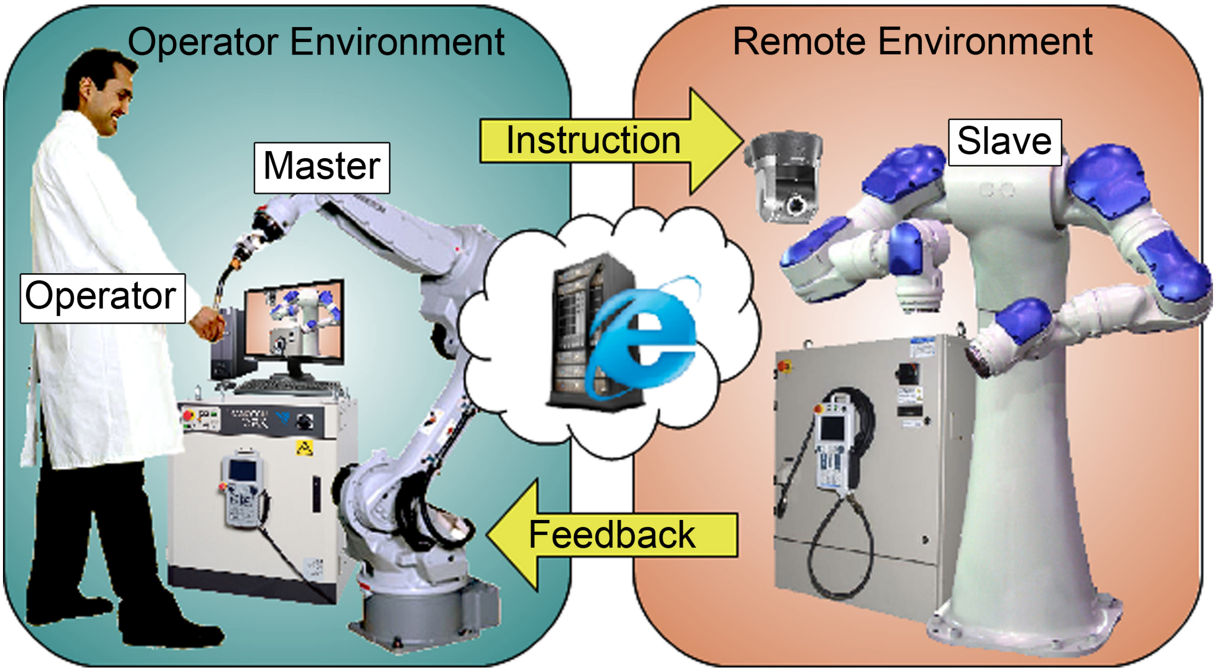


Figure 4.1: Telerobotic System Overview

4.1 System Overview and Requirements

Consider the telerobotic system illustrated in Figure 4.1. A telerobotic system constitutes two separate environments, an operator environment and a remote environment. The remote environment contains the telerobot to be controlled, along with the various sensors and devices required to provide visual and haptic feedback to the operator. The operator environment therefore requires devices capable of conveying such visual and haptic feedback in such a way as to excite the operator's sense of sight and touch respectively.

Furthermore, the operator environment requires a method to send accurate control instructions to the telerobot based on the feedback from the remote environment.

From this illustration the primary components of a telerobotic system become apparent. Firstly, two haptic interfaces are required. These interfaces will essentially be the Motoman UP6 and SDA10D industrial robots, mentioned in Section 2.4.1, adapted to include haptic measuring capabilities. Secondly, the system requires a control algorithm for manipulating the haptic data to control data required by the Motoman controllers. Thirdly, a communication system capable of communicating between the various devices in both the operator and remote environments is required. Finally, the haptic interfaces, control system and communication system needs to be combined into a telerobotic system.

4.2 Incorporating Haptics

In general the word "haptic" refers to the sense of touch[1]. To achieve haptic control in a remote environment, the operator needs to apply a "haptic" input to a device which can then transmit this input to the remote environment. The challenge though with haptic control is that, for it to be effective, the operator requires haptic feedback from the remote environment as well. This section details the process of converting the available Motoman industrial robots into haptic control devices capable of both sending and receiving haptic signals. The first step in identifying the appropriate approach is, thus, to take a look at the capabilities and limitations of the Motoman robots. Note that the two available robots, the UP6 and SDA10D are very similar robotic manipulators, with the "newer" SDA10D supporting all the functionalities provided by the UP6 along with newer technology such as Ethernet communication capabilities. To keep the design approach as modular as possible, all design decisions will be based on the functionality of the older UP6 robot.

Both Motoman manipulators are equipped with several encoders located at each actuated axis. The controller makes use of the values of these encoders to calculate the robot end-effector position in either Cartesian or joint coordinates. Similarly, control of the manipulator requires that the operator inputs a coordinate value to where the manipulator should move. The controller then utilizes built-in functions to determine the desired encoder value for each of the robot joints, before commanding the individual servo motors to rotate until the corresponding encoder value match the desired value. In general, regardless of the specific haptic device used, either position, force, velocity or a combination of these are used as an input to the control system [19]. The motion control functions of the Motoman robots require position and velocity based commands. Utilizing position and velocity as inputs into the control system therefore seems logical considering that the input and output of the control system would then be in unison. The control system would then only need to focus on the control aspects without the need to convert it the inputs into a usable form.

Note that the actuators of the robotic arm are locked in place when it is not executing a job specified by the controller, this means that the end-effector's position remains fixed. As such the robot arm can be modelled as a rigid system for haptic input purposes. This

[1]For an overview on haptics, along with several references to more detailed discussions, see Section 2.3

makes it impossible to use position or velocity directly as an input without the use of an additional mechanical spring-damper interface. Thus, for a rigid system, a force input is ideal as it only requires a force sensor (or combination of force sensors) to be fitted to the manipulator.

4.2.1 Sensor Requirements

A large variety of devices for measuring force are available in literature, including load cells, torque sensors, pressure sensors, etc. Before a choice can be made on what device to use, it is important to consider its application. Surgical applications require that the operator (or surgeon) uses fine, sensitive hand movements to perform any surgical procedure. A telerobotic system aimed towards telesurgical applications therefore requires that the input device be robust enough to detect the full range of force applied by a surgeon during a typical operation, whilst being able to sample the data at a rate higher than the surgeon's sensing bandwidth (the frequency with which the tactile and/or kinaesthetic stimuli are sensed) [48].

Extensive research regarding hand/wrist force exertion [51; 52] and tactile/kinaesthetic sensing [48; 49] is available in literature. It is found that the kinaesthetic sensing bandwidth, the rate at which forces can be sensed in the muscles and tendons, is double that of the force exertion bandwidth (the rate at which a human can respond to his or her kinaesthetic sense). Brain response calculation and muscle activation time are the primary factors for the difference in sensing and exertion bandwidths. It is also found that the minimum force a human can effectively perceive using the hand and wrist is 0.025N while, on the other end of the scale, the maximum finger force experienced is in the order of ±100N. These findings are summarized in Table 4.1. Human hand perception and capabilities are discussed more thoroughly in Section 2.3.1. Note that the maximum finger force values indicated in this table represent the highest possible force achievable from Table 2.1 - which combines the mean and standard deviation results (for males) to obtain the true maximum.

Table 4.1: Human Hand Perception - adapted from [48; 49; 51; 52]

Human hand Capabilities		**Measurement**
Perceptual bandwidth	Cutaneous sensing	320 Hz
	Kinaesthetic sensing	20 to 30 Hz
	Force exertion	10 to 15 Hz
	Force resolution	0.025 to 0.05 N
Finger load force (maximum)	Poke	70.59 N
	Pull	98.00 N
	Press	69.27 N

4.2.2 Choosing the Appropriate Haptic Sensor

Consider the human hand. It is capable of complex manoeuvres, all of which fall into 2 main categories. It can exert a load force (pulling, pushing, pinching or grasping) and a torsional force (twisting and turning). Note, though, that it is also capable of any combination or multiples of the two. So it is essential that the chosen sensor (or sensor combination) be capable of measuring these forces in all three Cartesian coordinates.

Fitting an existing robotic manipulator with sensors for haptic feedback to measure force and moments in all directions, requires one of two approaches. The first approach involves placing multiple sensors at the manipulator's joints (known as joint force and torque sensors). This approach requires that these readings then need to be manipulated based on the current manipulator orientation to calculate what the actual force and moment values acting on the end-effector is. This is a tedious process and requires extensive modelling and knowledge of the manipulator dynamics to achieve accurate results. The second approach addresses this issue by applying sensors directly at the end-effector (known as wrist force and torque sensors) [81].

Narrowing the type of sensor(s) down to a "wrist" type force and torque sensor, capable of measuring forces and torques in 3 dimensions, effectively results in searching for a multi-axis force and torque sensor which meets the human hand perceptual requirements listed in Table 4.1.

4.2.3 Description of the Net F/T Sensor

ATI Industrial Automation is a leading engineering-based world developer of robotic end effectors, including multi-axis force/torque sensing systems amongst others [82]. Comparing this project's haptic requirements in Table 4.1 to the specifications of all the available models supplied by ATI, a suitable sensor has been chosen. The Net F/T Gamma (calibration: SI-130-10) sensing system, shown in Figure 4.2, consists of a Net box, a F/T sensor and a connecting cable. The sensor transmits 3 components of force (Fx, Fy, Fz) and 3 components of torque (Tx, Ty, Tz) data at rate of up to 7000Hz. The sensor axis along which these components are measured is illustrated in Figure 4.3. The Net

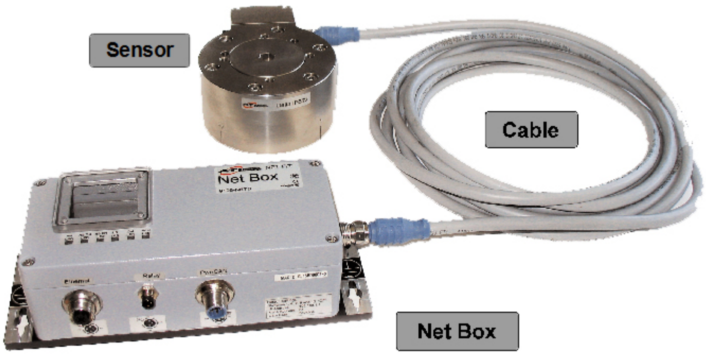


Figure 4.2: Net F/T Sensor [83]

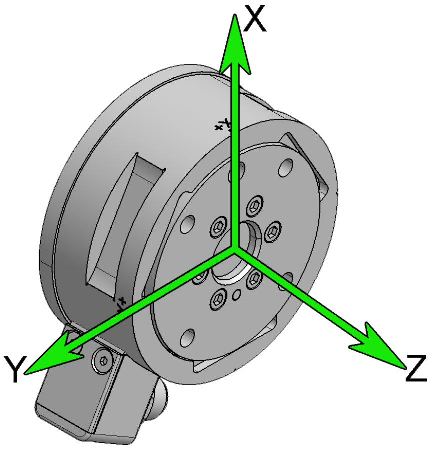


Figure 4.3: Sensor Reference Frame

box is equipped with multiple communication interfaces - including CAN bus, Ethernet and Ethernet/IP and can therefore be easily integrated into a local area network (LAN) for remote operation and monitoring. The Net box contains a web server which hosts a website for easy access to the Net F/T sensors configuration. The sensing system can be powered by Power over Ethernet (PoE) or via an external power supply.

The Net F/T Gamma sensor allows for the point of origin to be specified. This means the sensor can measure forces and torques acting on an arbitrary point relative to the sensor (as long as this point is connected to the sensing unit via a rigid connection). This makes it possible to specify the exact position of contact between the operator and Motoman from which to measure the applied forces and torques to the control system.

Comparing the Net F/T sensors specifications in Table 4.2 to the sensor requirements in Table 4.1, it is clear that the maximum force required is well within the limits capable by the sensor. Secondly, the force sensing resolution of the sensor is similar to the minimum force a human can perceive and is therefore acceptable. Furthermore, the sensing rate of the sensor is higher than what the human can effectively perceive (7000Hz as opposed to 500Hz for tactile sensing). These three requirements therefore indicate that the sensing system is capable of measuring the full range of force and torque the human hand can exert with sufficient resolution and at a rate that will not limit the operator's perception.

Table 4.2: Net F/T Sensing System Specifications [83]

Sensing	**Sensing Range**				**Resolution**			
Rate	Fx, Fy	Fz	Tx, Ty	Tz	Fx, Fy	Fz	Tx, Ty	Tz
<7000Hz	130N	400N	10Nm	10Nm	1/40N	1/20N	1/800Nm	1/800Nm

4.3 Communication System Development

In a telerobotic system comprising of more than one environment it is essential that a communication system is in place that allows interaction between the various sensors and devices. The communication system provides the methods for sending instructions and receiving feedback to both operator and remote environments. As such this section details the process of developing a functioning communication system, starting with an overview of what will be required to set up the communication interface between the available technologies. An in depth approach is then taken to discuss the communication set-up between these corresponding devices.

4.3.1 Overview of the Communication System

To communicate between two separate environments, a communication network must be in place to effectively "carry" the data between the two points. For the purposes of this research project, the university's intra-network is used. With a communication channel available to link the two environments the next step is making sure all the devices required for a telerobotic system are "network enabled". A quick look at the available devices (multi-axis F/T sensors, network camera, Motoman XRC and DX100 robot controllers) indicates that the Motoman XRC robot controller only has a serial interface and does not support Ethernet network interfacing. A Barix Barionet 100 device will therefore serve as a network enabling device for the MotomanUP6 robot with XRC controller. The diagram in Figure 4.4 illustrates the resulting communication channels for connecting all the devices to the communication system.

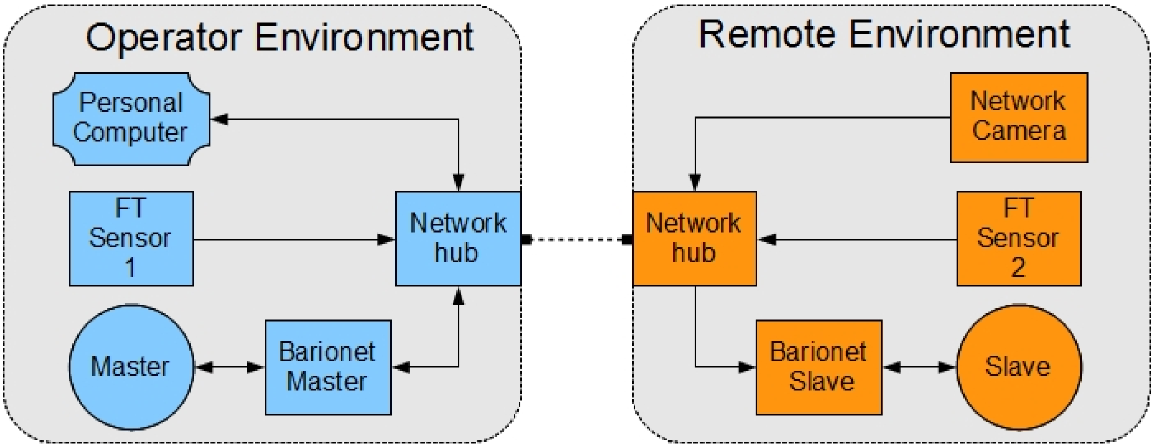


Figure 4.4: Device Communication Overview

4.3.2 Communication Architecture

The initial approach for the telerobotic communication system was to develop two separate communication systems, the first being the video communication system, as shown in the diagram in Figure 4.5, that basically displays the video stream of the network camera in the remote environment on a personal computer in the operator environment. In this communication system, the personal computer is responsible for intercepting the video feed of the network camera and displaying it on the screen. The second system, the haptic communication system, comprised of two Net F/T sensors, two Barionet communication devices and the Master and Slave manipulators. The second system relies on the Barionet devices to intercept the Net F/T sensors' readings, calculate the appropriate action to be taken and then follow the handshaking protocol of the Motoman robots, the Master and Slave respectively, also illustrated in Figure 4.5.

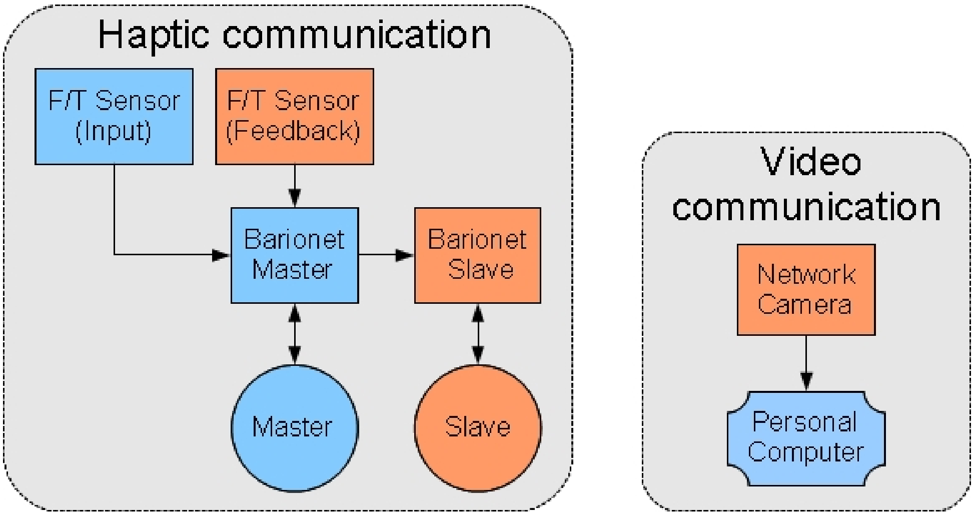


Figure 4.5: Initial Communication Architecture

After investigation into the Barionet programming limitations it was found that this was not a feasible solution. The Barionet programming language, known as BCL, only supports whole numbers (integers) for mathematical calculations [76]. This poses a problem as all calculations performed by the BCL program will ignore decimal values. This is especially detrimental to the accuracy of the control system as several calculations need to be made to transform the input data into robot commands. The same computation limitation is present for receiving sensory data.

The F/T sensors output their data in a custom structure, or packet, consisting of several integer values which must then be modified, according to the calibrated standard, before force readings can effectively be read. The process of calculating the measured force values of the sensors using BCL will, thus, result in inaccurate readings as fractional data is discarded after each calculation. These inaccurate readings are then used as an input for the control system for further calculations. This lowers the accuracy of the system even further during the process of determining the appropriate instruction to be sent to the Master and Slave devices respectively. Evidently, using the Barionet for any data manipulations or calculations will result in inaccurate sensory input into an inaccurate process. An alternative approach is needed.

4.3.3 Introducing the Interpreter

With the programming language of the Barionet (BCL) not capable of achieving the desired computational accuracy a different programming language needs to be considered. As previously mentioned a personal computer will already be used to receive video feedback from the network camera. A possible solution is, thus, to use the computational capabilities of a computer to intercept and manipulate the data before sending it to the respective Barionet, as shown in the diagram in Figure 4.6.

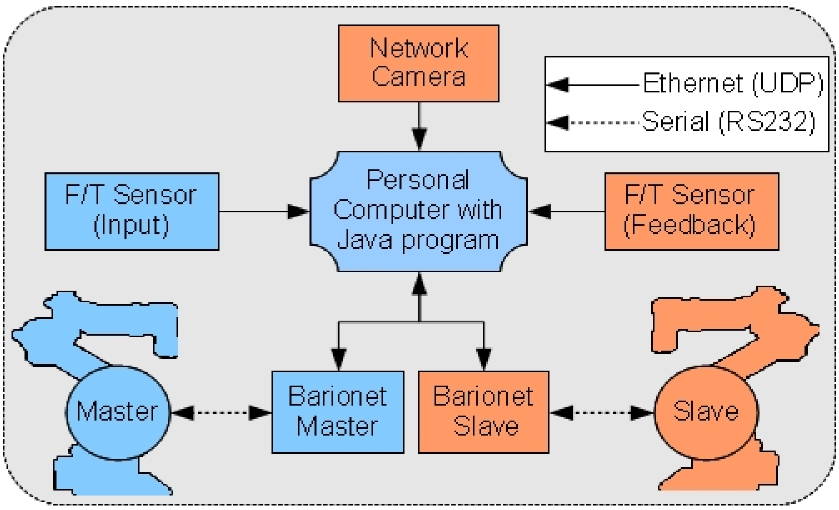


Figure 4.6: Java Based Communication Architecture

The Net F/T sensors are supplied with a Java interface capable of receiving and separating the various components of the data packet sent by a single sensor over a network.

This Java interface demonstrates that the Java runtime environment is suitable for both high-speed networking (communication) and computational (control system) applications. Furthermore, as this "demo" interface provides all the necessary communication algorithms to communicate with a F/T sensor, there is no need to develop these communication algorithms from scratch.

The communication system will therefore make use of the already defined algorithms within the Java environment and extend its capabilities according to the communication requirements in Figure 4.6.[2]

With the physical communication structure in place, the next step is developing the necessary communication protocols for sending and receiving data between the appropriate devices. The rest of Section 4.3 describes key components on how the communication between these devices is set up without delving into programming specific content.[3]

4.3.4 Barionet Communication Set-up

The Barionet device acts as a "middle man" between the computer and a Motoman controller, as such, it needs to be able to transmit data to the Motoman controller as well as receive data from both the computer and the controller. Communication with the Motoman controller requires a serial interface and as such the Barionet device needs to be situated close to the the Motoman controller. Communication with the Java controlled computer therefore makes use of the Ethernet interface to communicate over the network. The Barionet programming language, BCL, has built-in functions to effortlessly set up multiple protocols in a single program. To ensure that communication can occur, it is essential that these protocols are set up according to the Motoman communication requirements as listed in Table 4.3.

Table 4.3: Motoman Communication Requirements

Motoman Configuration	**Value**
Interface	RS-232C
Transmission speed	9600 bps
Parity	Even
Data bits	8
Stop bits	1
Flow Control	None
Port number	1

Communication between the Barionet and a specific computer also require a specific communication set-up. Configuring this UDP communication requires the address and port from which the UDP data will be transmitted. This address refers to the IP address of the computer (a means of identifying a specific computer on a network) on which the Java

[2]The compiler used for the purposes of this project is Netbeans IDE 6.8 running Java development kit 1.6 (JDK 1.6).

[3]For programming specific content refer to the attached CD at the back of the report, containing a complete copy of both the Java and BCL source code.

application will be run, whilst the port (communication end-point) can be any unused port on the network. A BCL code segment is provided in Figure 4.7 which demonstrates how the two communication protocols are set up and also shows a basic example of receiving data over Ethernet (UDP protocol) and then transmitting that data across the serial port (RS-232 protocol).

```
------------------------------------------------------------------------
BCL Communication Example
------------------------------------------------------------------------
DIM COM$                           // RS232 variable declaration
DIM UDP$                           // UDP variable declaration
DIM Msg$                           // Data variable declaration

COM$ = "COM:9600,E,8,1,NON:1"      // Configure serial port
OPEN COM$ AS 1                     // Open serial port (RS232)
UDP$ = "UDP:146.232.146.136:12301" // Configure UDP port
OPEN UDP$ AS 2                     // Open UDP port

DELAY x                            // Wait for UDP message to be received
READ 2, Msg$                       // Read UDP data into string Msg

WRITE 1, Msg$                      // Transmit Msg to Motoman (RS232)
DELAY x                            // Wait for Msg to be transmitted
------------------------------------------------------------------------
```

Figure 4.7: BCL Communication Example

Utilizing BCL's read and write functions, data can be transmitted and received over either serial (RS232 protocol) or Ethernet (UDP protocol). To ensure that the full UDP packet is read, sufficient time needs to be given after the host computer transmits the UDP message. Similarly, after writing data, the BCL program must wait for the complete packet to be transmitted. The minimum time required to wait after sending or receiving data is dependant on the amount of data that has been (or needs to be) transmitted and can be determined using Equation 4.3.1.

$$t_{wait} = \frac{D_{total}}{R} \tag{4.3.1}$$

With D_{total}, the total amount of data to be sent (or received), while R, represents the data transmission speed (bit rate) used between the devices. The Motoman communication requirements, listed in Table 4.3, indicate that R is fixed at 9600, so in order to calculate the minimum wait time only the amount of data (units in bits) per transmission is required. Note that to communicate with the Motoman controller, the serial interface needs to be configured according to the controller's exact specifications and even then a particular procedure needs to be followed to successfully control the manipulator.

4.3.4.1 Motoman Handshaking

Communication with the Motoman controllers requires that specific characters be transferred to the controller to "inform" it that it will receive an instruction to execute. This sequence is known as a handshaking protocol and is required for each command received from a host computer. This handshaking protocol, along with the relevant control characters, is shown in Figure 4.8 and Table 4.4 respectively. The handshaking protocol works as follows:

Step 1: The host computer sends an enquiry (ENQ) to the Motoman controller to establish a connection.
Step 2: The controller responds with an acknowledgement (ACK0) that the enquiry has been received and the link is established.
Step 3: The host computer then sends the full command packet. Where "COMMAND" is substituted with the appropriate instruction.
Step 4: The controller responds with a second acknowledgement (ACK1), indicating that the full packet has been received.
Step 5: The transmission link is terminated (EOT).

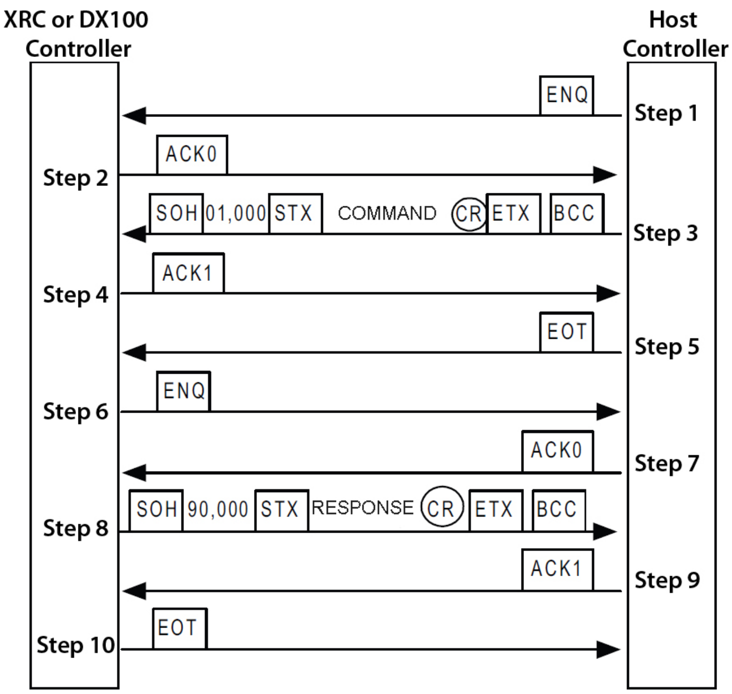


Figure 4.8: Motoman Handshaking Protocol

These 5 steps are then repeated (in step 6 to 10), with the host computer and Motoman controller's role reversed. In the second repetition, the controller sends a response packet to the host computer, indicating the state of the controller regarding the command that was received during the first cycle. As the Barionet is required to communicate with the Motoman and no mathematical calculations are required, this handshaking protocol could be developed in BCL, significantly reducing the network traffic as the only information needed from the Java application is, thus, the "COMMAND" itself.

4.3.4.2 Barionet Limitations and Considerations

During the development of the Barionet communication algorithms in BCL, several issues have been identified that require an alternative solution to function the way it was

Table 4.4: Transmission Control Characters

Control Character	**Code (hexadecimal)**	**Meaning of Control character**
CR	13	Carriage Return
SOH	1	Start of Heading
STX	2	Start of Text
ETX	3	End of Text
EOT	4	End of Transmission
ENQ	5	Enquiry
ACK0	10, 30	Even Affirmative Acknowledgement
ACK1	10, 31	Odd Affirmative Acknowledgement
BCC	Unique 3 digits	Error Check bits

initially intended.

The first limitation has to do with receiving UDP messages via the Barionet's built in "UDP event listener". This capability is ideal for being informed when a new UDP packet arrives, however, in doing so it interrupts the current BCL program and hence the handshaking protocol, that causes data transmission and receiving errors on the Motoman controller. An alternate method to receive the UDP data without interrupting the handshaking protocol is required. The proposed method makes use of a automatic UDP handle check after the handshaking protocol completes. This new approach therefore only listens for UDP events if the handshaking protocol is not currently running.

Another aspect that required refinement was the "wait" before reading and after writing data as demonstrated in the code snipped in Figure 4.7. While the minimum time to wait can be determined based on Equation 4.3.1 given the length of the message to receive or transmit, the Motoman controller does not respond immediately so the required wait time before reading is difficult to predict. Initially a very long wait time (in the order of 200ms) was used to inspect whether the rest of the handshaking protocol was functioning correctly, however, after a bit of testing it became apparent that even this long wait time could be problematic. This is because the time it takes for the Motoman to respond has to to with the command it receives, meaning that a command requiring more motion (and essentially more time to complete) takes longer to respond back to the Barionet device. Furthermore, with the command being calculated in real-time based on the operators actions, the command could vary significantly so a fixed wait time would not be sufficient as data transmission errors could still occur.

The proposed solution was to develop an event driven wait procedure within the handshaking protocol. The handshaking protocol continually monitors the serial interface for incoming data. Once the first data bit is received, the resulting wait time needed to receive the full packet could be calculated based on the baud rate. These solutions to the Barionet communication algorithms are illustrated in Figure 4.9. Note that only the first five steps of the handshaking protocol are illustrated.

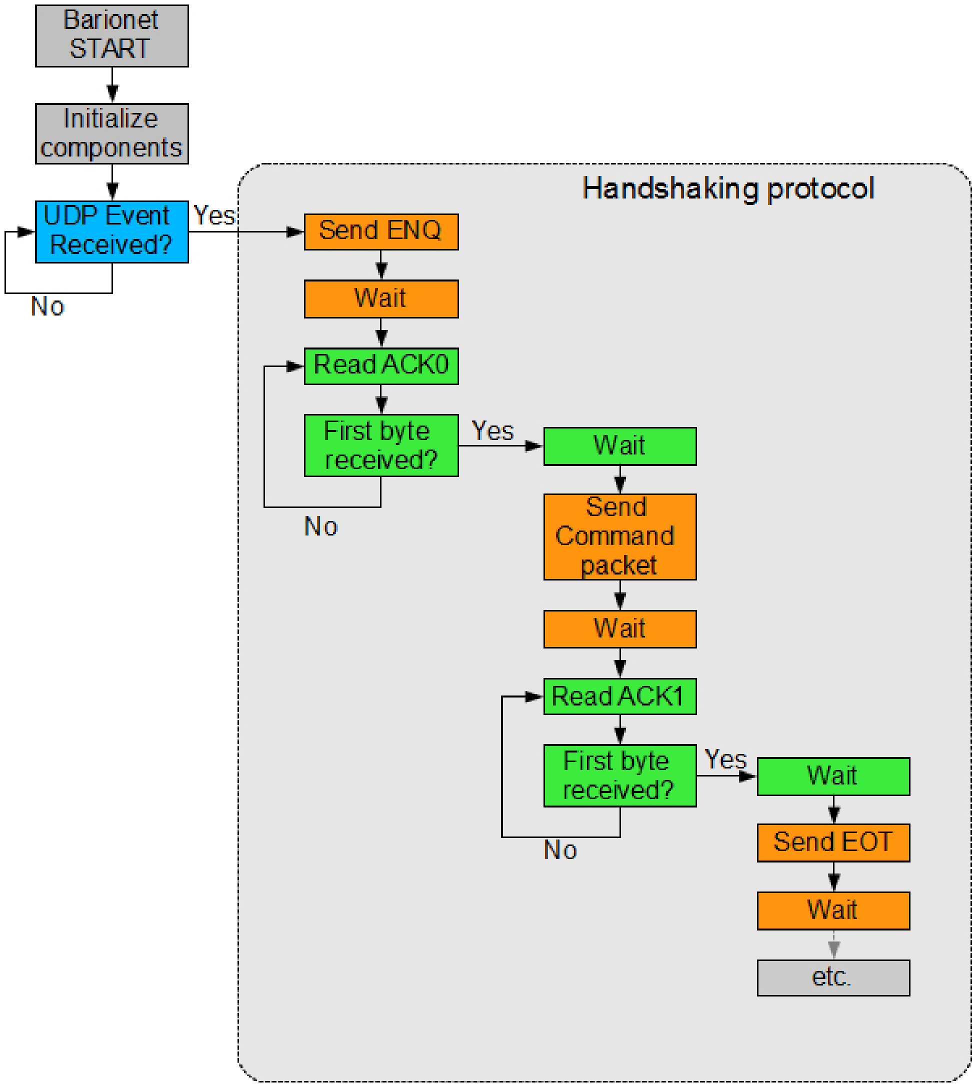


Figure 4.9: BCL Handshaking protocol

4.3.5 Java Communication Set-up

The Java application needs to be able to receive data from the F/T sensors and the network camera as well as transmit data to the Barionet. The F/T sensors communication algorithms are provided with the "demo" Java interface as described in Section 4.3.3, however, communication algorithms are still required for the network camera and Barionet.

While the Barionet communication algorithm has been developed for receiving UDP messages, the Java application still requires a method for sending these UDP messages to the Barionet device. Java contains two essential libraries aimed at network data communication. Communication with devices (Java IO package) and communication over a network (Java Net package) are used in combination to achieve device communication over a network. These Java libraries contain their own methods for sending data packets over a network using UDP protocol. It consists of creating a datagram packet and opening a datagram socket. The datagram packet contains the message (or command) to be sent as well as the address to which it should be sent. Opening a datagram socket establishes a communication link to the address specified in the datagram packet before the message is sent.

The Vivotek network camera is responsible for providing video feedback to the operator. Communication between the Java interface and the Vivotek network camera is, thus concerned with transmitting the video stream (captured by the camera) over the network and displayed to the operator. The network camera is equipped with various interfacing options for streaming the video over a network as noted in Section 2.4.3. An HTTP protocol is used for capturing a still frame or image from the video stream and transmitting it over the network to the Java interface. Repeating this communication sequence will replace the still frame with a newly acquired image. The speed and computation capabilities allow the image to be updated at a rate faster than what the human eye can detect, thereby creating the illusion of a video stream.

4.4 Control System Design

A control system is an interconnection of components to provide a desired function [84]. Figure 4.10 illustrates a block diagram of a generic feedback control system, consisting of a process, controller, command input, controlled output and measurement. Table 4.5 defines how these terminologies apply to the Master-Slave telerobotic system being developed in this project.

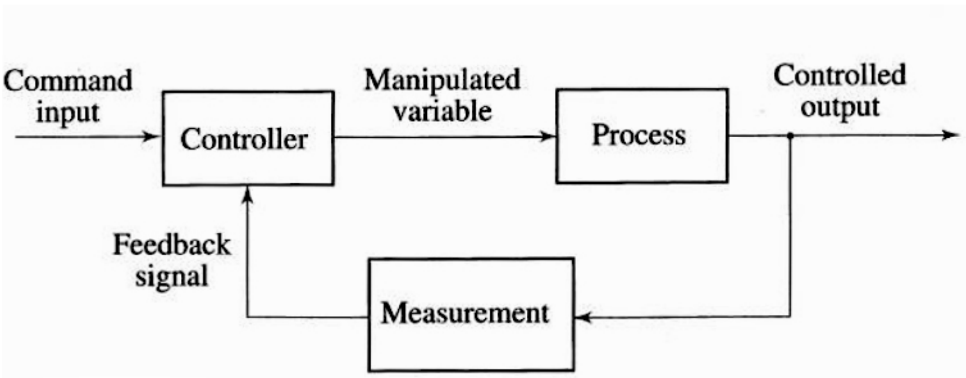


Figure 4.10: Generic Feedback Control System [85]

Before a controller can be designed it is necessary to outline the scope of the controller to identifying exactly what input it will receive, what factors will influence the input, the type of output it needs to deliver and the effect a specific output will have on the control algorithm.

4.4.1 Command Input

Technical specifications regarding ATI Industrial Automation's Net F/T sensing system has been given in Section 4.2.3, however, the effect of sensor orientation has not been discussed. While the sensing reference point can be changed, the coordinate system of the sensor remains fixed as illustrated in Figure 4.3. This has a profound effect on the sensor readings for a fixed force for any change in the sensor's orientation.

Consider the scenario illustrated in Figure 4.11. Two sensors, with different orientations, are measuring the same external force vector. While the magnitude of the force vector is measured accurately by both sensors, the force direction differs considerably as the

Table 4.5: Control System Terminology

Criteria	**Description**
Process	The portion of the telerobotic system to be controlled i.e. the Master and the Slave
Command input	Variables within the operators control that must influence the process i.e. the force and torque applied to the Master
Controller	The portion of the telerobotic system that determines how the process is controlled. This is the aspect that needs to be designed to allow an operator to control the Master in a intuitive manner with his or her hand while receiving force and torque feedback from the Slave
Manipulated variable	Variables outside the operators control that directly influence the process i.e. robot instructions (jobs)
Command output	The result of controlling the process i.e. The motion of the Master and Slave
Measurement	A measure of how well the process is being controlled i.e. the force and torque experienced by the Slave. The measure is capable of influencing the controller with the goal of improving the process i.e provide feedback to the operator

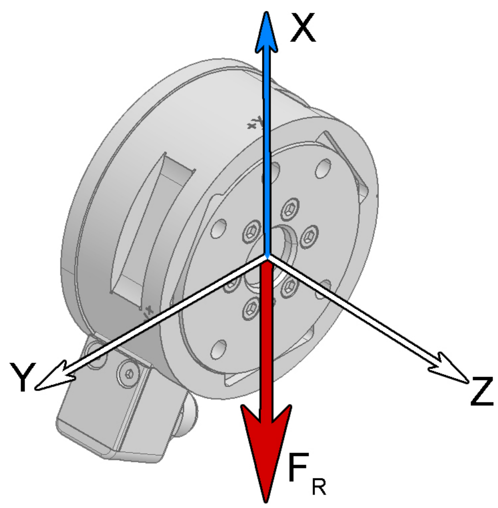
sensor's orientation effects the individual components (Fx, Fy and Fz) of the measured force. Orientation A will therefore measure the bulk of the external force in its x-axis while orientation B will measure the bulk in its z-axis.

This scenario illustrates the fact that the orientations of the sensors must be identically aligned (relative to a external reference point) for the force and torque feedback from the Slave to hold any value to the Master. These two sensors are fitted to the end-effectors of the Motoman robots. These sensors can be aligned by sending the same job (essentially a start-up job) to each robot which synchronizes the position and orientation of both robot end-effectors. This synchronization needs to be done every time before the telerobotic system is to be used. For the controller design purpose it will assume that this start-up job is executed (that the sensors are synchronized) before the control system begins to control the process.

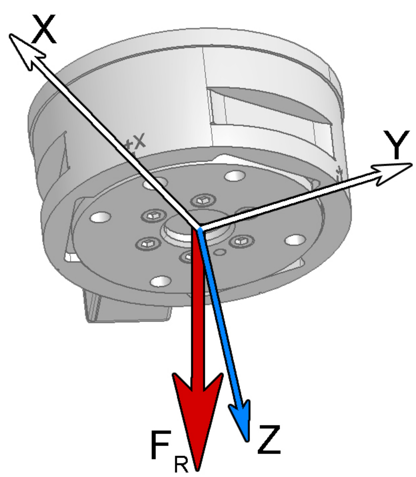
4.4.2 Manipulated Variable

As mentioned in Section 2.4.1, control of the Motoman robots over a network requires the use of the Motoman's Host control functions. Several methods are available for controlling the manipulator via these functions, each having a different effect on where and how the Motoman controllers move the robot's end-effector - these functions are:

- MOVJ



(a) Orientation A



(b) Orientation B

Figure 4.11: Sensor Orientation Effect

- MOVL
- IMOV
- PMOVJ
- PMOVL

Note that all these functions are position and speed based move commands with the only difference being the coordinate system and the motion type used to reach the specified coordinate. Table 4.6 lists the commands according to these two criteria. The two unique

Table 4.6: Motion Command Comparison

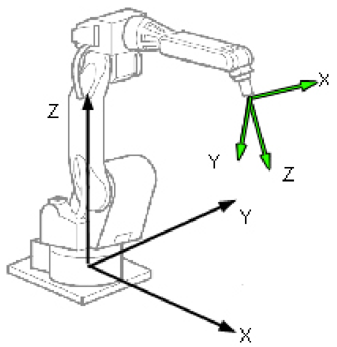
Command	Motion Type	Coordinate System
MOVJ	Joint	Fixed
MOVL	Linear	Fixed
IMOV	Linear	Fixed, Tool
PMOVJ	Joint	Pulse
PMOVL	Linear	Pulse

motion types are linear motion and joint motion. In linear motion, the end-effector is limited to a linear path between its current position and commanded position while joint motion has no restrictions and merely moves/rotates the joints to reach the appropriate coordinates and orientation. Theoretically, if a force vector is applied to the end-effector, the end-effector should move in a linear fashion in the same direction as the applied force, as such linear motion will be used in this project.

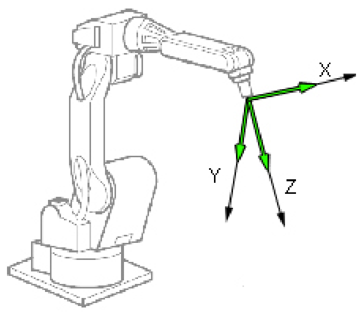
The Motoman controller allows for three unique coordinate systems to identify the position and orientation and two unique motion types to specify the trajectory to take. Figure 4.12 illustrates the difference between the three coordinate systems and how each relates to the coordinate system of the sensor when mounted to the end-effector.

The “fixed” coordinate system takes an arbitrary reference point (such as the base of the robot or any custom defined user point) and sets it as the origin of the coordinate system for all movement commands. The “tool” coordinate system makes use of a variable coordinate system where the reference point is at the tip of the end-effector and the coordinate axes are always perpendicular to the end-effector’s current orientation. In the pulse coordinate system, the reference point is replaced with servo motor pulse positions. So instead of specifying a certain coordinate and orientation for the end-effector, the operator specifies the number of pulses (angles of rotation) for each of the servo axis (S, L, U, R, B, T).

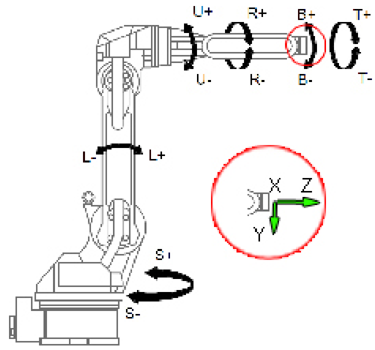
Regardless of the coordinate system used, these functions all require 6 values to fully quantify the position and orientation of the manipulator’s end-effector. For the “fixed” and “tool” coordinate systems, three values are needed for positional accuracy (x, y, z) and three angular values (θ_x, θ_y, θ_z) are needed for orientational accuracy. The pulse coordinate system relies on the careful selection of each servo pulse position to reach a specific coordinate and orientation.



(a) Fixed



(b) Tool



(c) Pulse

Figure 4.12: Coordinate Systems

Comparing these “black” coordinate systems to the “green” coordinate system used by the Net F/T sensor described in Section 4.4.1, it becomes apparent that the “tool” coordinate system has one key similarity which makes it ideal for simplifying the control algorithms. With the sensor mounted to the end-effector, the tool coordinate reference frame and sensor’s reference frame will remain fixed relative to each other - regardless of the current orientation of the manipulator, as illustrated in Figure 4.12b. The resulting effect is that the individual components of the measured force (Fx, Fy, Fz) can be directly linked to a specific axis on the tool coordinate system. This forgoes the need for extensive calculations that would be required to determine the position and orientation of the end-effector via inverse kinematics [86], as would be the case if using the “pulse” coordinate system (Figure 4.12c). This also relinquishes the need to incorporate the differences in orientation between the Motoman reference frame and the sensor’s reference frame, as would be the case if using the “fixed” coordinate system (Figure 4.12a). As such the “IMOV” command will be used for controlling the manipulator. An example exemplifying the “IMOV” command syntax is provided below:

IMOV d1, **d2**, **d3**, **d4**, **d5**, **d6**, **d7**, **d8**, **d9**, d10, d11, d12, d13, d14, d15, d16, d17

With variable **d2** specifying the end-effector motion speed. Variable **d3** specifies the coordinate system (tool or fixed). Variables **d4**, **d5** and **d6** represent the incremental distance value in the x, y and z coordinates respectively with measurements in millimetres. Variables **d7**, **d8**, **d9** represent the incremental angular values (measured in degrees) around the x, y and z axis respectively. These exemplified variables will be determined by the control system while the other variables remain fixed. Only the key control variables are highlighted in this section. For a full explanation on each variable and its function refer to Appendix H.

4.4.3 Designing the Controller

As mentioned in Table 4.5, the purpose of the controller is to perform two functions:

- Allow an operator to control the Slave (via the Master) in a intuitive manner with is hand by changing the command input variables.
- Provide force and torque feedback from the Slave to the operator (via the Master).

Controlling the Master and Slave in an intuitive manner with a human hand means that the Command input (applied force and torque) need to be converted into the manipulated variable (an IMOV instruction) which moves the end-effector in the direction the operator is moving his/her hand. This motion must be similar to as if the operator was physically moving the object (pen, scalpel etc.) The second aspect, providing force and torque feedback, means that the force and torque acting on the Slave need to influence the operator's ability to control the Master, either by applying a force back to the Master or by resisting the Operators input in some way. This section therefore focuses on designing a position based force feedback control system.

4.4.3.1 Intuitive Control Formulation

Intuitive control is responsible for relating a force and torque input to a IMOV instruction (consisting of velocity and a change in position and orientation) that would move the end-effector as if the operator was physically moving the object.

Consider the illustration of a hand gripping a pencil in Figure 4.13a. The resulting force vector, $\vec{F}$, and torque, $\vec{\tau}$, applied to the object (a pencil in this illustration) is shown on the FBD (free body diagram) in Figure 4.13b. Note that specific assumptions were made with regards to the FBD to simplify the development of the equations of motion somewhat.
These assumptions are:

- The operator wants to feel as if he is manipulating an object (pen or scalpel) and not a powerful robotic arm. The physical and dynamic properties of the robotic arm will therefore not be considered.
- Gravitational effects on the object are neglected. This is to prevent the controller from moving the Master if the operator is not applying a force or torque to the Master due to the objects weight.

- Only a resulting external force (i.e. $\sum F \neq 0$) can cause the Master and Slave to move.
- The object geometry is simplified to that of a solid cylinder of length, L, and diameter, d.
- The operator makes contact at the tip of the cylinder. All forces and torque are applied to a single point at the tip of the object as shown in the diagram.
- The intervals between these successive calculations must at least match the human kinaesthetic bandwidth (30Hz) i.e. be not longer than 0.03s. Thus, for object dynamics constant acceleration for the interval is assumed.

With these assumptions in mind, the dynamic equation used to derive the relationship between force, distance and linear velocity is provided in Equation 4.4.1 [87].

$$\vec{F} = m\vec{a} \tag{4.4.1}$$

Where the external force vector ($\vec{F}$) is directly proportional to the mass of the object (m) and its linear acceleration vector ($\vec{a}$). Similarly the dynamic equation relating to the relationship between torque, orientation and angular velocity is given in Equation 4.4.2 [87].

$$\vec{\tau} = I\vec{\dot{\omega}} \tag{4.4.2}$$

The externally applied angular moment vector (more commonly known as torque), ($\vec{\tau}$), is directly proportional to the object's moment of inertia (I) and angular acceleration ($\vec{\dot{\omega}}$). From these two fundamental equations of motion the desired control algorithm equations can be derived. The translational motion due to an externally applied force is expressed in terms of distance (Equation 4.4.3) and velocity (Equation 4.4.4).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{F_x}{2m} \\ \frac{F_y}{2m} \\ \frac{F_z}{2m} \end{bmatrix} t^2 + \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} t + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{4.4.3}$$

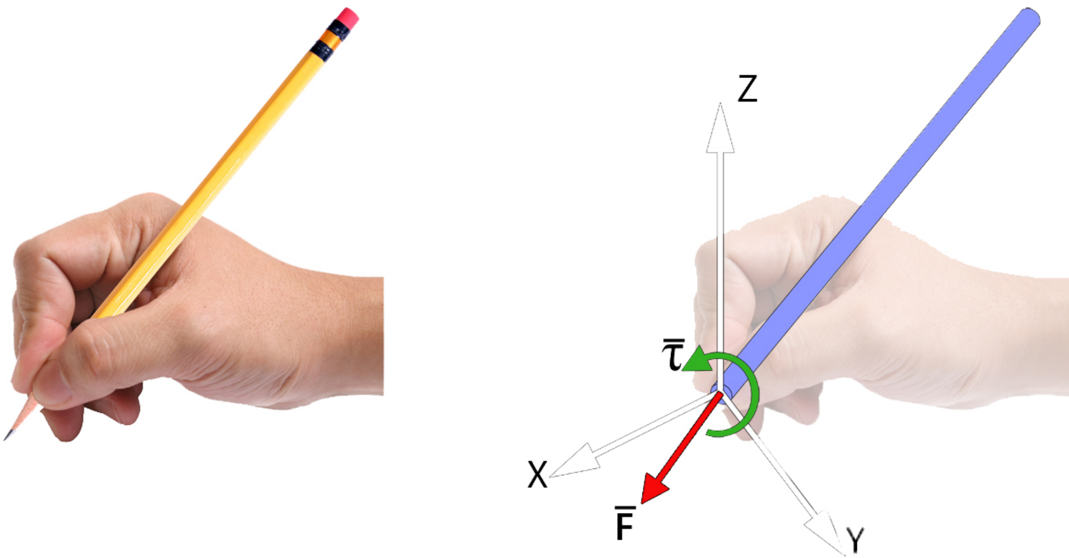
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{F_x}{m} \\ \frac{F_y}{m} \\ \frac{F_z}{m} \end{bmatrix} t + \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{4.4.4}$$

Where, m, is the mass of the object, t, the duration of the applied force and $\dot{x}_0$, $\dot{y}_0$, $\dot{z}_0$, the initial object velocity in the x, y and z axis respectively. Similarly the angular motion due to an externally applied torque is expressed in terms of orientation (Equation 4.4.5) and angular velocity (Equation 4.4.6).

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{2I_{xx}} \\ \frac{\tau_y}{2I_{yy}} \\ \frac{\tau_z}{2I_{zz}} \end{bmatrix} t^2 + \begin{bmatrix} \dot{\theta}_{x0} \\ \dot{\theta}_{y0} \\ \dot{\theta}_{z0} \end{bmatrix} t + \begin{bmatrix} \theta_{x0} \\ \theta_{y0} \\ \theta_{z0} \end{bmatrix} \tag{4.4.5}$$

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{I_{xx}} \\ \frac{\tau_y}{I_{yy}} \\ \frac{\tau_z}{I_{zz}} \end{bmatrix} t + \begin{bmatrix} \dot{\theta}_{x0} \\ \dot{\theta}_{y0} \\ \dot{\theta}_{z0} \end{bmatrix} \tag{4.4.6}$$

Where, I_{xx}, I_{yy} and I_{zz}, is the mass moment of inertia about the x, y and z-axis respectively , t, the duration of the applied torque and $\dot{\theta_{x0}}$, $\dot{\theta_{y0}}$ and $\dot{\theta_{z0}}$, the initial angular velocity about the x, y and z-axis respectively. Refer to Appendix A for a full derivation of how these equations are formulated.



(a) Example Reference (b) Free Body Diagram

Figure 4.13: FBD Illustration

4.4.3.2 Adding Haptic Feedback

The previous section details the relationship between the sensor output and manipulator input. It does not, however, take into account any feedback from the remote environment. It is therefore essential that the command inputs (F_x, F_y, F_z, τ_x, τ_y, τ_z) into the equations of motion, in the previous section, incorporates both the operator's input on the Master as well as the feedback from the Slave in the remote environment before determining the distance, orientation and speed of the Master and Slave.

Incorporating haptic feedback requires the use of either active or passive feedback as described in Section 2.3.2. This thesis will make use of both as the inherently "safe" nature of passive feedback will aid in the initial testing of the system until all start-up issues are resolved before switching to the realistic (intuitive) performance of active feedback for the experimentation phase.

In passive feedback, only the force and torque applied to the Master can influence the trajectory and speed of the Master and Slave while any force and torque applied to the Slave can at most provide resistance to the Master and Slave's motion. In active feedback, the sensor reading are combined except when the force acting on the Slave is in the same direction (aligned) as the force acting on the Master. In this case the force with the largest magnitude controls the Master and Slave.

Table 4.7 provides a example of active and passive feedback which clarifies what the controller needs to achieve. Note the example only illustrates a simple case where two one-dimensional forces (Fx1 and Fx2) are considered.

Table 4.7: Difference Between Active and Passive Control

	Master [Fx1]	Slave [Fx1]	Aligned?	Passive	Active
a	1	5	Yes	Only Master	Only Slave
b	1	-5	No	Provide Resistance	Combine
c	5	1	Yes	Only Master	Only Master
d	5	-1	No	Provide Resistance	Combine

In example *a*, both Master and Slave experience a force in the same direction. Passive feedback can only provide a resistance and therefore only Fx1 is used to determine the manipulated variable (IMOV instruction) which controls both robots. In example *b* Fx1 and Fx2 are opposing forces with Fx2 larger in magnitude. Passive feedback will, thus provide a 5N resistance, meaning that the Master and Slave will not move until the operator (Fx1) exceeds 5N. Active feedback, however combines Fx1 and Fx2 to determine the motion of the Master and Slave. In this case the manipulated variable is created according to a -4N force. The Master and Slave will therefore move against the operator, effectively applying a force to the operator's hand. Example *d* illustrates the opposite of example *b*. In this case both passive and active feedback functions the same. Fx1 and Fx2 are combined and the manipulated variable is created according to a 4N force. The Master and Slave will thus move according to a 4N force even though the Operator is applying 5N, the operator thus feels a resistance of 1N.

These examples illustrate the following: if the Slave is experiencing a force (or torque) in the same direction as the Master, the manipulated variable is based on the greater of the two for active feedback while passive feedback only implements the magnitude of the Master. If the Master and Slave are experiencing opposing forces (or torques), the manipulated variable is based on the sum of the two opposing components for both active and passive feedback, except for the case where the magnitude of the Slave is greater than the Master, in which case passive feedback will keep both manipulators stationary by setting the resulting input to zero. These haptic feedback criteria's can be formulated as:

$$A_i = \begin{cases} Maximum(M_i, S_i) & \text{if components } M_i \text{ and } S_i \text{ are aligned} \\ M_i + S_i, & \text{if components } M_i \text{ and } S_i \text{ are opposing} \end{cases} \tag{4.4.7}$$

$$P_i = \begin{cases} M_i, & \text{if components } M_i \text{ and } S_i \text{ are aligned} \\ M_i + S_i, & \text{if components } M_i \text{ and } S_i \text{ are opposing, where } |M_i| > |S_i| \\ 0, & \text{if components } M_i \text{ and } S_i \text{ are opposing, where } |M_i| \leq |S_i| \end{cases} \tag{4.4.8}$$

Where A and P is the resulting force or torque input for active and passive feedback respectively, M, the Master reading, S, the Slave reading and i representing the x, y or z Cartesian component of the force (or torque) being considered.

4.4.4 The Developed Control System

After identifying and developing the individual aspects of the control system throughout Section 4.4.3, the final step is combining it into a functional control system. Figure 4.14 shows the developed control system for controlling the Master-Slave telerobotic system being developed in this project.

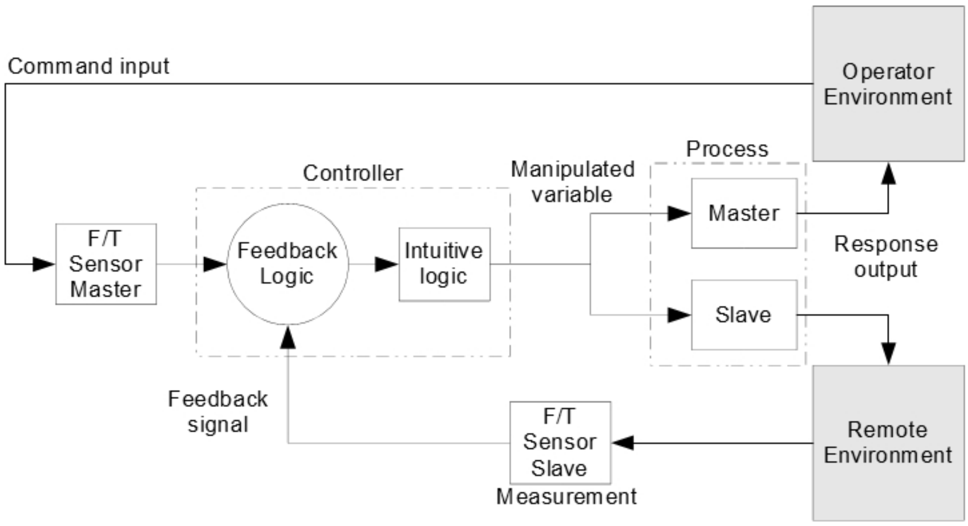


Figure 4.14: Position-based Haptic Control System for a Master-Slave Telerobotic System

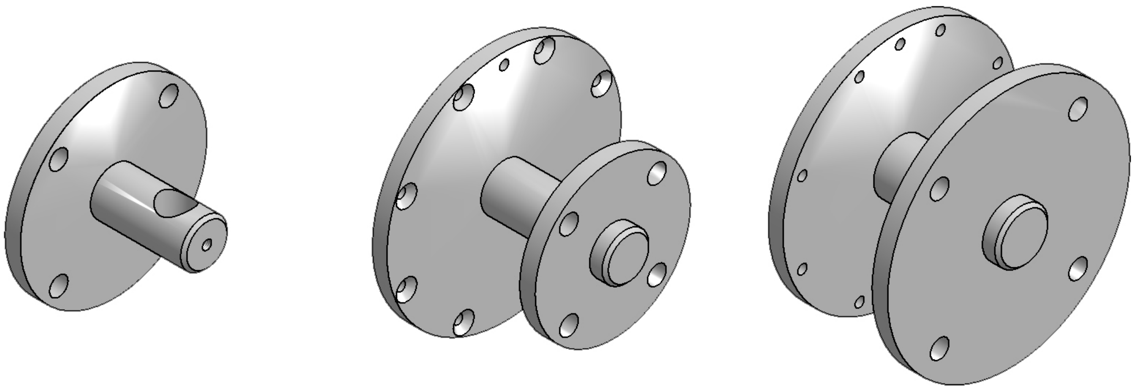
Evidently, this project make use of a variation of force control, known as position-based force control. The command input variables (force and torque) are measured and sent to Java application. Calculations are performed by the controller and an IMOV instruction is constructed and that moves both the Master and the Slave. Both robots receive the same IMOV instruction therefore the positional accuracy is expected to be good, based on the rated precision of the Motoman robots (see Appendix I). Thus, the positional accuracy between the Master and Slave should remains high regardless of the fact that the control system focuses force control.

4.5 Combining the Interfacing Subsystems

The telerobotic interface consists of two components. The physical haptic interface, through which the operator controls the Master and Slave and the software interface, which is responsible for the communication system, the control system as well as providing video feedback to the operator.

4.5.1 The Physical Interface

The physical interface comprises of two industrial Motoman robots and two ATI Industrial Automation F/T sensing systems. These sensors need to be mounted to the manipulators as well as provide a method for connecting the desired object, such as a scalpel or pen to the sensor equipped manipulator. The designed mounting plates are illustrated in Figure 4.15, whilst Figure 4.16 depicts a fully assembled physical interface on the Master. Refer to Appendix F for the full technical specifications. Note that the sensor can be mounted in 4 different orientations to the end-effector of the manipulator. It is therefore necessary to make the required adjustments to synchronize the sensor and manipulator's reference frames.



(a) Sensor-Object (b) Motoman(UP6)-Sensor (c) Motoman(SDA10)-Sensor

Figure 4.15: Mounting Plates

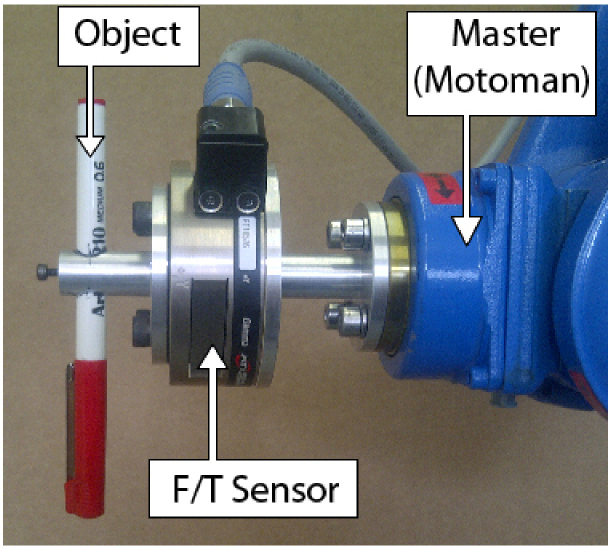


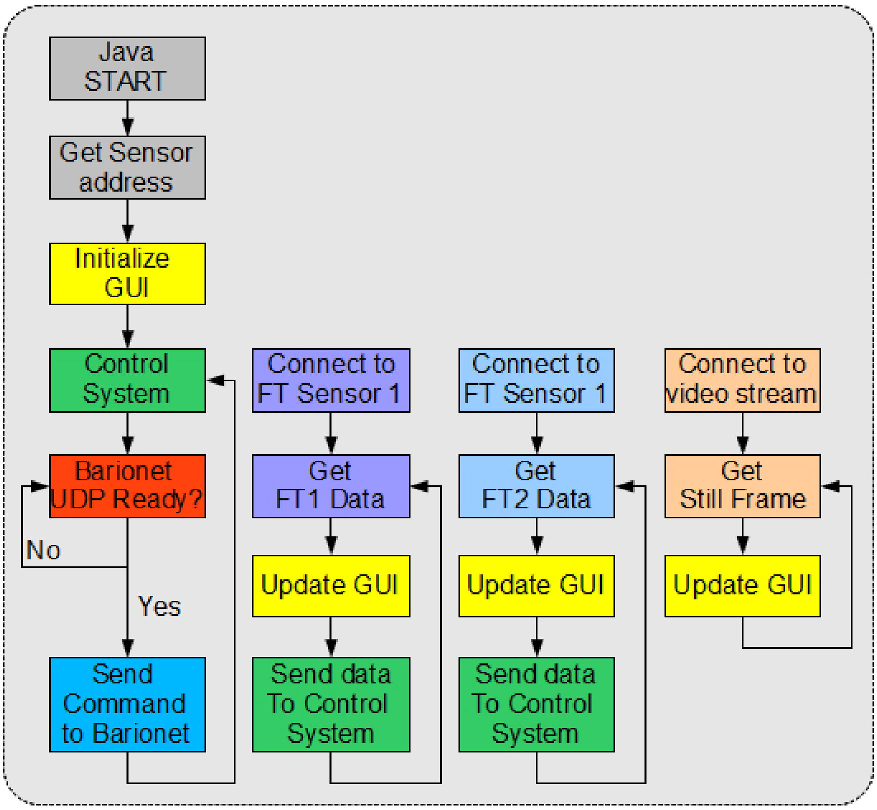
Figure 4.16: Fully Assembled Haptic Interface

4.5.2 The Software Interface

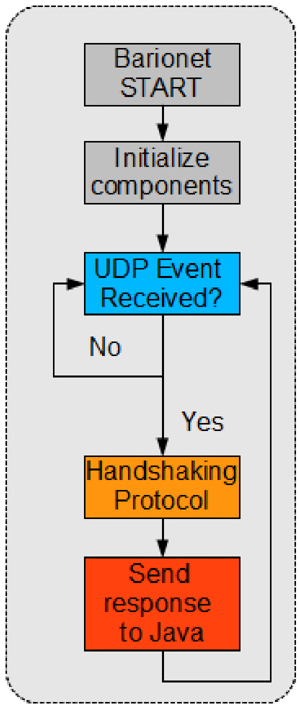
A telerobotic system, involving real-time control and feedback, requires that the developed subsystems be capable of performing their functions simultaneously. The Java interface must therefore be able to read the force and torque data from both the Master and Slave sensors, calculate the appropriate command, send these commands to the robot via the Barionet devices whilst continually displaying video feedback of the remote environment on a graphical user interface (GUI).

To accomplish this, the Java application is divided into several smaller processes, known as threads. A Java application would usually execute code sequentially, however, when using multiple threads (multi-treading) several bits of code can be executed concurrently. The diagram in Figure 4.17 illustrates the resulting process used to achieve communication and control of the telerobotic system in real-time.

Each branch indicates a separate thread. The main thread (left most branch) initializes



(a) Java Environment



(b) Barionet Environment

Figure 4.17: Telerobotic System Process Flow

the Java application and displays the GUI, after which it is responsible for the control system algorithms and sending the calculated robot commands to the relevant Barionet devices. The second and third tread continually intercepts data from the Master and Slave F/T sensor's respectively, each then updating the GUI and passing the formatted data to the control system. The final thread deals with the video feedback algorithms, by continuously intercepting the network camera's video stream, capturing a "snapshot" and updating the GUI. The figure also indicates the single process running on the Barionet device for communicating with the Motoman via the handshaking protocol discussed in Section 4.3.4.1.

4.5.2.1 The Graphical User Interface

The visual interface is developed in Java by incorporating two libraries, the Java.Swing class and the Java.awt class to the already developed communication and control system. This allows the Java application to be represented graphically. The primary focus of the graphical user interface is to provide video feedback from the remote environment, however, for the experimentation phase it is beneficial to graphically display the haptic readings and control system commands being sent, whilst also being able to turn off the system in case of a unexpected error. Figure 4.18 gives an example of the graphical interface.

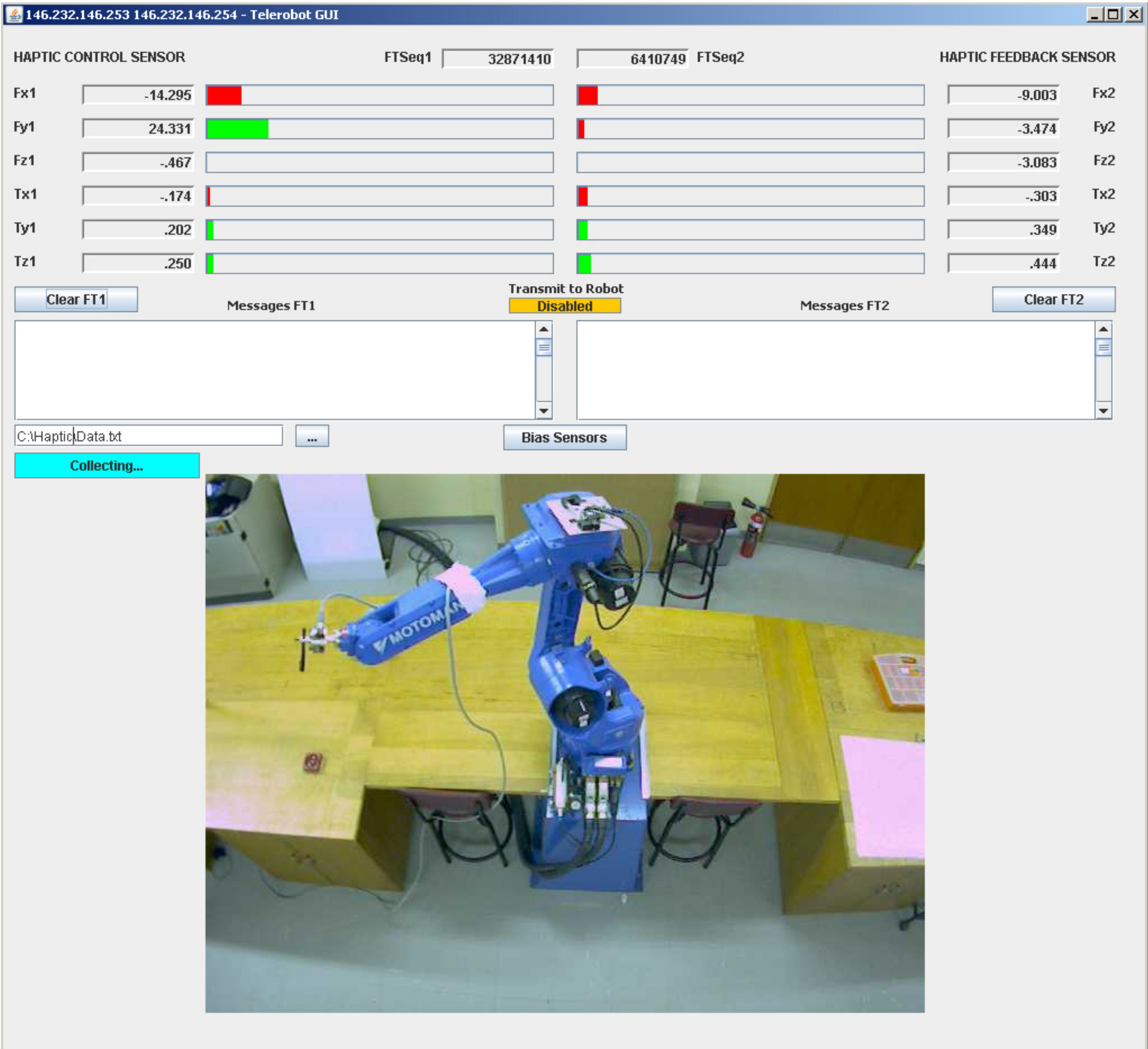


Figure 4.18: Graphical User Interface

4.6 System Limitations and Considerations

For the operator to be able to control the Slave in real-time via the Master, the control system needs to continuously relay updated control commands to both Master and Slave manipulators. The main issue experienced during initial testing is that the Motoman robots used are unable to receive commands in real-time from a host computer primarily because the controllers are unable to process a command and execute another at the same time using the Host Control Functions.

Before processing a new command the Motoman controller needs to complete the previous one. Command completion effectively means that the robot moves to a set coordinate at a specified speed and then stops. This means that the control of the Motoman robots are limited to single start-stop based commands as one command needs to finish execution before the next command is accepted to be processed. Now this is not a control system limitation, since the underlying issue is that the Motoman robots are not designed for real-time control. It is a hardware limitation, however, it has a severe impact on the control system.

Stopping after each command simplifies the equations of motion as the initial linear and angular velocity components will always be zero. While this simplification appears to be beneficial for calculating the next command, it only emphasizes a major limitation as the operator will never be able to control the Master smoothly due to the start stop procedure and secondly any sudden change to the input, be it from the operator or environment cannot be adapted to as the current command needs to be completed first. The latter poses a serious problem, especially if the current command is to move the end-effector at high speed over a long distance. During this time the Motoman robot will be unable to adapt whether or not the force that resulted in the Motoman's motion is still being applied (or even a different force altogether).

On the other end of the scale, applying a slight touch to the Master will translate into a command where the incremental positional changes are in the order of less than a millimetre. Combine this with the start-stop procedure and the result is an extremely jittery motion which prevents the operator from any real intuitive control over the Motoman robots. What is left is a system where the operator is constantly reminded, both visually and haptically, that he or she is controlling an industrial robotic manipulator which is not very intuitive for slight force changes but also not very safe to operate for large changes.

As this is a hardware limitation, these issues cannot be completely resolved, however, applying limitations to the control system can make the Motoman response more predictable and thus safer and easier to control. The rest of this section discusses how the control system is adapted to minimize the severity of these limitations.

4.6.1 Jittery Motion

Ideally the control system would only make use of velocity control, whereby the distance of each command remains fixed at the desired positional accuracy, say for example 0.1mm. Manipulating the end-effector would then consist of continuous 0.1mm translation commands at various velocities. While this ideal situation is possible, it degrades the performance of the system to a large extent. Using small incremental commands to manipulate the Motoman robots result in continuous acceleration and deceleration of the Motorman joints. Practically this comes across as jittery motion which makes the control of the end-effector difficult to judge. During initial testing it was discovered that for small input forces, the perceived jittery motion was less than for larger forces. This is due to the lower acceleration and deceleration requirements placed on the Motoman actuators to achieve the desired velocity.

A simple solution was thus, to limit the maximum velocity to a point where the perceived jittery motion did not hinder the operators ability to get a "feel" for the system. This posed another problem as the continuous start-stop procedure already limited the speed with which the end-effector could be moved, now the velocity has been lowered to minimize the jittery motion resulting in a further decrease in achievable speed. Moving the end-effector over a short distance, such as a few millimetres therefore took several seconds. A different approach was required. Another way to minimize the start-stop procedure is to use less commands to move the robot over a specific distance. This means using larger incremental values to control the end-effector.

4.6.2 Large Magnitude Inputs

Reducing the number of start-stop procedures meant that the Operator could cover a far greater ground over the same time, however, this resulted in further issues where the ability of the Motoman to adapt became an issue. The Motoman carries the command out till completion, which means that the motion cannot be halted or interrupted. Any change in the input force therefore goes by unnoticed by the Motoman as it cannot process the required change in trajectory and velocity. This means that the operator can apply a large magnitude impulse force to the Master, and it will complete the command even though the operator is not applying the force any more. This poses an accuracy concern as the Master will ignore the operators instruction until it is ready to process a new command. More importantly, though, this poses safety concerns as a large force by the operator could result in the Master or Slave colliding with an object even though the operator tries to counteract the motion.

The proposed solution it to limit the maximum (and minimum) distance that the Motoman can move during a single command with the aim of giving the operator the confidence in controlling the Master, knowing that if an unforeseen object gets in the way of the current end-effector trajectory or if the operator misjudges the amount of motion required, it will move at most this maximum distance before stopping. Unfortunately, this is a trade off between the control accuracy and response speed, both of which is required for telesurgical applications. The specific value to which the maximum distance should be limited to is, thus, a debatable issue and is based on the application as well as operator preference, skill and focus.

A trial-and-error approach is used to determine acceptable specifications for the maximum increment, minimum increment and maximum velocity. The resulting specifications are quantified in Table 4.8. Note that these specifications are based on the operator's personal preference. The maximum increment of 40mm provided the operator the surety that his method of control would suffice in providing the necessary response speed and safety. A combination of the minimum increment and maximum velocity ensured that the operator could manipulate the Master at the desired accuracy without being overly hindered by the continuous start-stop procedure.

Table 4.8: Combining Sensory Readings

Command Criteria	Specification
Maximum increment	40 mm
Minimum increment	0.5 mm
Maximum Velocity	50% of robot capability

Note that these specifications need to limit the motion of the robots, thus it must be incorporated into the control system. Figure demonstrates the updated control system where these control limitation are represented by the filter block, which modifies the IMOV instruction (manipulated variable) before it is sent to the Motoman robots.

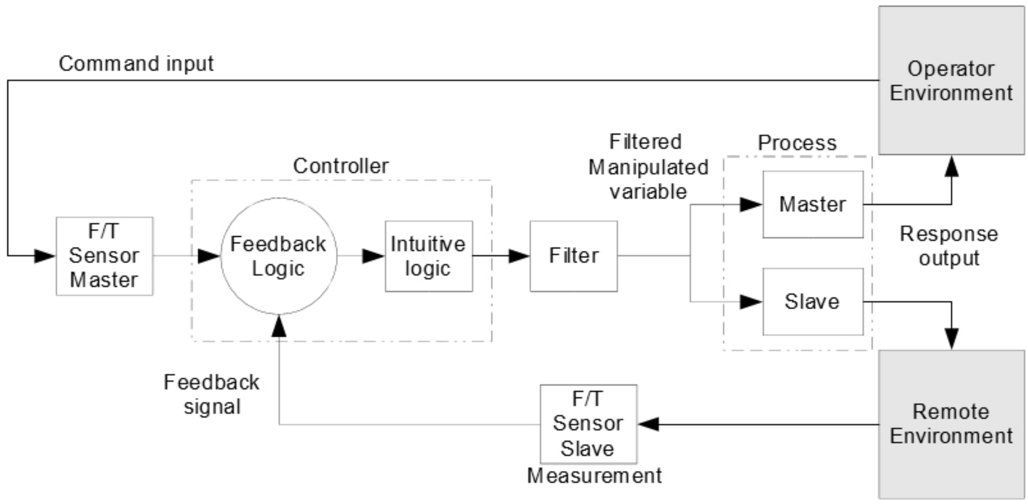


Figure 4.19: Updated Control System

Chapter 5

Evaluation Methods

This chapter details the methods and procedures used for evaluating the developed telerobotic system. Firstly, several evaluation methods and procedures are discussed for determining the pose capabilities, particularly the ability to move the haptic input manipulator (and subsequently the telerobot) to a specific point or along a particular path. The latter half of this chapter discusses the evaluation procedures for the haptic system.

5.1 Accuracy, Repeatability and Equitability

In order to evaluate the industrial robotic manipulators of the telerobotic system, a common set of metrics should be understood and must be established. The concepts of robotic repeatability and accuracy are most often used to describe the capabilities of motion systems [88], however, they are adapted to be applied to haptic systems as well. These concepts are defined below and are illustrated in Figure 5.1.

Pose Accuracy: The maximum position or orientation error obtained when moving to any point defined in Cartesian space. [89].

Pose Repeatability: The ability of the robot's end-effector to return to a particular point over and over again. [89].

Haptic Accuracy: The maximum force or torque difference between the remote and operator environment at the point of contact.

Haptic Repeatability: The ability of the haptic system to provide the same feedback for a given haptic input over and over again.

While these definitions describe accuracy and repeatability with regard to robotics, it is still required to determine how many samples (or measurements) must be taken to accurately quantify each term with the desired confidence level. For practical purposes, all evaluations adhere to a 95% confidence level with a 5% confidence interval. From these two criteria the minumum number of cycles required for each procedure can be determined, using Equation 5.1.1 [90], to ensure that the evaluated results are of equitable standard.

$$n = \left(\frac{T\sigma}{k\overline{x}}\right)^2 \tag{5.1.1}$$

Where n is the minimum number of measurements to be taken, σ and $\overline{x}$ is the standard deviation and mean value of the characteristic to be measured respectively, while k is the confidence interval and T is a value based on the desired confidence level - which is obtained from the t-distribution table in Appendix G.

The process of determining the minimum number of measurements is an iterative process, often requiring an accurate guess of the t-distribution variable. The proposed approach is to take 15 samples, i.e. an initial t-distribution variable value of 2.145 according to the t-distribution table. Using this initial guess the minimum sample size will be determined. Further samples are then taken until the minimum calculated samples are met. At this point the test is repeated with an updated t-distribution variable. This sequence is reiterated until the minimum sample criteria conforms to the current sample count.

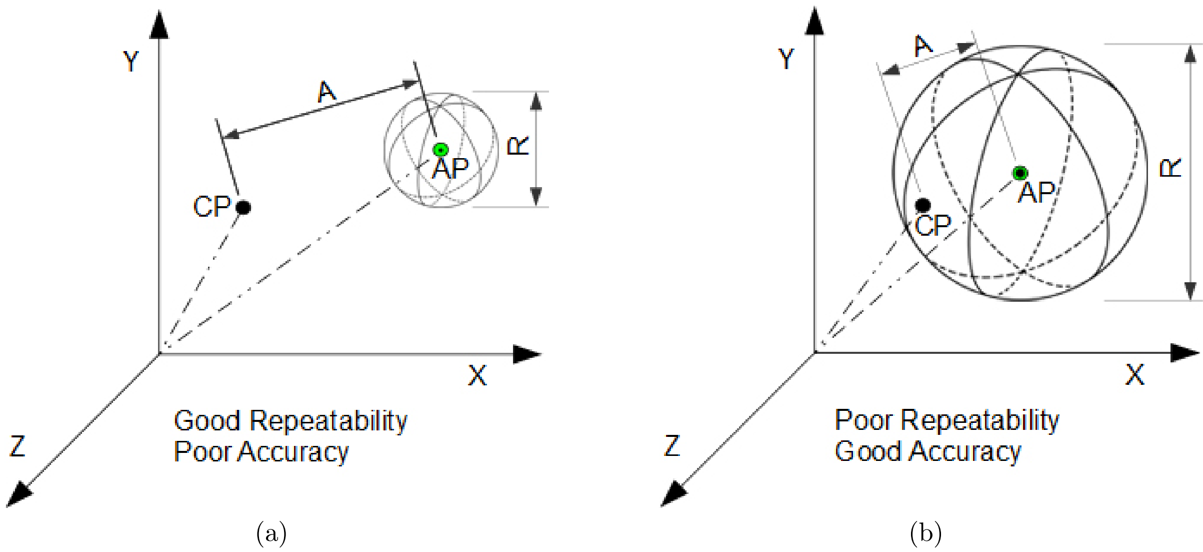


Figure 5.1: Accuracy and Repeatability

5.2 Evaluation Standards

According to literature the two standards commonly used for industrial robotic manipulator evaluation are:

- ISO 9283 International Standard
- ANSI/RIA R15.05-1 American National Standard

Utilizing the ISO 9283 test specifications allows for the measurement of several performance characteristics, including:

- Pose accuracy and pose repeatability
- Distance accuracy and distance repeatability
- Position overshoot
- Path accuracy and path repeatability
- Cornering deviations

The ISO 9283 standards are preferred as it offers an applicable approach, a well structured methodology and creates better simplicity - as all characteristics concerned can be evaluated on a single test plane [14]. The standard defines important performance characteristics, describes how they are specified and how they should be tested [91]. The test planes and test paths of this standard are defined with respect to a cube located inside the workspace of the robot. A diagonal plane inside the bounding cube is used to locate the "test plane", as illustrated in Figure 5.2a, containing the evaluation points and paths. This enables the evaluation procedure to incorporate all three Cartesian coordinates simultaneously as each point on the diagonal plane is uniquely defined in terms of x, y and z coordinates.

It is important to note that the ISO standard provides a means of evaluating a pre-programmed industrial robot with specific regard to motion. It therefore needs to be adapted before it can be used for evaluating a telerobotic system controlled in real-time by a human operator based on the operator's haptic and visual judgement.

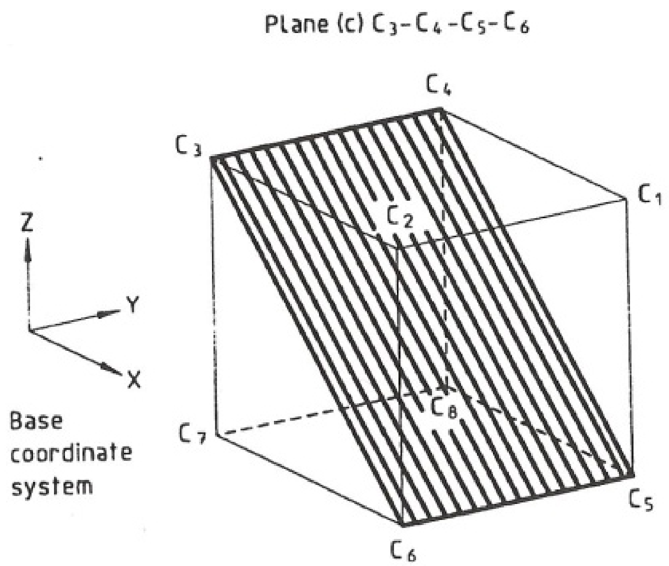
5.3 Adapting the Standards for Haptic Control

Unlike traditional robotic systems where the robotic manipulator is pre-programmed to move to the points defined in a "test plane" within the robot's workspace, the current system requires that an operator moves the end-effector manually by applying a force until the point is reached (known as the attained point). This requires a visual point for the operator to move to (referred to as the command point). A physical test structure therefore needs to be developed and positioned within the workspace of the robotic manipulator as shown in Figure 5.2a. Furthermore, while an industrial manipulator can be commanded to move between two points using either a linear, circular or spline trajectory, a human operator requires a "visual" reference path to follow, as such these command points are added to the test plane and are connected to form paths as illustrated in Figure 5.2b.

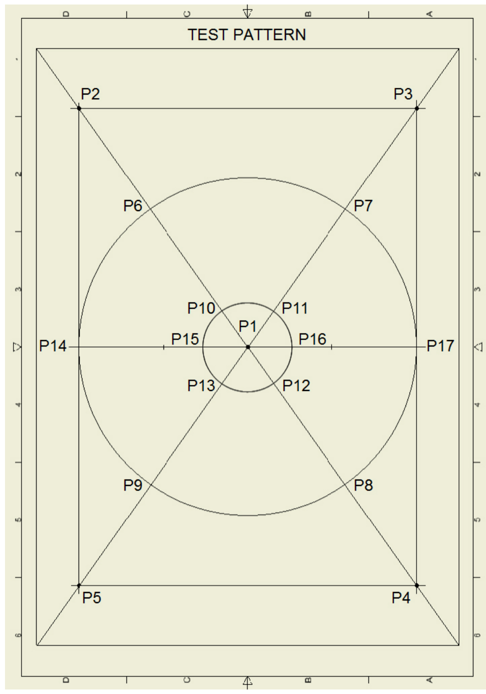
The ISO9283 is utilized for robot motion characteristic evaluation in an industry where these robots focus on repetitive tasks. It states procedures to determine robot accuracy and repeatability by performing the same task in exactly the same way and the calculations are based on these assumptions. This project, however, incorporates human aspects, meaning that even if a specific sequence is followed to move between command points, the speed and approach angle will vary each time as the operator moves the Master to the desired position based on his or her visual and haptic judgement. Furthermore, the accuracy of the attained points is essentially based on human acceptability - whether or not the operator feels that the desired accuracy has been reached. As such the order in which the experimental procedure takes place, specifically the pose evaluation in the next section, does not influence its result.

5.4 Pose Evaluation Procedures

If a force vector is applied to the Master and the Slave is not in contact with an object, i.e. no haptic feedback influences, the resulting effect is that both the Master and Slave manipulators will move in the direction of applied input force. Furthermore, once the input force is removed both manipulators will come to a halt. This allows the operator



(a) Diagonal Plane Test Structure [91]



(b) Command Points and Paths

Figure 5.2: Physical Test Plane with Command Points and Paths

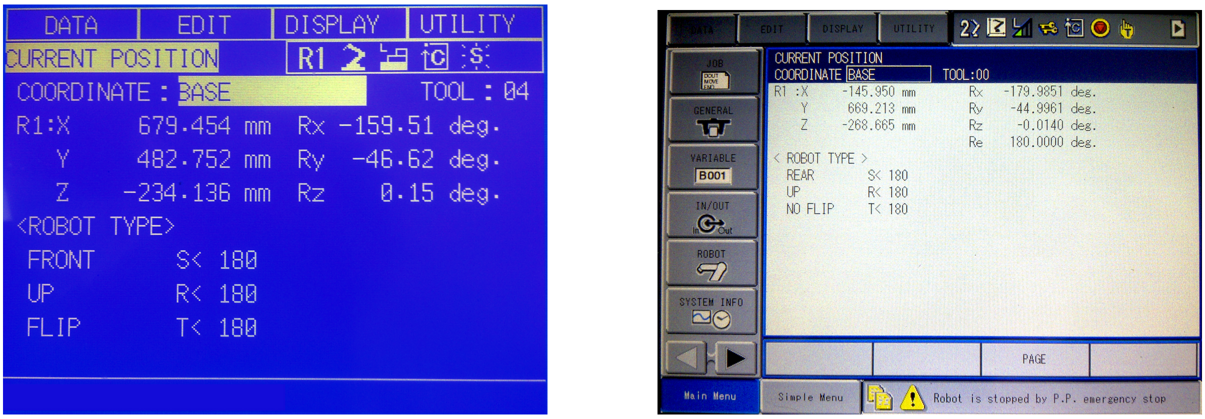
to move the robot in a controlled manner as well as position the end-effector at a specific coordinate value where it will remain whilst measurements are taken. This is the approach that will be used for all pose related evaluations. The pose evaluation consists of two aspects:

1. It evaluates how accurately an operator can manipulate the Master to a command point (Operator-Master evaluation)

2. It compares how accurately the Slave followed the Master's position (Master-Slave evaluation).

The Operator-Master evaluation makes use of external reference or command points (Figure 5.2b) to which the operator moves the Master's end-effector. For this the internal encoders of the Motoman robots can be utilised for calculating the accuracy and repeatability with which the operator can control the Master. To determine the actual command point encoder values, the Operator moves the end-effector via the programming pendant to each command point. At this position, the encoder values (in Cartesian format) is recorded. These command points remain fixed for the entire experimental procedure. The Operator then follows the exact same procedure, however, this time the end-effector is moved by the operators hand-eye coordination to as close to the command point as possible for n number of cycles. These attained points are then recorded, using the programming pendant display in Cartesian format. The evaluation then compares the attained coordinates to the command point coordinates. The evaluation will focus on command points P1, P2, P3, P4 and P5.

The Master-Slave evaluation involves comparing the Master's attained value to the Slave's attained value. To do this accurately, however, requires an external measurement system that does not merely compare the encoder values of the Master to that of the Slave after a move instruction. This is however the approach used as time and resource limitation prevent the evaluation from using external measurement systems. The Master-Slave evaluation therefore follows the same approach as the Operator-Master evaluation by moving the Master by hand-eye coordination to a command point, at which stage the attained coordinates are recorded from both programming pendant's display. Figure 5.3 illustrates an example of the programming pendant display for both controllers, from which to record the data. The measured data is used for both the accuracy and repeatability calculations.



(a) XRC Controller Display (b) DX100 Controller Display

Figure 5.3: Programming Pendant Displays

5.4.1 Operator-Master Accuracy

Operator-Master accuracy (OPA_i) is evaluated by comparing the attained points of the Master to the fixed command points on the physical test structure. The ISO 9283 provides Equations 5.4.1 to calculate the resulting positional accuracy in a 3-dimensional space [91]:

$$OPA_i = \sqrt{(\overline{x} - x_c)^2 + (\overline{y} - y_c)^2 + (\overline{z} - z_c)^2} \tag{5.4.1}$$

With x_c, y_c and z_c the command point coordinates of the point being evaluated, whilst $\overline{x}$, $\overline{y}$ and $\overline{z}$ are the mean values of the attained points, calculated according to Equation's 5.4.2 through 5.4.4. In these equations, x_j, y_j and z_j represent the coordinates for the j-th attained point, with n being the total number of cycles to be measured.

$$\overline{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{5.4.2}$$

$$\overline{y} = \frac{1}{n}\sum_{j=1}^{n} y_j \tag{5.4.3}$$

$$\overline{z} = \frac{1}{n}\sum_{j=1}^{n} z_j \tag{5.4.4}$$

5.4.2 Master-Slave Accuracy

Master-Slave accuracy (MSA_i) is evaluated by comparing the attained points of the Master to the attained points of the Slave. Unlike the Operator-Master accuracy evaluation, which makes use of a fixed command point, the Master-Slave accuracy compares two "varying" attained values. As such the individual coordinates are compared to determine the true positional error vector for each cycle. These positional error vectors are then used to determine the positional accuracy between the Master and the Slave according to Equation 5.4.5.

$$APC_i = \frac{1}{n}\sum_{i=1}^{n} e_{p_i} \tag{5.4.5}$$

With n the total number of measurements (cycles) taken regardless of command point, i, the individual measurement of a specific iteration of a command point and e_{p_i} the true vector error of the i-th iteration between the Master and Slave as calculated according to Equation 5.4.6

$$e_{p_i} = \sqrt{(A_{x_i} - B_{x_i})^2 + (A_{y_i} - B_{y_i})^2 + (A_{z_i} - B_{z_i})^2} \tag{5.4.6}$$

Where A_{x_i}, A_{y_i} and A_{z_i} are the attained coordinate values of the Master, whilst B_{x_i}, B_{y_i} and B_{z_i} represents the attained coordinate values for the Slave.

5.4.3 Positional Repeatability

Positional repeatability (RP_l) is determined by evaluating the closeness of the attained points after n repeat visits to the same command point. ISO 9283 provides Equation 5.4.7 through Equation 5.4.8 to calculate positional repeatability [91]:

$$RP_l = \bar{l} + 3S_l \tag{5.4.7}$$

With $\bar{l}$, the average distance between the barycentre of the attained points, calculated according to Equation 5.4.9 and Equation 5.4.10 respectively and S_l, the standard deviation of the attained points, determined using Equation 5.4.8. Note that $\overline{x}$, $\overline{y}$, $\overline{z}$ and x_j, y_j, z_j are defined in Section 5.4.1.

$$S_l = \sqrt{\frac{\sum_{j=1}^{n} \left(l_j - \bar{l}\right)^2}{n-1}} \tag{5.4.8}$$

$$\bar{l} = \frac{1}{n}\sum_{j=1}^{n} l_j \tag{5.4.9}$$

With

$$l_j = \sqrt{(x_j - \overline{x})^2 + (y_j - \overline{y})^2 + (z_j - \overline{z})^2} \tag{5.4.10}$$

5.5 Haptic Evaluation Procedures

The haptic evaluation requires that both haptic control and haptic feedback be assessed. Unlike the pose evaluations, where the Slave moves unencumbered through the air, haptic evaluation requires that the end-effector of the Slave experience forces and torques to determine how accurately the applied input force on the Master can be exercised by the Slave as well as to what degree haptic feedback can be perceived by the human operator. Two commercially available latex exercise bands will be utilized to apply forces and torques to the end-effector of both manipulators. Motion of the end-effector will, thus, either stretch (increasing the perceived force/torque) or relax (decreasing perceived force/torque) the latex band. The evaluation consists of two separate experiments, known as steady state and dynamic experiments.

Note that before any evaluation can be done, the sensors need to be calibrated first. This is accomplished by manually moving the Master and Slave to a position where the latex bands are clearly relaxed, after which the operator clicks the "Bias Sensors" button on the Java interface, refer to Figure 4.18, which calibrates both sensors readings and resets them to zero.

5.5.1 Steady State Procedure

The steady state experiment analyses the accuracy that is achievable by the Slave if the Master is experiencing a fixed haptic force. It provides insight into the performance characteristics of the control system before a more practical evaluation is conducted. Steady state refers to the point where the Master and Slave measured readings are equivalent. A steady state condition exists when the control system stops sending commands to the Master and Slave, which indicates that the desired haptic accuracy has been reached.

The haptic steady state evaluation will apply a pre-load to the Master, this pre-load is documented using the control sensor readings displayed on the developed graphical user interface as shown in Figure 4.18 in Section 4.5.2.1. The Master manipulator will remain stationary, thus, keeping the pre-load constant while the Slave moves to exercise the same force/torque. Once the telerobot reaches a steady state, the attained load is recorded from the feedback sensor's display on the graphical user interface. This attained value counts as a single data sample. The Slave is then manually reset to a new starting position and the sequence is repeated for n number of samples.

5.5.2 Dynamic Procedure

The dynamic experiment analyses the dynamic response of control system to determine the Slave following capabilities when the input force on the Master varies over time. It measures the ability of the telerobotic system to adapt to "random" input changes while providing an insight into the motion limitation of the Motoman robots when they are commanded from an external computer.

The dynamic evaluation starts from a steady state to prevent any initial force variations between the Master and Slave to influence the accuracy and repeatability of the Slave following characteristics. From this steady state, a pre-determined path is followed by the Master, using a combination of linear, joint and circular motion trajectories. This pre-determined path is programmed into the programming pendant of the Master to ensure that the path is highly repeatable, allowing the results between samples to be compared. The dynamic evaluation requires continuous data collection as the Master follows the pre-determined path. The data collection is accomplished using the "collecting" button on the graphical user interface as shown in Figure 4.18 in Section 4.5.2.1. The Master and Slave is then manually reset to the initial steady state starting position and the sequence is repeated for n number of samples.

Table 5.1 demonstrates the format of the collected data using the GUI. Where Id identifies the sensor data collected, $Sync$, identifies the Master-Slave synchronized combination used to calculate a command, whilst the rest specifies the individual force and torque readings in metric units (Newton and Newton-metre respectively) followed by the corresponding time stamp, in milliseconds, of when the data was recorded.

Table 5.1: GUI Data Collection Format Example

Id	Sync	Fx	Fy	Fz	Tx	Ty	Tz	Time
2	372	1.652	-3.269	4.840	-0.267	0.006	-0.026	1316540536567
1	373	5.674	7.066	16.981	0.024	0.125	-0.064	1316540536598
2	373	1.652	-3.269	4.840	-0.267	0.006	-0.026	1316540536660

5.5.3 Haptic Accuracy

Haptic accuracy (AH_i) is evaluated by comparing the attained force (or torque) to the set point force (or torque) vector. Note that for the steady state evaluation this set point is the applied pre-load value whilst for the dynamic evaluation this set point varies continuously as the applied input is not constant. For simplicity both evaluations will follow the same procedure used for determining the positional control accuracy in Section 5.4.2 with the exception of substituting the positional data for haptic data. Equations 5.5.1 and 5.5.2 can be derived and applied to calculate the resulting vector force (or torque) error and hence the accuracy of the haptic system in a 3-dimensional space:

$$AH_i = \frac{1}{m}\sum_{i=1}^{m} e_{h_i} \tag{5.5.1}$$

With m the total number of measurements taken, i, the individual measurement of a specific iteration and e_{h_i} the true vector error of the i-th iteration between the haptic input sensor and feedback sensor as calculated according to Equation 5.5.2

$$e_{h_i} = \sqrt{(A_{Hx_i} - B_{Hx_i})^2 + (A_{Hy_i} - B_{Hy_i})^2 + (A_{Hz_i} - B_{Hz_i})^2} \tag{5.5.2}$$

Where A_{Hx_i}, A_{Hy_i} and A_{Hz_i} are the attained force (or torque) vector component values of the haptic input manipulator whilst B_{Hx_i}, B_{Hy_i} and B_{Hz_i} represents the attained force (or torque) vector component values for the telerobot.

The haptic (force or torque) accuracy, consists of two components, a magnitude and a direction. The above two equations determine the magnitude accuracy. Equation 5.5.3 calculates the directional component's accuracy.

$$\theta_e = \frac{1}{n}\sum(\theta_{i_M} - \theta_{i_S}) \tag{5.5.3}$$

With θ_{i_M} and θ_{i_S} the x, y or z component angle of the Master and Slave respectively, each calculated by using Equation 5.5.4.

$$\theta_i = \frac{180}{\pi} \cos^{-1} \left(\frac{H_i}{H_R} \right) \tag{5.5.4}$$

Where H_i the attained force (or torque) vector component value in the x, y or z axis while H_R is the vector magnitude.

5.5.4 Haptic Repeatability

Haptic repeatability (RH_h) is determined by evaluating the closeness of the attained force or torque vectors after n repeat measurements for the same applied force/torque. Substituting the positional data of Equation 5.4.7 through Equation 5.4.10 in Section 5.4.3 with haptic data, Equation 5.5.5 through 5.5.8 can be derived and applied to calculate the force or torque repeatability:

$$RH_h = \overline{h} + 3S_h \tag{5.5.5}$$

With $\overline{h}$, the average distance between the barycentre of the attained points, calculated according to Equation 5.4.9 and Equation 5.5.8 respectively and S_h, the standard deviation of the attained points, determined using Equation 5.5.6. Note that $\overline{H_x}$, $\overline{H_y}$, $\overline{H_z}$ and H_{x_j}, H_{y_j}, H_{z_j} are defined in Section 5.5.3.

$$S_h = \sqrt{\frac{\sum_{j=1}^{n} \left(h_j - \overline{h}\right)^2}{n-1}} \tag{5.5.6}$$

$$\overline{h} = \frac{1}{n} \sum_{j=1}^{n} h_j \tag{5.5.7}$$

With

$$h_j = \sqrt{\left(H_{x_j} - \overline{H_x}\right)^2 + \left(H_{y_j} - \overline{H_y}\right)^2 + \left(H_{z_j} - \overline{H_z}\right)^2} \tag{5.5.8}$$

Chapter 6

Experimentation and Results

This chapter documents the results along with experimental findings of the evaluation procedures discussed in the previous chapter. It provides an understanding of the performance of the developed telerobotic system along with identifying its limitations. This chapter is divided into two main sections, pose and haptic performance respectively. The pose performance section discusses all elements related to performing positional related experiments while the latter discusses the results of the developed telerobotic system with regards to force control. Note that due to time restrictions, no torque evaluation is performed in this paper.

6.1 Pose Experimentation

Positional experiments follow the methodology described in Section 5.4. A prerequisite for performing pose experiments is the use of a physical test plane inside the workspace of the Master to provide the operator with a visual means of identifying the command points. A test structure is constructed according to the ISO 9283 standard and is shown in Figure 6.1.

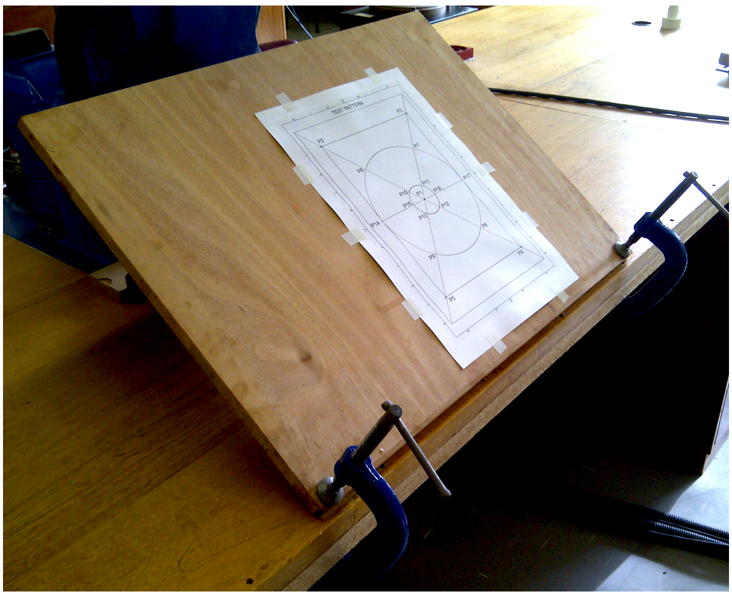


Figure 6.1: Diagonal Test Plane Structure

6.1.1 Defining the Command Points

Determining the command points makes use of the programming pendant of the Master to move a pencil (attached to the end-effector) to the desired position. At this position the Cartesian coordinates are recorded from the display of the programming pendant using a "fixed" coordinate system, as illustrated in Figure 4.12. All attained coordinates during experimentation make use of the same fixed coordinate system for comparative purposes. Table 6.1 summarizes the Cartesian coordinates of the command points P_1 through to P_5, using the Motoman robot's build-in fixed (base) coordinate system.

Table 6.1: Command Point Coordinates

Command Point	Measurement [mm] x	y	z
P1	728.548	58.882	-310.309
P2	637.252	-31.562	-239.441
P3	638.947	150.730	-242.123
P4	832.147	148.401	-389.770
P5	830.597	-33.972	-388.324

6.1.2 Pose Evaluation Equitability

The results of the minimum required measurements were obtained for each pose by using Equation 5.1.1 as described in Section 5.1. The pose evaluation makes use of 15 initial measurements to each of the 5 command points. These measurements are then standardized to ignore the associated command point by calculating the positional error vector for each measurement allowing the measurements to be combined to form a total of 75 initial samples.

The mean ($\overline{x}$) and standard deviation (σ) of the samples count is calculated and by applying a 5% confidence interval (k) and a 95% confidence level (T), the minimum required measurements (n) are calculated as 29.35 (hence 30 measurements). This minimum value is less than the 75 initial measurements suggesting that the pose evaluation is equitable. This indicates that there is a 95% surety that the mean error of the initial sample is within $\pm$ 5% of the mean of the population.

6.1.3 Pose Accuracy and Repeatability

Pose accuracy and repeatability calculations were computed using the methodology discussed in Section 5.4. The operator moves a pencil, attached to the end-effector of the Master, to the desired command point as accurately as possible as shown in Figure 6.2, after which the attained point is documented from the programming pendant of each Motoman robot. The attained points for command points P_1 through to P_5 are tabulated in Appendix B.

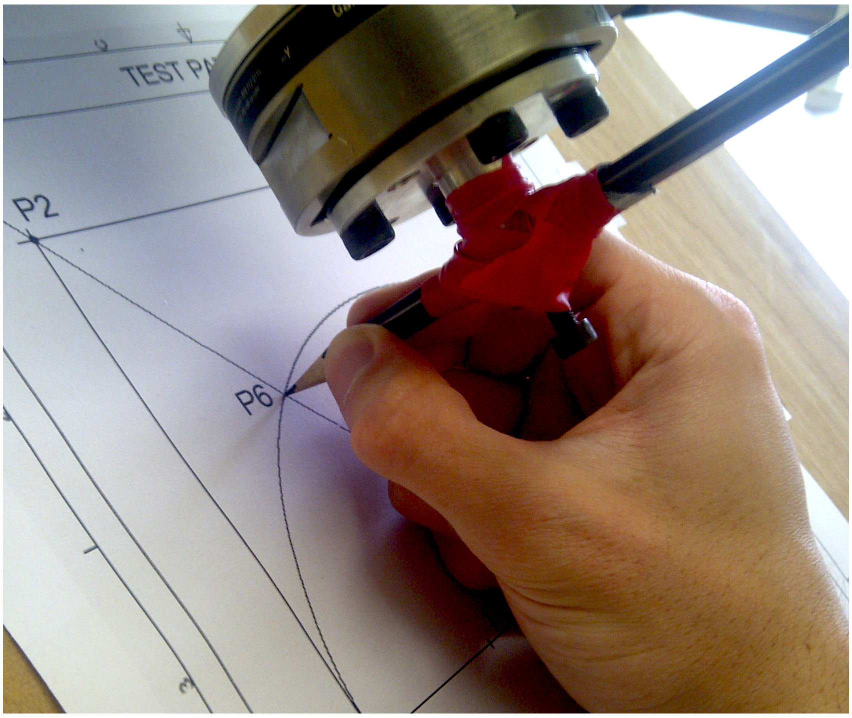


Figure 6.2: Operator Moving Input Manipulator

6.1.3.1 Accuracy Results

Using Equation 5.4.1 in Section 5.4.1, the accuracy with which the operator can move the Master to each command point has been calculated. These results are summarised in Table 6.2.

From the results it is clear that it is possible to move the robot by hand to any of the five

Table 6.2: Pose Accuracy Results: Operator-Master

Point	**Coordinate Accuracy [mm]**			**OPA [mm]**
	$\overline{x}$	$\overline{y}$	$\overline{z}$	
P1	-0.491	-0.512	-0.499	0.867
P2	-0.516	-0.527	-0.528	0.907
P3	-0.504	-0.474	-0.517	0.864
P4	-0.508	-0.499	-0.503	0.872
P5	-0.499	-0.488	-0.513	0.866
Average	**-0.504**	**-0.500**	**-0.512**	**0.875**

command points with an average accuracy of less than 1mm (0.875mm). The marginal accuracy variation between command points is approximately 0.04mm. This indicates that the Master can be manipulated manually to any coordinate within the Master's work envelope while still achieving the same result. Using Equation 5.4.5 in Section 5.4.2, the accuracy with which the Slave follows the haptic input manipulator has been calculated. These results are summarised in Table 6.3.

Table 6.3: Pose Accuracy Results: Master-Slave

Command Point	APC [mm]
P1	0.042
P2	0.052
P3	0.053
P4	0.058
P5	0.044
Average	**0.050**

The average accuracy achieved by the Slave with regards to following the Master is 0.05mm, meaning that the position of the Slave differs from that of the Master manipulator by approximately one twentieth of a millimetre. Note that this result is based on comparing the variation of each robot's internal encoder values before and after the Master and Slave have moved. This is not a clear indication of the true accuracy between the robotic manipulators end-effectors though as the robot precision is not taken into account. To obtain a true accuracy result, it will be required to use an external measurement system.

6.1.3.2 Repeatability Results

Using Equation 5.4.7 in Section 5.4.3, the repeatability with which the operator could move the Master to each command point has been calculated. Subsequently, the repeatability of the Slave has been calculated. These results are summarised in Table 6.4 and Table 6.5 respectively.

Table 6.4: Pose Repeatability Results - Master

Command Point	RPI [mm]
P1	0.638
P2	0.621
P3	0.561
P4	0.489
P5	0.613
Average	**0.584**

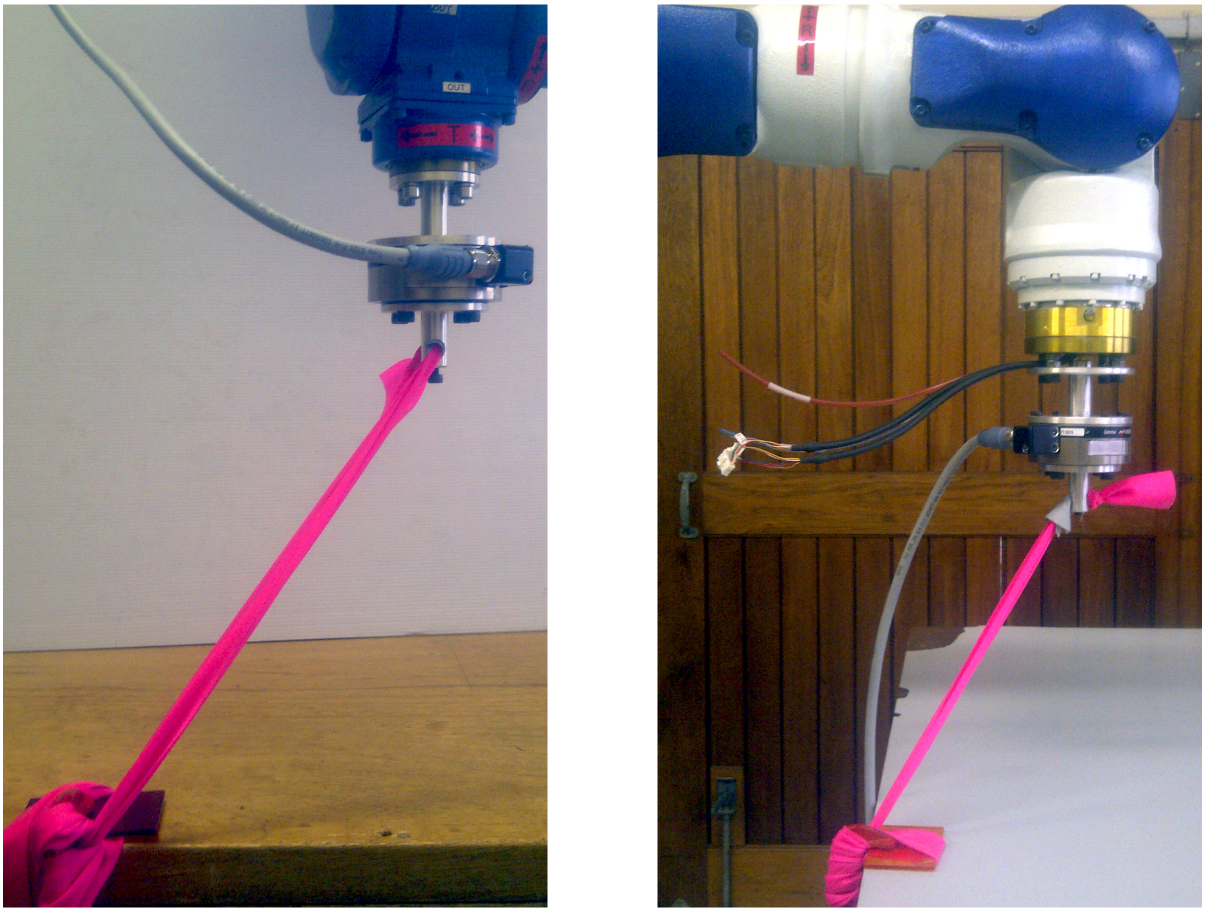
Comparing the repeatability of Master to the Slave manipulator, both similar industrial robots fitted with the same sensors, it is clear that the inclusion of human elements in the Master manipulator adversely effect the repeatability of the system. With a repeatability of 0.584mm, the Slave repeatability is approximately five times better at 0.116mm. This value is also evidently very close to its expected repeatability which is 0.1mm as stated by the Motoman product manuals in Appendix I.

Table 6.5: Pose Repeatability Results - Slave

Command Point	RPC [mm]
P1	0.113
P2	0.106
P3	0.133
P4	0.112
P5	0.115
Average	**0.116**

6.2 Force Experimentation

The force evaluation makes use of latex exercise bands fitted to the Master and Slave robots to effectively apply a controlled force depending on the current position of the robot end-effectors. The experimental set-up is demonstrated in Figure 6.3 with each latex band (pink coloured band in images) attached to the end-effector of the Master and Slave device on the one end and fixed to a rigid table on the other. This causes any motion of the Motoman end-effectors to either stretch or relax the latex band which varies the force experienced by the F/T sensors.



(a) Master band (b) Slave band

Figure 6.3: Haptic Evaluation using Latex bands

Note that the force vector always points towards the fixed position on the table. Practically this would insinuate that the end-effector should move towards that point if no other external force is applied. This would be the case, however, to perform the experiments in a controlled manner which allows the data to be compared, the Master is controlled via the Motoman programming pendant to either remain stationary, as is the case of the steady state evaluation, or follow a fixed path at various speeds, the latter being the case for the dynamic evaluation. In both steady state and dynamic evaluation the Slave is free to move (or respond) to the applied force, with the end goal of mimicking the applied force on the Master. Note that the Slave's motion is not pre-programmed as is the case with the Master. The control system described in Section 4.4 determines the appropriate command which is transmitted to the Slave.

6.2.1 Force Evaluation Equitability

The results of the minimum required measurements were calculated by using Equation 5.1.1, as described in Section 5.1, for steady state and dynamic evaluations respectively. Both steady state and dynamic evaluations make use of 15 initial data samples. For the steady state analysis, the data samples are the instantaneous difference between the Master and Slave force readings (further known as the force error) when a steady state, as described in Section 5.5, has been reached. For the dynamic analysis, which contains a wide variety of force readings over the duration of the experiment, a collective view of the collected force readings, i.e. a mean of the individual force errors, is used to represent the data sample.

The mean ($\overline{x}$) and standard deviation (σ) of the data samples are calculated and by applying a 5% confidence interval (k) and a 95% confidence level (t), the minimum required measurements (n), for both evaluations are calculated. These results are tabulated in Table 6.6. The minimum value of both evaluations is less than the 15 initial measurements, indicating that there is a 95% surety that the mean error of the initial sample is within ± 5% of the mean of the population. This establishes that the available data is sufficient in making the forcce evaluations equitable.

Table 6.6: Minimum Number of Samples for Force Evaluation

Haptic Evaluation	Required Samples [n]
Steady State	13
Dynamic	4

6.2.2 Steady State Results

Using Equation 5.5.1 in Section 5.5.3, the accuracy with which the Slave can apply a force vector equivalent to that experienced by the Master has been calculated. Subsequently, the repeatability has been calculated using Equation 5.5.5. These results are summarised in Table 6.7. A sample of a data set is available in Appendix C, whilst the full steady state experimental data set can be found on the attached CD at the back of the report.

Table 6.7: Steady State Force Results

Accuracy		Min	Max	Mean
Magnitude [N]		0.107	0.147	0.123
Direction [deg]	x	0.000	0.123	0.038
	y	0.001	0.267	0.057
	z	0.001	0.191	0.069

The results indicate that the Slave can mimic the applied force experienced by the Master with an mean accuracy of 0.123N (in magnitude) with a mean directional error of less than 0.07 degrees in either x, y or z axis after a steady state condition has been reached. The repeatability of the Steady State evaluation is calculated as 0.034N, indicating the degree to which the result of the various samples differed. These two results effectively mean that the the Slave will mimic the force experienced by the Master with a resulting error of 0.123 ± 0.034 Newtons once a steady state condition has been achieved.

Figure 6.4 illustrates the resulting force vector magnitude of performing the steady State analysis[1]. Where the pre-load applied to the Master is kept constant throughout the experiment while the Slave responds by moving the end-effector until a steady state is reached. Note that with the latex band attached to the end-effector, a change in force indicates that the Motoman robot is moving. Thus, the steady state is the point where both Master and Slave manipulators are stationary, or rather the point where the pre-load force on the Master is applied by the Slave as shown in the figure.

The steady state evaluation only looks at a specific point on the graph to calculate its accuracy and repeatability, however, by considering the entire graph further observations can be made. One of which is the response rate of the system. Initially it responds quickly to the large difference between the Master and Slave measurements, but as the difference becomes smaller so too does the response slow down. This is expected as the control system compares the force error (force difference between the Master and Slave) to calculate the next movement command to send to the Slave. So for force error change (as is the case in Figure 6.4), the succeeding command is adjusted to incorporate the new force readings. Thus, a smaller force error leads to a shorter move command at a lower velocity, effectively resulting in the attained response of the Slave.

6.2.3 Dynamic Results

Using Equation 5.5.1 in Section 5.5.3, the force accuracy with which the Slave can follow the the Master has been calculated. Subsequently, the repeatability has been calculated using Equation 5.5.5. These results are summarised in Table 6.8 with Figure 6.5 graphically demonstrating the following capabilities of the Slave[2]. Note that a sample of a data set is available in Appendix D, whilst the full dynamic experimental data set can be found

[1] The steady-state force component (Fx, Fy and Fz) graphs are given in Appendix E

[2] The dynamic force component (Fx, Fy and Fz) graphs are given in Appendix E

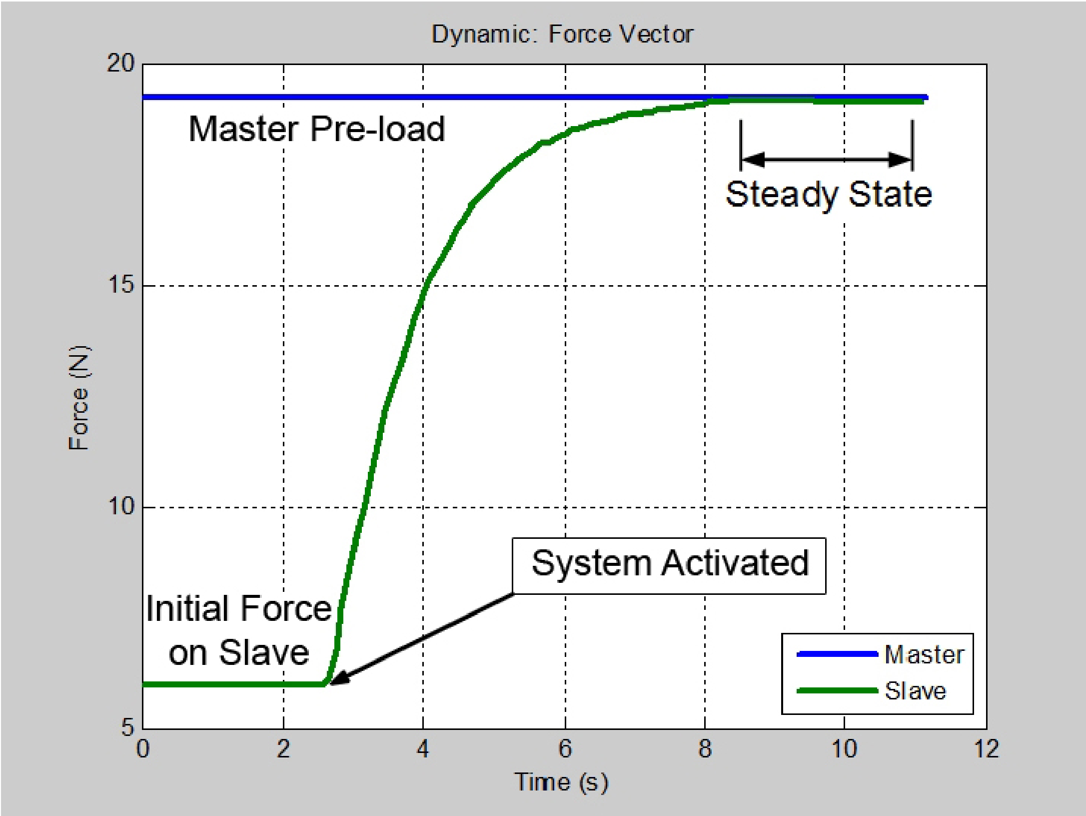


Figure 6.4: Steady State Force Vector Results Graph

on the attached CD at the back of the report.

Table 6.8: Dynamic Force Results

Accuracy		Min	Max	Mean
Magnitude [N]		0.091	6.567	1.639
Direction [deg]	x	0.034	7.402	2.862
	y	0.020	15.015	3.865
	z	0.001	14.973	4.081

The results indicate that the control system was capable of manipulating the position of the Slave so that the Slave could mimic the applied force on the Master with a mean accuracy of 1.639N (in magnitude) with a mean directional error of less than 4.1 degrees in either x, y or z axis. Note that there is quite a large difference between the minimum, maximum and mean values obtained during the experiment. Furthermore, these results are consistently achievable with a calculated repeatability of 0.021N. This means that there is practically no visible difference between the various data sets and that Figure 6.5 is a good indication of the overall dynamic results.

Graphically analysing the data provides a greater understanding of why there is such a

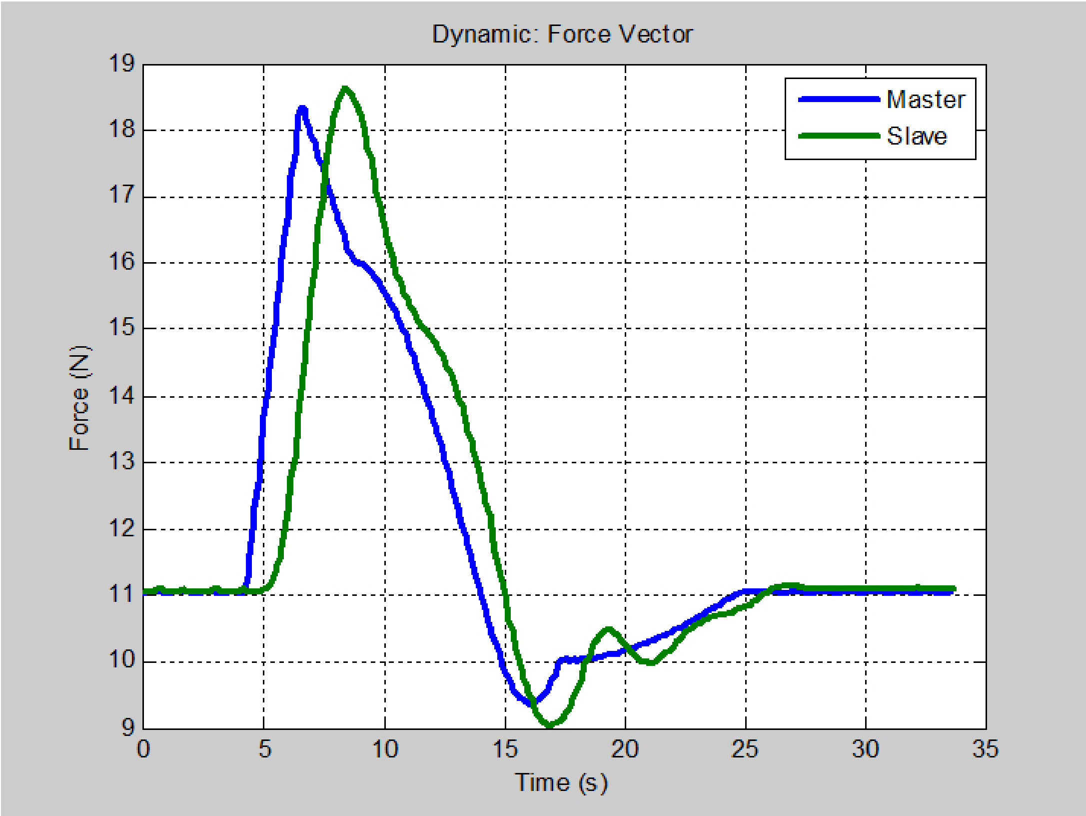


Figure 6.5: Dynamic Experimental Results Graph

large difference between the minimum and maximum accuracy results. While the graph suggests that the Slave follows the Master, there is a delay of approximately 1.2 seconds in the Slave's response (indicated by the green line shifted slightly to the right). This delay is the main contributing factor to the large accuracy variation over the duration of the experiment as it makes use of the instantaneous force error between Master and Slave and not the resulting applied error. Ignoring the delay will therefore provide a clearer picture of the actual perceived force accuracy of the system. Figure 6.6 demonstrates the scenario where this delay is disregarded, with the resulting accuracy documented in Table 6.9.

The newly attained result in Table 6.9 provides close to an 81% improvement to the maximum force error, lowering it from 6.567N to 1.218N while the mean force error improved from 1.639N to 0.208N, an increase in accuracy of aproximately 88%. This indicates that the actual force following capabilites of the Slave is very good, albeit only after a delay of 1.2 seconds. Note that these results are based on an experimental procedure which falls within the control capabilities of the system, meaning that the Master is not controlled in such a manner which makes it impossible for the Slave to follow or that the applied force does not vary significantly more than once within that 1.2 second delay period. These limitations along with other findings are further discussed in the next section.

6.2.4 Dynamic Limitations

The dynamic evaluation poses an interesting result. It is found that the Slave experiences a delay of about 1.2 seconds before achieving the perceived force of the Master. Now this

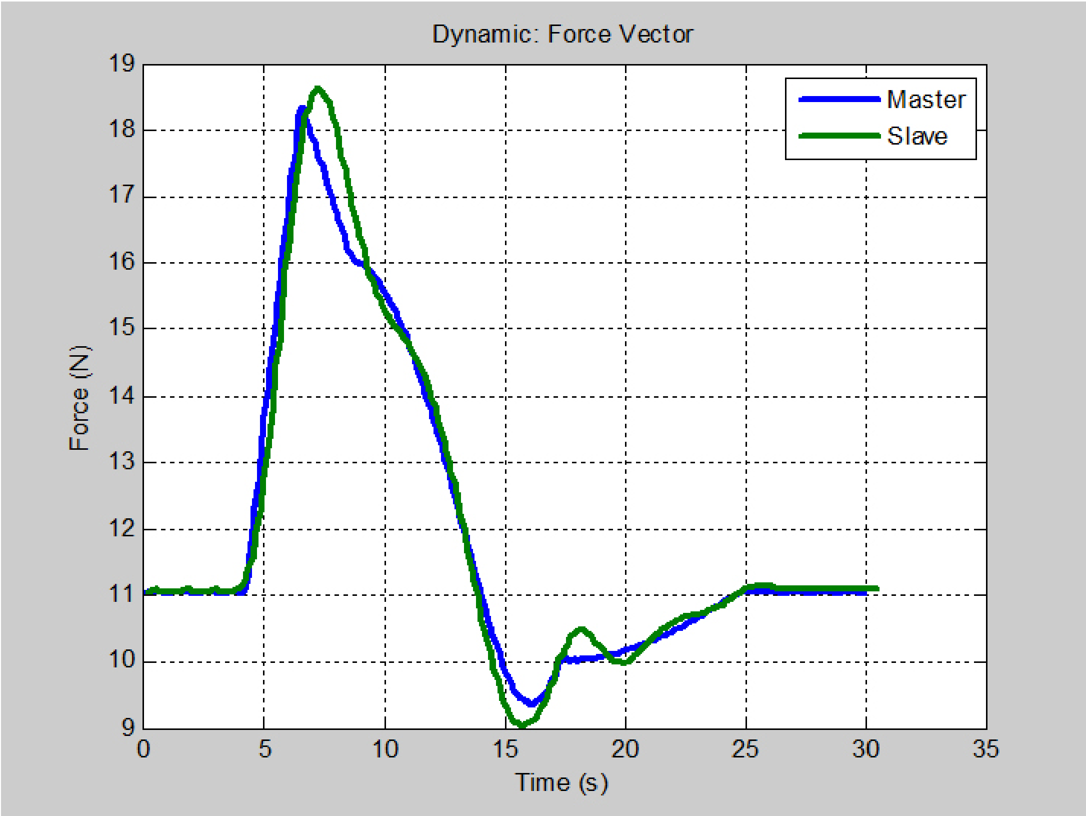


Figure 6.6: Dynamic Experimental Results: Shifted Response

Table 6.9: Dynamic Results: Shifted Response

Experiment	Results [N] Min	Max	Mean
Accuracy	0.001	1.218	0.208

opens up a number of questions, one of which is how the response of the Slave is affected if the Master experiences sudden changes during that delayed interval. Figure 6.7 illustrates such an event in the highlighted area, L_1.

What can be seen is that the Slave slightly overshoots the initial Force magnitude change due to this delay, however, more importantly, the second force change is completely ignored. This is due to how and when the control system calculates the next command to issue to the Slave. Each command sent to the Slave makes use of the most up to date sensory readings, however, a command is only sent once the Motoman robot responds to the Java interface that it is ready to process the next command. Thus, any force change that occurs during the time in which Slave is unable to receive the next command effectively results in a loss of that sensory information. The Slave does not respond to such changes simply because the control system continuously updates the succeeding command with the latest sensory information until it is sent to the Slave. While it is possible to record all these commands with the aim of following the Master to a finer degree, this will result in a further, more significant delay which is not desirable for real-time control applications. This restriction is therefore a hardware limitation as the Motoman robots are unable to

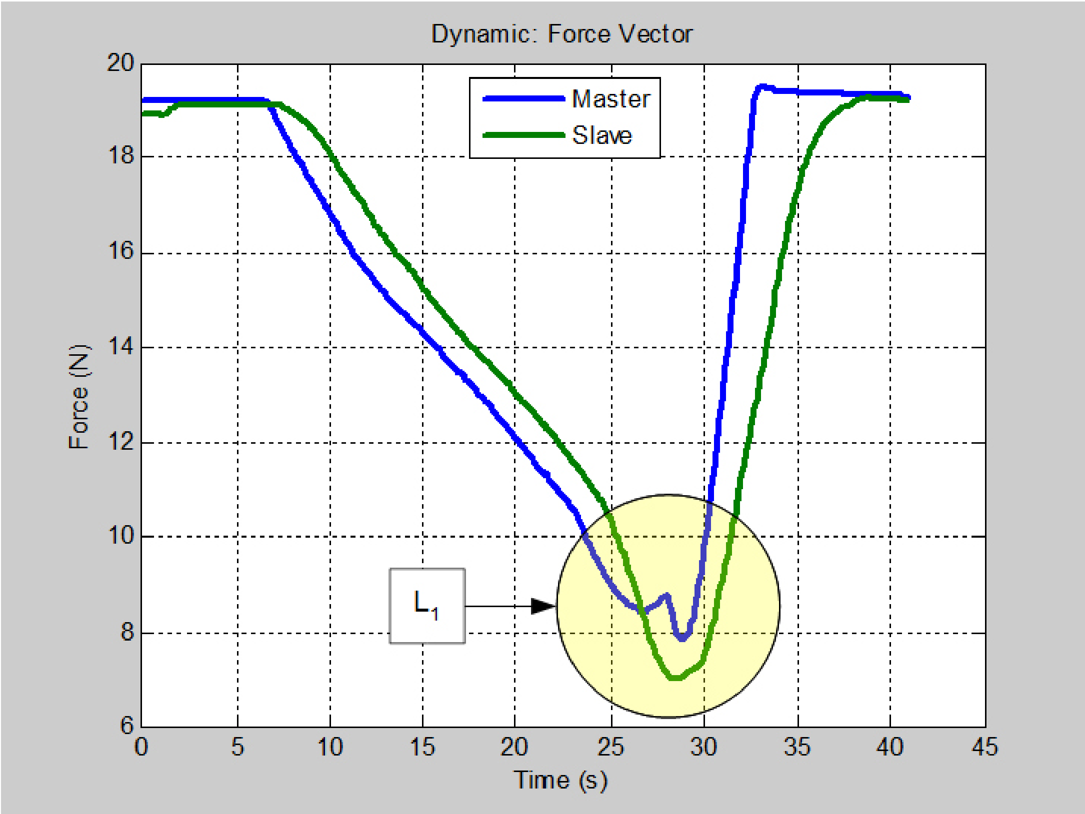


Figure 6.7: Limitation Example A

be controlled in real-time from an external host controller (computer). This limitation is discussed in detail in Section 4.6.

Another aspect to consider is the speed with which the Slave can respond to sudden large magnitude changes. Both Motoman robots have the same experimental set-up (F/T sensor fitted to the end-effector and attached to a latex band) and roughly the same velocity capabilities, however, the highlighted area, L_2 in Figure 6.8 demonstrates a sudden large magnitude force change on the Master, with the Slave unable to respond at the same rate. This is again due to the inability to control the Motoman robots in real time, resulting in the trade off between response speed and accuracy as described in Section 4.6.

The final issue discovered during both pose and forcce experimental procedures is a communication concern where a Motoman robot would occasionally fail to respond, thus, remaining stationary regardless of the applied force or motion of the other manipulator. This scenario is illustrated at the highlighted area, L_3 in Figure 6.8. Upon closer inspection it was found that this is due to a break in communication during the Motoman handshaking protocol, specifically Step 6 in Figure 4.8, in Section 4.3.4.1 where the Motoman controller would fail to respond back to its associated barionet communication device. After this occurs the Motoman controller becomes non-responsive and will issue a warning flag if communication tries to continue. The communication to the Motoman robot is in effected suspended, preventing new commands to be sent. This phenomena seems to occur at random intervals, some ranging from a few seconds after the telerobotic system is activated to only after several minutes of operation. If this occurs, the telerobotic system (Java interface, barionet devices and Motoman robots) need to be rebooted

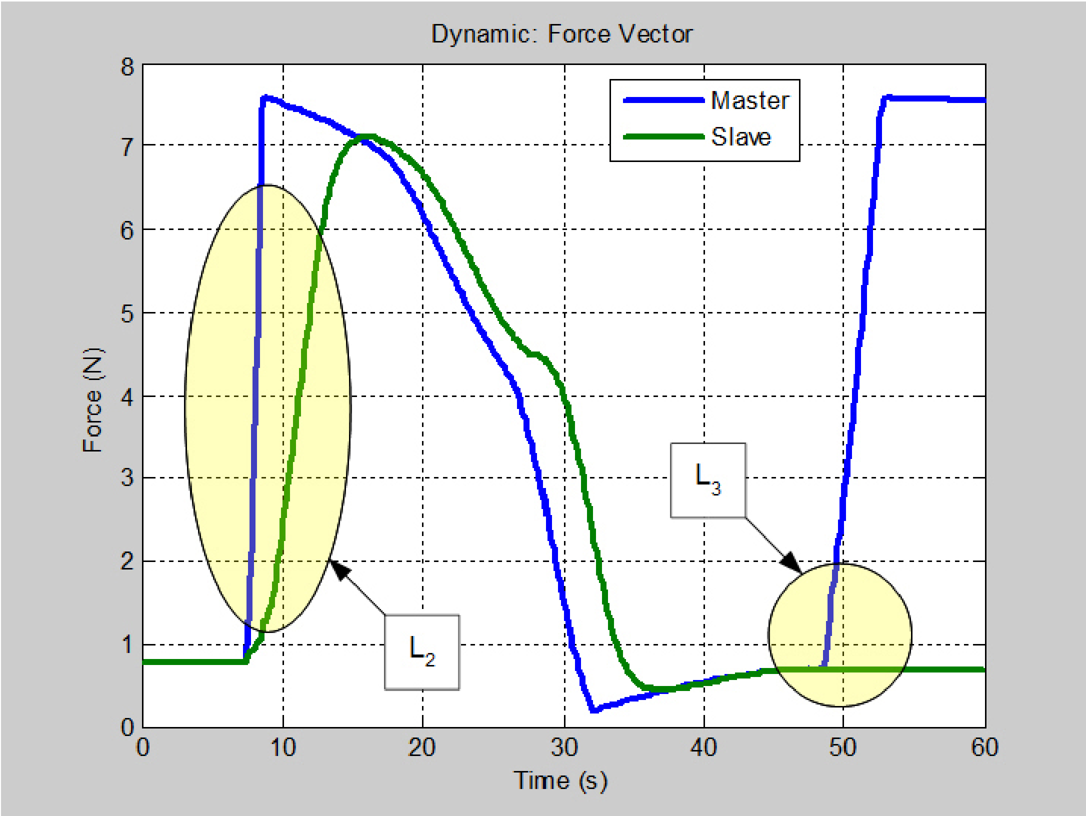


Figure 6.8: Limitation Example B

before normal operation can commence. A solution as to why this break in communication occurs could not be found, however, it seems to only occur on the "older" Motoman XRC controller. All experimental procedures therefore disregarded the attained data in which this communication error occurred and the sample evaluation was subsequently repeated to obtain usable results.

Chapter 7

Conclusion and Recommendations

A standard industrial robot operates remarkably well in a controlled repetitive environment but lacks the ability to adapt within a dynamic environment. Telerobotic systems on the other hand are suited for dynamic environments as they incorporate a human operator into the control loop. Visual and haptic feedback improves the operator's perception of the remote environment, allowing greater dexterity in controlling a telerobot. This project sets out to develop a telerobotic system that would allow an operator to control a standard industrial robot (Slave) in an dynamic remote environment via another industrial robot (Master) in the operator environment. This thesis describes the development and implementation of such a Master-Slave telerobotic system that allows an operator to move the Master in an intuitive manner by using his hand-eye coordination while receiving force feedback from the Slave.

To develop a quality system, it is important to consider the entire life cycle of the proposed telerobotic system. This project forms part of the first attempt at applied telerobotic research at the University of Stellenbosch and due to time and resource constraints, a full life cycle design was not feasible. The scope of this thesis is, thus, limited to the initial stages of the product life cycle. The design process discussed in Section 3 provides a systems engineering approach that directed the development, implementation and documentation phase of the project to ensure that the research objectives in Section 1.3 could be met while adhering to the projects constraints in Section 1.4. The design process, in reality, is an on-going iterative process that requires continuous refinement. This thesis documents one such iteration.

This project made use of two similar industrial robotic manipulators, a Motoman UP6 and a Motoman SDA10D. Each robot was fitted with a Net F/T Gamma multi-axis force and torque sensor in Section 4.2 to transform them into haptic enabled devices. One of which is used as a haptic input device (Master), while the other assumed the role of the telerobot (Slave), which would provide haptic feedback to the operator.

Subsequently an Ethernet-based, bilateral communication system was developed in Section 4.3 that made use of a central computer and two Barix Barionet 100 devices to connect the operator environment to the remote environment. A Java application running on the computer is responsible for receiving the force and torque readings from both F/T sensors while transmitting a robot instruction to the Motoman robot's respective Barionet communication device.

A position based force feedback control system was developed that allowed an operator to control the Master using his hand-eye coordination in an intuitive manner in Section 4.4. The developed controller consists of a feedback logic unit and an intuitive logic unit as shown in Figure 4.14. The feedback logic unit combines the sensor readings to incorporate the forces and torques experienced in the remote environment before the intuitive logic unit transforms the resulting input into a robot instruction (IMOV) that moves both Master and Slave.

The control system was integrated into the communication system's Java interface and extended to provide the operator with a graphical user interface (Section 4.5.2). The developed Java interface made use of concurrent processes to displayed the video feedback to the operator, receive force and torque data from both F/T sensors, determine a suitable robot instruction in the control system as well as transmit the instruction to the Master and Slave's respective Barionet devices simultaneously.

The accuracy and repeatability of the positioning and force following capabilities of the developed telerobotic system has been evaluated in Section 6 according to the procedures discussed in Section 5. The positioning results indicate that an operator was able to move the Master to within 0.875±0.584mm of the desired point whilst the Slave was able to follow the Master with a much higher positional accuracy and repeatability at 0.050±0.116mm. Note that the steady state force results, that measures the accuracy with which the Slave can mimic the force applied on the Master when time is not considered, was found to be 0.123±0.034 Newtons with a mean directional error of less than 0.07 degrees in either x, y or z axis. The dynamic results, which measures the Slave's ability to follow a varying input force in real-time indicates a much lower force accuracy result at 1.639±0.021N with a mean directional error of less than 4.1 degrees in either x, y or z axis.

The poor dynamic result is due to hardware limitations of the Motoman robot used. These limitations cause control related issues such as jittery motion, communication delays and adverse effects when sudden or large input forces are considered. While the developed telerobotic system allows an operator to control the Master in a intuitive manner using his hand-eye coordination, these issues continuously remind the operator that he or she is not only manipulating an object (such as a pencil or scalpel), but in fact a powerful industrial robotic manipulator.

List of References

[1] Thompson, J.M., Ottensmeyer, M.P., Sheridan, T.B. and D, S.: Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with local manipulative assistance. *Telemed Journal*, vol. 5, pp. 129–137, 1999.

[2] UNECE: World robotics 2004 (and 2006). [Online], 2004.
Available at: `http://www.unece.org/press/pr2004/04robots_index.htm.`

[3] Torgny and Brogårdh: Present and future robot control development - an industrial perspective. *Annual Reviews in Control*, vol. 31, no. 1, pp. 69 – 79, 2007. ISSN 1367-5788.

[4] Bloom, M.B., Salzberg, A.D. and Krummel, T.M.: Advanced technology in surgery. *Current Problems in Surgery*, vol. 39, no. 8, pp. 33 – 830, 2002. ISSN 0011-3840.

[5] Curley, K.C.: An overview of the current state and uses of surgical robots. *Operative Techniques in General Surgery*, vol. 7, no. 4, pp. 155 – 164, 2005. ISSN 1524-153X. Imaging and Robotics in Surgical Practice.

[6] EN ISO 8373: Manipulating industrial robots - vocabulary. 1994.

[7] Wallén, J.: The history of the industrial robot. Tech. Rep. LiTH-ISY-R-2853, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, May 2008.

[8] RobotWorx: Industrial robot parts - arm, controller, drives, sensors. [Online], 1999.
Available at: `http://www.robots.com/faq/18/what-are-the-main-parts`

[9] Westerlund, L.: In: *The extended arm of man: a history of the industrial robot*. Informationsforlaget, Stockholm, Sweden, 2000. ISBN 9177364678, 9789177364672.

[10] Moran, M.: Evolution of robotic arms. *Journal of Robotic Surgery*, vol. 1, pp. 103–111, 2007. ISSN 1863-2483.

[11] Luh, J.: An anatomy of industrial robots and their controls. *Automatic Control, IEEE Transactions on*, vol. 28, no. 2, pp. 133 – 153, feb 1983. ISSN 0018-9286.

[12] Blomdell, A., Bolmsjo, G., Brogardh, T., Cederberg, P., Isaksson, M., Johansson, R., Haage, M., Nilsson, K., Olsson, M., Olsson, T., Robertsson, A. and Wang, J.: Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *Robotics Automation Magazine, IEEE*, vol. 12, no. 3, pp. 85 – 94, sept 2005. ISSN 1070-9932.

[13] Zelinsky, A.: Robotics: Modeling, planning, and control (siciliano, b. et al; 2009) [on the shelf]]. *Robotics Automation Magazine, IEEE*, vol. 16, no. 4, p. 101, december 2009. ISSN 1070-9932.

[14] Brussel, H.V.: Evaluation and testing of robots. *CIRP Annals - Manufacturing Technology*, vol. 39, no. 2, pp. 657 – 664, 1990. ISSN 0007-8506.

[15] Pan, Z., Polden, J., Larkin, N., Duin, S.V. and Norrish, J.: Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87 – 94, 2012. ISSN 0736-5845.

[16] Britannica, E.: Tele-operation. [Online], 2010.
Available at: `http://encyclopedia.jrank.org/articles/pages/6910/Tele-Operation.html`

[17] Britannica, E.: Telepresence. [Online], 2010.
Available at: `http://encyclopedia.jrank.org/articles/pages/6911/Telepresence.html`

[18] Ferre, M., Buss, M., Aracil, R., Melchiorri, C. and Balaguer, C.: *Advances in Telerobotics*. Springer Publishing Company, Incorporated, 2007. ISBN 3540713638, 9783540713630.

[19] Hokayem, P.F. and Spong, M.W.: Bilateral teleoperation: An historical survey. *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006. ISSN 0005-1098.

[20] Opperman, G.: Micro-scale machine vision using a robot mounted camera in a google earth manner. Final year project, Stellenbosch University, 2009.

[21] Ferrell, W.R.: Remote manipulation with transmission delay. 1964.
Available at: `http://hdl.handle.net/1721.1/11065`

[22] Ferrell, W.R. and Sheridan, T.B.: Supervisory control of remote manipulation. *Spectrum, IEEE*, vol. 4, no. 10, pp. 81 –88, oct 1967. ISSN 0018-9235.

[23] Fong, C., Dotson, R. and Bejczy, A.: Distributed microcomputer control system for advanced teleoperation. In: *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 987 – 995. apr 1986.

[24] Lee, S., Bekey, G. and Bejczy, A.: Computer control of space-borne teleoperators with sensory feedback. In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 205 – 214. mar 1985.

[25] Sato, T. and Hirai, S.: Language-aided robotic teleoperation system (larts) for advanced teleoperation. *Robotics and Automation, IEEE Journal of*, vol. 3, no. 5, pp. 476 –481, october 1987. ISSN 0882-4967.

[26] Bejczy, A. and Kim, W.S.: Predictive displays and shared compliance control for time-delayed telemanipulation. In: *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, vol. 1, pp. 407–412. jul 1990.

[27] Bejczy, A., Kim, W. and Venema, S.: The phantom robot: predictive displays for teleoperation with time delay. In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 546 –551 vol.1. may 1990.

[28] Sheridan, T.: Telerobotics. *Automatica*, vol. 25, no. 4, pp. 487 – 507, 1989. ISSN 0005-1098.

[29] Stafford, F.Q.: The trojan room coffee pot. [Online], 1995.
Available at: `http://www.cl.cam.ac.uk/coffee/qsf/coffee.html`

[30] Liu, P., Meng, M. and Yang, S.: Data communications for internet robots. *Autonomous Robots*, vol. 15, no. 3, pp. 213–223, 2003.

[31] Wei, W. and Kui, Y.: Teleoperated manipulator for leak detection of sealed radioactive sources. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, pp. 1682 – 1687 Vol.2. 26-may 1, 2004. ISSN 1050-4729.

[32] Funda, J. and Paul, R.: A symbolic teleoperator interface for time-delayed underwater robot manipulation. In: *OCEANS '91. 'Ocean Technologies and Opportunities in the Pacific for the 90's'. Proceedings.*, pp. 1526 –1533. oct 1991.

[33] Hirzinger, G., Brunner, B., Dietrich, J. and Heindl, J.: Sensor-based space robotics-rotex and its telerobotic features. *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 649 –663, oct 1993. ISSN 1042-296X.

[34] Funda, J., Taylor, R., Eldridge, B., Gomory, S. and Gruben, K.: Constrained cartesian motion control for teleoperated surgical robots. *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 3, pp. 453 –465, jun 1996. ISSN 1042-296X.

[35] Diolaiti, N. and Melchiorri, C.: Teleoperation of a mobile robot through haptic feedback. In: *Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE*, pp. 67 – 72. 2002.

[36] Zeng, G. and Hemami, A.: An overview of robot force control. *Robotica*, vol. 15, no. 5, pp. 473–482, 1997.

[37] Fite, K. and Goldfarb, M.: Position control of a compliant mechanism based micromanipulator. In: *ICRA'99*, pp. 2122–2127. 1999.

[38] Volpe, R. and Khosla, P.: A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Transactions on Automatic Control*, vol. 38, pp. 1634–1650, 1993.

[39] Raibert, M.H. and Craig, J.J.: Hybrid position/force control of manipulators. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, vol. 103, 1981.

[40] Hogan, N.: Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, March 1985.

[41] O'Malley, M.K. and Gupta, A.: Haptic interfaces. In: *HCI Beyond the GUI*, pp. 25 – 73. Morgan Kaufmann, San Francisco, 2008. ISBN 978-0-12-374017-5.

[42] Seraji, H.: Adaptive admittance control: An approach to explicit force control in compliant motion. In: *ICRA'94*, pp. 2705–2712. 1994.

[43] Kim, V., Chapman, W., Albregcht, R., B.M., B., Young, J., L.W, N. and Chitwood, W.: Early experience with telemanipulative robot-assisted laparoscopic cholecystectomy using da vinci. *Surgical Laparoscopy Endoscopy and Percutaneous Techniques*, vol. 12, no. 1, pp. 33–40, 2002.

[44] Senapati, S. and Advincula, A.: Telemedicine and robotics: Paving the way to the globalization of surgery. *International Journal of Gynecology and Obstetrics*, vol. 91, no. 3, pp. 210 – 216, 2005. ISSN 0020-7292.

[45] Smith, C.M.: Human factors in haptic interfaces. *Crossroads*, vol. 3, pp. 14–16, April 1997. ISSN 1528-4972.

[46] Wall, S.A. and Harwin, W.: A high bandwidth interface for haptic human computer interaction. *Mechatronics*, vol. 11, no. 4, pp. 371 – 387, 2001. ISSN 0957-4158.

[47] Klomp, F.: Haptic control for dummies: An introduction and analysis. Master's thesis, Eindhoven University of Technology, 2006.

[48] Shimoga, K.: Finger force and touch feedback issues in dexterous telemanipulation. In: *Intelligent Robotic Systems for Space Exploration, 1992. Proceedings. Fourth Annual Conference on*, pp. 159 –178. sep- 1 oct 1992. ISSN 1063-7117.

[49] Sutter, P.H., Iatridis, J. and Thakor, N.: Space telerobotics, 1989. proceedings., nasa conference on. pp. 157–66. 1989.

[50] Brooks, T.: Telerobotic response requirements. In: *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, pp. 113 –120. nov 1990.

[51] Astin, A.: *Finger force capability: measurement and prediction using anthropometric and myoelectric measures*. Master's thesis, Faculty of the Virginia Polytechnic Institute and State University, 1999.

[52] Hallbeck, M.: Flexion and extension forces generated by wrist-dedicated muscles over the range of motion. *Applied Ergonomics*, vol. 25, no. 6, pp. 379 – 385, 1994. ISSN 0003-6870.

[53] Jones, L.: Matching forces: constant errors and differential thresholds. *Perception*, vol. 15, pp. 681–687, 1989.

[54] Tan, H.Z., Srinivasan, M.A., Eberman, B. and Cheng, B.: Human factors for the design of force-reflecting haptic interfaces. *Dynamic Systems and Control*, vol. 55, no. 1, pp. 353–359, 1994.

[55] Pang, X., Tan, H. and Durlach, N.: Manual discrimination of force using active finger motion. *Attention, Perception & Psychophysics*, vol. 49, pp. 531–540, 1991. ISSN 1943-3921.

[56] Park, K., Bae, B. and Koo, T.: A haptic device for pc video game application. *Mechatronics*, vol. 14, no. 2, pp. 227 – 235, 2004. ISSN 0957-4158.

[57] Sensable: Phantom premium 6dof. [Online], 2011.
Available at: `http://www.sensable.com/haptic-phantom-premium-6dof.htm`

[58] Sone, J., Inoue, R., Yamada, K., Nagae, T., Fujita, K. and Sato, M.: Development of a wearable exoskeleton haptic interface device. *Journal of Computing and Information Science in Engineering*, vol. 8, no. 4, p. 041009, 2008.

[59] Wall, S.A. and Brewster, S.: Sensory substitution using tactile pin arrays: Human factors, technology and applications. *Signal Processing*, vol. 86, no. 12, pp. 3674 – 3695, 2006. ISSN 0165-1684.

[60] Rov, A.M., Neelakantan, V.A. and Washington, G.N.: Design and development of passive and active force feedback systems using magnetorheological fluids. *ASME Conference Proceedings*, vol. 2003, no. 37076, pp. 333–339, 2003.

[61] Rosenberg, L. and Brave, S.: Using force feedback to enhance human performance in graphical user interfaces. In: *Conference companion on Human factors in computing systems: common ground*, CHI '96, pp. 291–292. ACM, New York, NY, USA, 1996. ISBN 0-89791-832-0.

[62] Lawrence, D. and Chapel, J.: Performance trade-offs for hand controller design. In: *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, vol. 4, pp. 3211 –3216. may 1994.

[63] Grange, S., Conti, F., Rouiller, P., Helmer, P. and Baur, D.C.: Overview of the delta haptic device. In: *In Eurohaptics'01*. Harper Collins Publishers, 2001.

[64] Inc, Q.C.: 5 dof haptic wand system. [Online], 2011.
Available at: http://www.quanser.com/net/industrial/Systems_and_Products/Sys_5Dof_Haptic.aspx

[65] Kim, Y., Jangwook Lee, J., Lee, S. and Kim, M.: A force reflected exoskeleton-type master arm for human-robot interaction. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, no. 2, pp. 198–212, 2005.

[66] Delp, S., Loan, P. and Basdogan, C.: Surgical simulation: An emerging technology for training in emergency medicine. *Presence - Virtual Environments for Medicine*, vol. 6, no. 2, pp. 147–159, 1997.

[67] Sulaiman, S., Blandford, A. and Cairns, P.: Haptic experience and the design of drawing interfaces. *Interacting with Computers*, vol. 22, no. 3, pp. 193 – 205, 2010. ISSN 0953-5438.

[68] Smyth, T.N. and Kirkpatrick, A.E.: A new approach to haptic augmentation of the gui. In: *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI '06, pp. 372–379. ACM, New York, NY, USA, 2006. ISBN 1-59593-541-X.

[69] Grange, S., Conti, F., Helmer, P., Rouiller, P. and Baur, C.: The delta haptic device as a nanomanipulator. In: *SPIE Microrobotics and Microassembly III*, pp. 99–111. 2001.

[70] Dimension, F.: delta.6 haptic device (data sheet). [Online], 2011.
Available at: http://www.forcedimension.com/downloads/specs/specsheet-delta.6.pdf

[71] Motoman: Motoman up6 specifications (data sheet). [Online], 2011.
Available at: www.micromech.co.uk/dir_products/pdf/motoman/up6-series_robot.pdf

[72] Motoman: Motoman sda10 specifications (data sheet). [Online], 2011.
Available at: www.motoman.com/datasheets/SDA10D.pdf

[73] Motoman: Motoman xrc specifications (data sheet). [Online], 2011.
Available at: www.motoman.com/datasheets/XRC%20Controller.pdf

[74] Motoman: Motoman dx100 specifications (data sheet). [Online], 2011.
Available at: www.motoman.com/datasheets/DX100%20Controller.pdf

[75] Barionet: Barionet 100 / barionet 50 user manual. [Online], 2011.
Available at: http://www.barix.com/downloads/Barionet_Family/51/

[76] Barionet: Barix control language(bcl) programmers manual. [Online], 2011.
Available at: http://www.barix.com/downloads/Barionet_Family/51/

[77] Vivotek: Network camera user's manual. [Online], 2011.
Available at: http://www.vivotek.com/products/model.php?network_camera=pz7151

[78] Ullman, D.: *The Mechanical Design Process*. McGraw-Hill, 2003. ISBN 0072373385, 9780072373387.

[79] M., A. and Badawy: Technology management simply defined: A tweet plus two characters. *Journal of Engineering and Technology Management*, vol. 26, no. 4, pp. 219 – 224, 2009. ISSN 0923-4748.

[80] Cetindamar, D., Phaal, R. and Probert, D.: Understanding technology management as a dynamic capability: A framework for technology management activities. *Technovation*, vol. 29, no. 4, pp. 237 – 246, 2009. ISSN 0166-4972.

[81] Lee, J.: Apply force/torque sensors to robotic applications. *Robotics*, vol. 3, no. 2, pp. 189 – 194, 1987. ISSN 0167-8493. Special Issue: Sensors.

[82] ATI Industrial Automation, I.: Ati industrial automation: Robotic end effectors and automation tooling. [Online], 2011.
Available at: `http://www.ati-ia.com`

[83] ATI Industrial Automation, I.: Ati industrial automation: Net f/t. [Online], 2011.
Available at: `http://www.ati-ia.com/products/ft/ft_NetFT.aspx`

[84] Gopal, M.: *Digital Control and State Variable Methods: Conventional and Neuro-Fuzzy Controlsystems*. McGraw-Hill Education, 2004. ISBN 0070483027.

[85] Gopal, M.: *Control Systems: Principles and Design*. McGraw-Hill Education, 2002. ISBN 0070482896.

[86] Lloyd, J.E. and Hayward, V.: Kinematics of common industrial robots. *Robotics and Autonomous Systems*, vol. 4, pp. 169–191, 1988.

[87] Meriam, J.L. and Kraige, L.G.: *Engineering Mechanics: Dynamics*. John Wiley and Sons, Incorporated, 2003. ISBN 047126606X, 9780471266068.

[88] Young, K. and Pickin, C.: Accuracy assessment of the modern industrial robot. *Industrial Robot*, vol. 27, no. 06, pp. 427–436, 2000.

[89] Greenway, B.: Robot accuracy. *Industrial Robot*, vol. 27, no. 04, pp. 257–265, 2000.

[90] Figliola, R.S. and Beasly, D.E.: *Theory and Design for Mechanical Measurements, 4th Edition*. John Wiley and Sons, Incorporated, 2006. ISBN 0471445932, 9780471445937.

[91] EN ISO 9283: Manipulating industrial robots - performance criteria and related test methods. 1998.

Appendices

Appendix A

Motion Derivation

A.1 Equation of Linear Motion

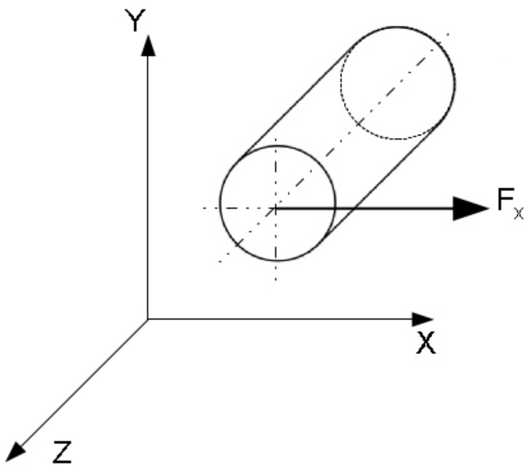
An external force, F_x, acting on a rod of mass, m, length, L and radius, r, is depicted in the free body diagram in Figure A.1a. Utilizing Newton's second law (Equation A.1.1), the dynamic equation can be derived to establish the relationship between force, distance and linear velocity. where $\ddot{x}$ refers to the acceleration of the rod in the x-direction.

$$F_x = m\ddot{x} \tag{A.1.1}$$

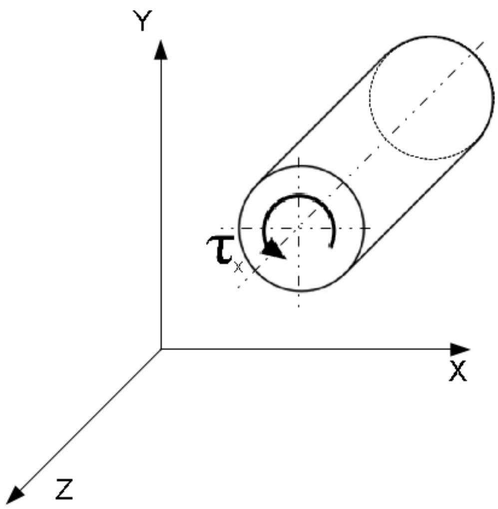
Rearranging Equation A.1.1 to express acceleration in terms of force we obtain Equation A.1.2.

$$\ddot{x} = \frac{F_x}{m} \tag{A.1.2}$$

Integrating the acceleration over time we obtain Equation A.1.3, the linear velocity of the rod at time t.



(a) External Force



(b) External Torque

Figure A.1: Force Torque Free Body Diagram

$$\int \ddot{x} = \dot{x} = \frac{F_x}{m}t + \dot{x_0} \tag{A.1.3}$$

Where $\dot{x_0}$ is the initial velocity of the rod. Integrating the linear velocity over time we obtain Equation A.1.4, the position of the rod at time t.

$$\int \dot{x} = x = \frac{F_x}{2m}t^2 + \dot{x_0}t + x_0 \tag{A.1.4}$$

Where x_0 is the initial position of the rod. The same process can be followed for a vector force consisting of three force components, F_x, F_y and F_z to obtain the resulting position (Equation A.1.5) and velocity (Equation A.1.6).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{F_x}{2m} \\ \frac{F_y}{2m} \\ \frac{F_z}{2m} \end{bmatrix} t^2 + \begin{bmatrix} \dot{x_0} \\ \dot{y_0} \\ \dot{z_0} \end{bmatrix} t + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{A.1.5}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{F_x}{m} \\ \frac{F_y}{m} \\ \frac{F_z}{m} \end{bmatrix} t + \begin{bmatrix} \dot{x_0} \\ \dot{y_0} \\ \dot{z_0} \end{bmatrix} \tag{A.1.6}$$

A.2 Equation of Angular Motion

Consider an external torque, τ_x, acting on the a rod of mass, m, length, L and radius, r, as depicted in the free body diagram in Figure A.1b. Making use of the fundamental torque (Equation A.2.1), which is a variation on Newton's second law when applied to moments, the dynamic equation can be derived to establish the relationship between Torque, orientation and angular velocity.

$$\tau_x = I_{xx}\ddot{\theta_x} \tag{A.2.1}$$

Where $\ddot{\theta_x}$ and I_{xx} refers to the angular acceleration of the rod and mass moment of inertia around the object's x-axis respectively. Rearranging Equation A.2.1 to express angular acceleration in terms of torque we obtain Equation A.2.2.

$$\ddot{\theta_x} = \frac{\tau_x}{I_{xx}} \tag{A.2.2}$$

Integrating the acceleration over time we obtain Equation A.2.3, the angular velocity of the rod at time t.

$$\int \ddot{\theta_x} = \dot{\theta_x} = \frac{\tau_x}{I_{xx}}t + \dot{\theta_{x0}} \tag{A.2.3}$$

Where $\dot{\theta_{x_0}}$ is the initial angular velocity of the rod. Integrating the angular velocity over time we obtain Equation A.2.4, the orientation of the rod at time t.

$$\int \dot{\theta_x} = \theta_x = \frac{\tau_x}{2I_{xx}}t^2 + \dot{\theta_{x0}}t + \theta_{x0} \tag{A.2.4}$$

Where θ_{x_0} is the initial position of the rod. The same process can be followed for a vector torque consisting of three torque components, τ_x, τ_y and τ_z to obtain the resulting orientation (Equation A.2.5) and angular velocity (Equation A.2.6).

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{2I_{xx}} \\ \frac{\tau_y}{2I_{yy}} \\ \frac{\tau_z}{2I_{zz}} \end{bmatrix} t^2 + \begin{bmatrix} \dot{\theta}_{x0} \\ \dot{\theta}_{y0} \\ \dot{\theta}_{z0} \end{bmatrix} t + \begin{bmatrix} \theta_{x0} \\ \theta_{y0} \\ \theta_{z0} \end{bmatrix} \tag{A.2.5}$$

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \frac{\tau_x}{I_{xx}} \\ \frac{\tau_y}{I_{yy}} \\ \frac{\tau_z}{I_{zz}} \end{bmatrix} t + \begin{bmatrix} \dot{\theta}_{x0} \\ \dot{\theta}_{y0} \\ \dot{\theta}_{z0} \end{bmatrix} \tag{A.2.6}$$

Where the mass moment of inertia, I_{xx}, I_{yy}, I_{zz} for the tip of the rod is defined in Equation A.2.7 [87].

$$\begin{bmatrix} I_{xx} \\ I_{yy} \\ I_{zz} \end{bmatrix} = \begin{bmatrix} \frac{mr^2}{4} \\ \frac{mr^2}{4} \\ \frac{mr^2}{2} \end{bmatrix} + \begin{bmatrix} \frac{ml^2}{12} \\ \frac{ml^2}{12} \\ 0 \end{bmatrix} \tag{A.2.7}$$

Appendix B

Experimental Data for the Pose Evaluation

Pose P1			
n	**x**	**y**	**z**
1	728.198	58.265	-310.948
2	728.192	58.370	-310.900
3	728.032	58.421	-310.920
4	727.719	58.427	-310.725
5	727.900	58.229	-310.753
6	728.179	58.206	-310.791
7	728.286	58.651	-310.369
8	727.720	58.578	-310.718
9	728.136	58.299	-311.061
10	728.279	58.240	-310.857
11	727.947	58.378	-310.923
12	728.203	58.555	-310.895
13	728.035	58.143	-310.601
14	728.012	58.446	-310.835
15	728.017	58.356	-310.824

Table B.1: Pose 1 Data

Pose P2			
n	**x**	**y**	**z**
1	636.830	-32.064	-239.901
2	636.585	-32.016	-239.972
3	636.698	-32.077	-239.872
4	637.006	-32.300	-240.059
5	636.440	-32.053	-240.099
6	636.708	-32.270	-240.155
7	636.575	-32.346	-239.896
8	636.918	-32.177	-240.089
9	637.044	-32.174	-240.080
10	636.831	-32.155	-240.030
11	637.023	-32.101	-240.087
12	636.610	-32.167	-239.718
13	636.646	-31.831	-239.662
14	636.389	-31.782	-239.806
15	636.729	-31.825	-240.099

Table B.2: Pose 2 Data

Pose P3			
n	**x**	**y**	**z**
1	638.380	150.235	-242.579
2	638.555	150.166	-242.851
3	638.280	150.279	-242.422
4	638.206	150.160	-242.564
5	638.714	150.210	-242.799
6	638.483	150.716	-242.777
7	638.585	150.161	-242.495
8	638.523	150.095	-242.647
9	638.367	150.130	-242.609
10	638.577	150.007	-242.642
11	638.287	150.254	-242.542
12	638.393	150.280	-242.676
13	638.507	150.575	-242.676
14	638.559	150.187	-242.890
15	638.226	150.383	-242.433

Table B.3: Pose 3 Data

Pose P4			
n	**x**	**y**	**z**
1	831.648	147.843	-390.102
2	831.714	147.860	-390.300
3	831.634	147.919	-390.166
4	831.561	148.041	-390.347
5	831.676	147.844	-390.434
6	831.533	147.912	-390.319
7	831.705	147.589	-390.156
8	831.399	147.893	-390.192
9	831.968	147.926	-390.164
10	831.640	147.800	-390.401
11	831.918	147.915	-390.441
12	831.524	148.115	-390.260
13	831.466	147.898	-390.099
14	831.568	147.984	-390.281
15	831.629	147.995	-390.434

Table B.4: Pose 4 Data

Table B.5: Pose 5 Data

Pose P5			
n	**x**	**y**	**z**
1	830.138	-34.772	-388.527
2	830.032	-34.057	-388.827
3	830.430	-34.448	-389.014
4	830.119	-34.700	-389.025
5	829.765	-34.382	-388.726
6	830.117	-34.290	-389.018
7	830.070	-34.224	-388.794
8	830.048	-34.308	-388.733
9	830.057	-34.550	-388.976
10	830.075	-34.370	-388.823
11	830.212	-34.639	-388.909
12	830.087	-34.247	-388.668
13	830.206	-34.820	-388.891
14	830.011	-34.555	-388.778
15	830.108	-34.531	-388.841

Appendix C

Experimental Data for the Steady State Force Evaluation

Table C.1: Steady State Force Error Summary

Steady State [N]			
n	**Fx**	**Fy**	**Fz**
1	0.003	-0.097	-0.057
2	0.015	-0.113	-0.050
3	0.006	-0.099	-0.042
4	-0.006	-0.097	-0.065
5	0.011	-0.106	-0.065
6	0.019	-0.097	-0.057
7	0.011	-0.116	-0.057
8	-0.003	-0.090	-0.072
9	0.010	-0.113	-0.064
10	0.010	-0.131	-0.065
11	0.001	-0.122	-0.049
12	-0.008	-0.099	-0.078
13	0.018	-0.103	-0.064
14	0.014	-0.096	-0.063
15	0.014	-0.096	-0.070

Table C.2: Steady State Force and Time Data Sample

-	Master				Slave			
m	**Fx [N]**	**Fy [N]**	**Fz [N]**	**T [s]**	**Fx [N]**	**Fy [N]**	**Fz [N]**	**T [s]**
...	...	...	...	...	...	...	...	...
60	5.657	7.042	16.974	6.001	5.652	5.742	16.816	5.984
61	5.662	7.044	16.959	6.110	5.664	5.836	16.821	6.094
62	5.662	7.044	16.959	6.204	5.668	5.943	16.825	6.187
63	5.662	7.044	16.966	6.313	5.680	6.032	16.837	6.297
64	5.662	7.044	16.974	6.407	5.688	6.109	16.856	6.390
65	5.662	7.044	16.981	6.516	5.692	6.184	16.866	6.500
66	5.666	7.042	16.981	6.610	5.691	6.251	16.877	6.594
67	5.666	7.047	16.981	6.719	5.691	6.323	16.895	6.703
68	5.657	7.047	16.981	6.813	5.690	6.395	16.891	6.797
69	5.657	7.047	16.981	6.923	5.694	6.450	16.908	6.906
70	5.657	7.042	16.981	7.016	5.694	6.498	16.895	7.000
71	5.657	7.047	16.981	7.126	5.697	6.539	16.897	7.109
72	5.657	7.042	16.981	7.219	5.701	6.584	16.899	7.203
73	5.657	7.047	16.966	7.329	5.697	6.625	16.893	7.312
74	5.657	7.047	16.959	7.423	5.696	6.702	16.904	7.406
75	5.662	7.044	16.974	7.532	5.691	6.727	16.907	7.516
76	5.662	7.044	16.981	7.626	5.691	6.790	16.909	7.609
77	5.657	7.042	16.981	7.735	5.691	6.838	16.918	7.719
78	5.657	7.047	16.981	7.844	5.690	6.852	16.920	7.812
79	5.657	7.047	16.981	7.938	5.682	6.885	16.922	7.922
80	5.657	7.047	16.981	8.048	5.677	6.917	16.938	8.016
81	5.662	7.044	16.959	8.141	5.677	6.921	16.931	8.125
82	5.662	7.044	16.966	8.251	5.672	6.938	16.947	8.219
83	5.662	7.044	16.974	8.344	5.668	6.945	16.932	8.328
84	5.662	7.044	16.974	8.454	5.672	6.943	16.931	8.422
85	5.662	7.044	16.981	8.548	5.672	6.943	16.916	8.531
86	5.662	7.039	16.981	8.657	5.672	6.943	16.908	8.625
87	5.666	7.047	16.981	8.751	5.672	6.938	16.908	8.734
88	5.662	7.044	16.981	8.860	5.668	6.936	16.901	8.828
89	5.662	7.044	16.981	8.954	5.668	6.936	16.893	8.937
90	5.657	7.047	16.981	9.063	5.664	6.933	16.894	9.031
91	5.657	7.047	16.981	9.157	5.668	6.931	16.901	9.141
92	5.657	7.042	16.981	9.266	5.664	6.928	16.893	9.234
93	5.657	7.042	16.974	9.360	5.664	6.928	16.893	9.344
94	5.653	7.044	16.966	9.470	5.659	6.926	16.893	9.438
95	5.662	7.044	16.966	9.563	5.660	6.926	16.900	9.547
96	5.662	7.044	16.966	9.673	5.664	6.924	16.892	9.641
97	5.662	7.044	16.966	9.766	5.664	6.924	16.892	9.750
98	5.662	7.044	16.974	9.876	5.659	6.921	16.877	9.844
99	5.662	7.044	16.981	9.970	5.660	6.916	16.869	9.953
...	...	...	...	...	...	...	...	...

Appendix D

Experimental Data for the Dynamic Force Evaluation

Table D.1: Dynamic Force Vector Error Summary

n	Mean Error [N]
1	1.53
2	1.68
3	1.72
4	1.63
5	1.66
6	1.68
7	1.69
8	1.68
9	1.65
10	1.52
11	1.53
12	1.65
13	1.76
14	1.64
15	1.57

Table D.2: Dynamic Force and Time Data Sample

-	Master				Slave			
m	**Fx [N]**	**Fy [N]**	**Fz [N]**	**T [s]**	**Fx [N]**	**Fy [N]**	**Fz [N]**	**T [s]**
...	...	...	...	...	...	...	...	...
100	-2.328	7.199	13.583	10.078	-1.509	7.261	14.781	10.063
101	-2.395	7.203	13.492	10.187	-1.548	7.229	14.629	10.156
102	-2.462	7.202	13.410	10.281	-1.574	7.204	14.462	10.266
103	-2.529	7.196	13.312	10.391	-1.613	7.172	14.303	10.359
104	-2.592	7.188	13.207	10.500	-1.648	7.142	14.166	10.469
105	-2.655	7.169	13.110	10.594	-1.683	7.127	14.045	10.563
106	-2.714	7.153	13.013	10.703	-1.722	7.114	13.940	10.672
107	-2.773	7.128	12.909	10.797	-1.765	7.098	13.827	10.766
108	-2.828	7.104	12.819	10.906	-1.817	7.087	13.714	10.875
109	-2.895	7.060	12.685	11.031	-1.887	7.075	13.595	11.016
110	-2.942	7.017	12.581	11.125	-1.943	7.066	13.491	11.109
111	-2.985	6.976	12.484	11.234	-1.987	7.060	13.401	11.219
112	-3.031	6.928	12.365	11.328	-2.056	7.062	13.328	11.313
113	-3.070	6.884	12.261	11.437	-2.130	7.063	13.247	11.422
114	-3.104	6.828	12.150	11.531	-2.186	7.053	13.174	11.516
115	-3.134	6.770	12.038	11.641	-2.255	7.056	13.101	11.625
116	-3.168	6.709	11.934	11.734	-2.316	7.054	13.049	11.719
117	-3.194	6.653	11.823	11.844	-2.385	7.052	12.977	11.828
118	-3.220	6.582	11.705	11.953	-2.450	7.047	12.911	11.922
119	-3.238	6.516	11.579	12.047	-2.523	7.042	12.838	12.031
120	-3.252	6.442	11.460	12.156	-2.593	7.036	12.771	12.125
121	-3.274	6.364	11.357	12.250	-2.653	7.019	12.689	12.234
122	-3.279	6.275	11.247	12.359	-2.709	7.001	12.614	12.328
123	-3.280	6.194	11.136	12.453	-2.770	6.979	12.525	12.438
124	-3.281	6.107	11.025	12.562	-2.839	6.963	12.443	12.531
125	-3.274	6.020	10.914	12.672	-2.891	6.942	12.353	12.641
126	-3.271	5.931	10.796	12.766	-2.956	6.919	12.271	12.734
127	-3.264	5.845	10.685	12.875	-2.999	6.884	12.165	12.844
128	-3.244	5.750	10.575	12.969	-3.055	6.822	12.036	12.938
129	-3.229	5.653	10.472	13.078	-3.098	6.769	11.936	13.047
130	-3.189	5.546	10.362	13.203	-3.137	6.722	11.829	13.141
131	-3.174	5.454	10.251	13.297	-3.171	6.663	11.722	13.250
132	-3.150	5.352	10.148	13.406	-3.201	6.597	11.613	13.344
133	-3.118	5.249	10.045	13.500	-3.231	6.523	11.481	13.453
134	-3.082	5.149	9.950	13.609	-3.253	6.448	11.350	13.547
135	-3.030	5.005	9.804	13.750	-3.270	6.376	11.225	13.656
136	-2.985	4.899	9.700	13.859	-3.291	6.296	11.086	13.750
137	-2.936	4.786	9.612	13.953	-3.295	6.213	10.960	13.859
138	-2.883	4.690	9.532	14.062	-3.290	6.110	10.826	13.953
139	-2.814	4.530	9.400	14.219	-3.294	6.012	10.685	14.063
...	...	...	...	...	...	...	...	...

Appendix E

Force Component Results

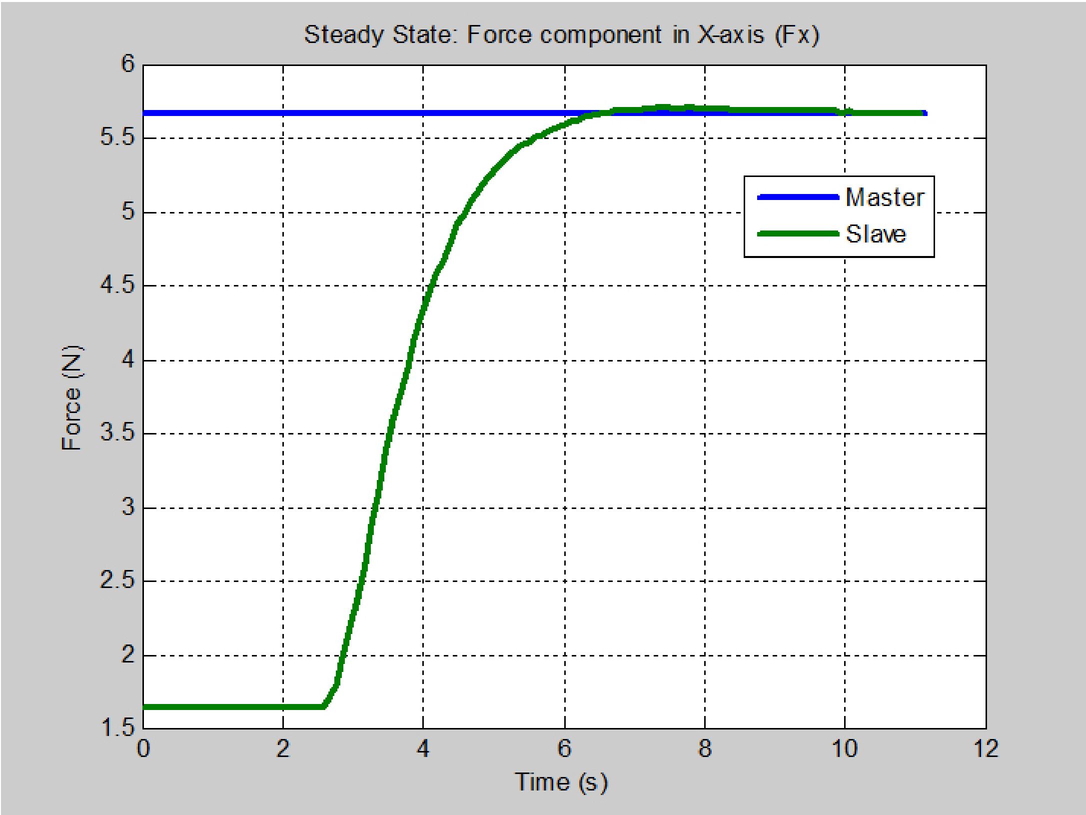


Figure E.1: Steady State Fx Results Graph

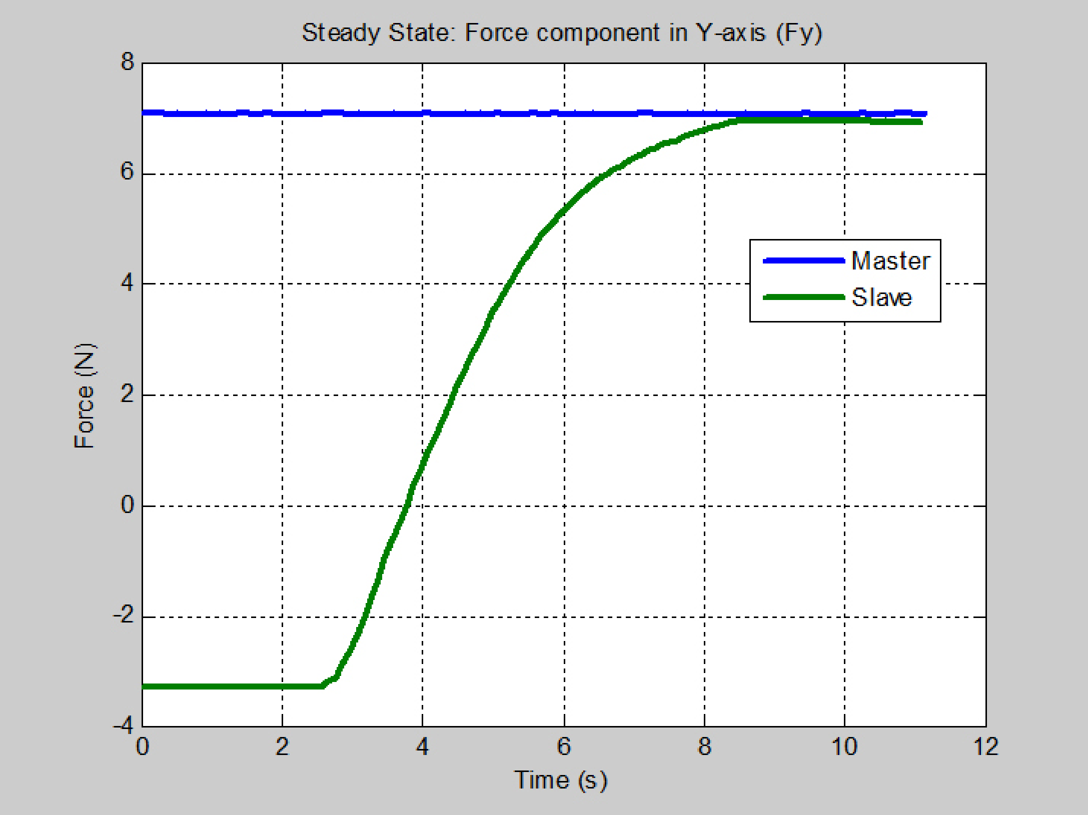


Figure E.2: Steady State Fy Results Graph

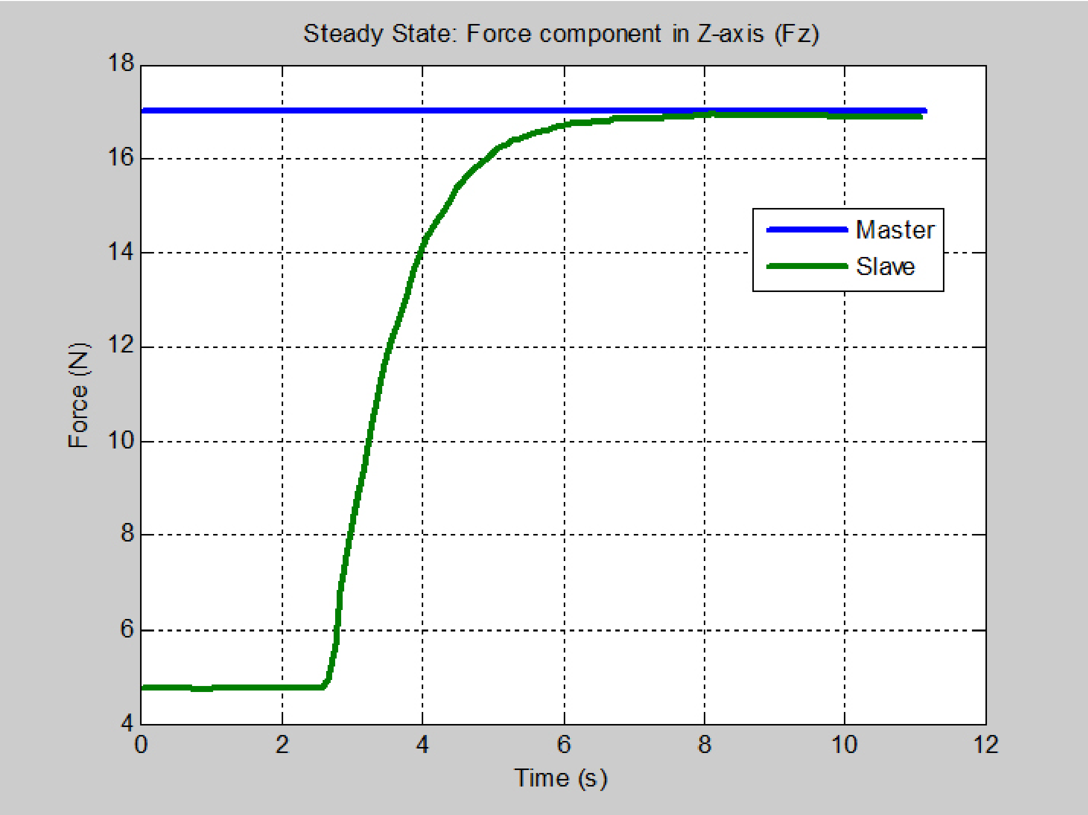


Figure E.3: Steady State Fz Results Graph

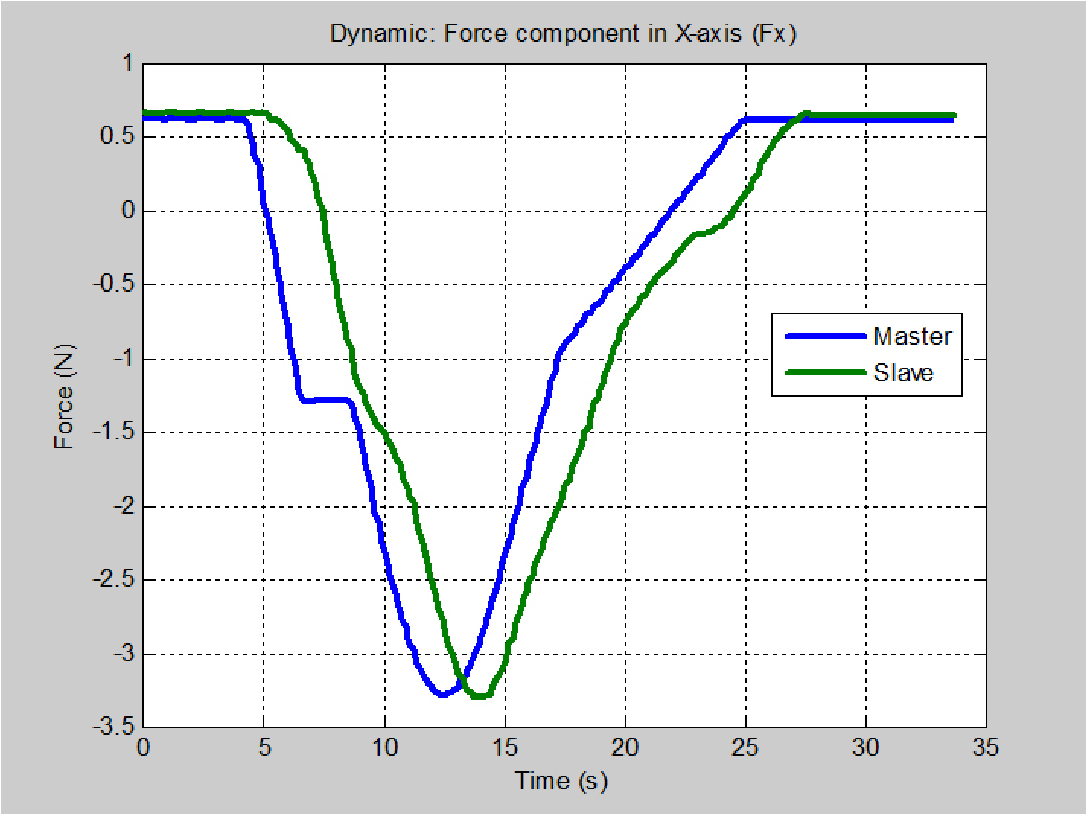


Figure E.4: Dynamic Fx Results Graph

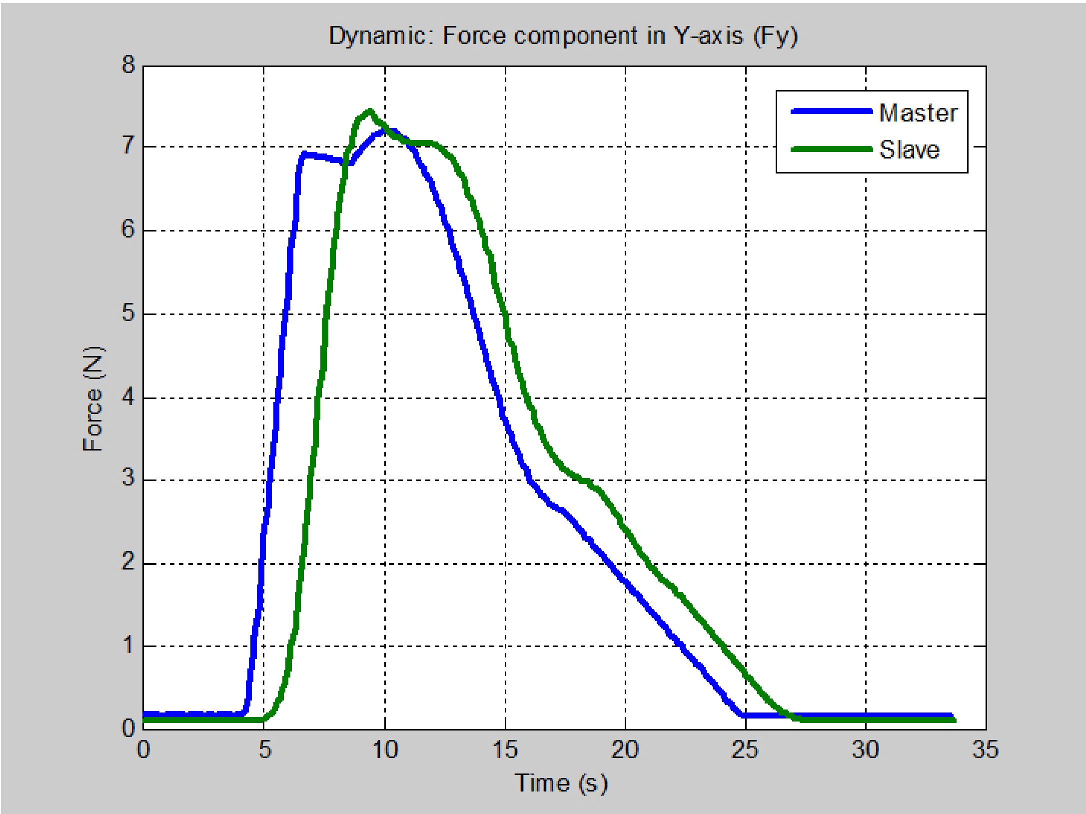


Figure E.5: Dynamic Fy Results Graph

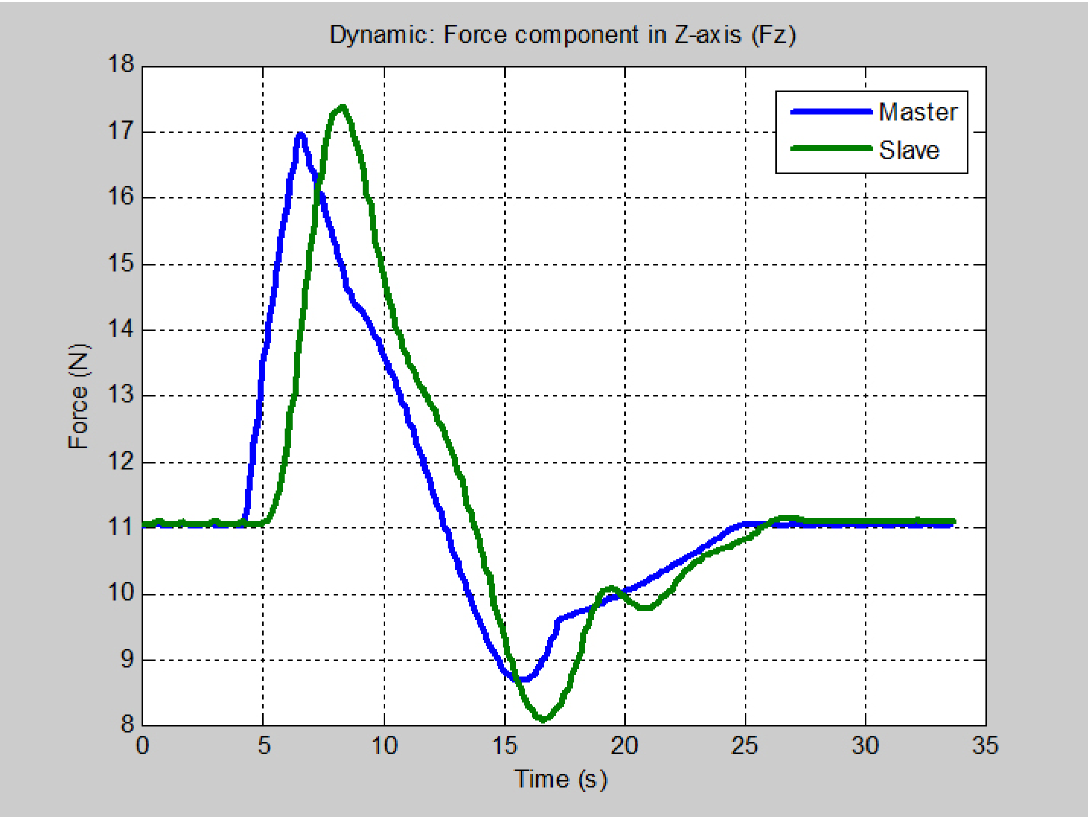


Figure E.6: Dynamic Fz Results Graph

Appendix F

Technical Drawings

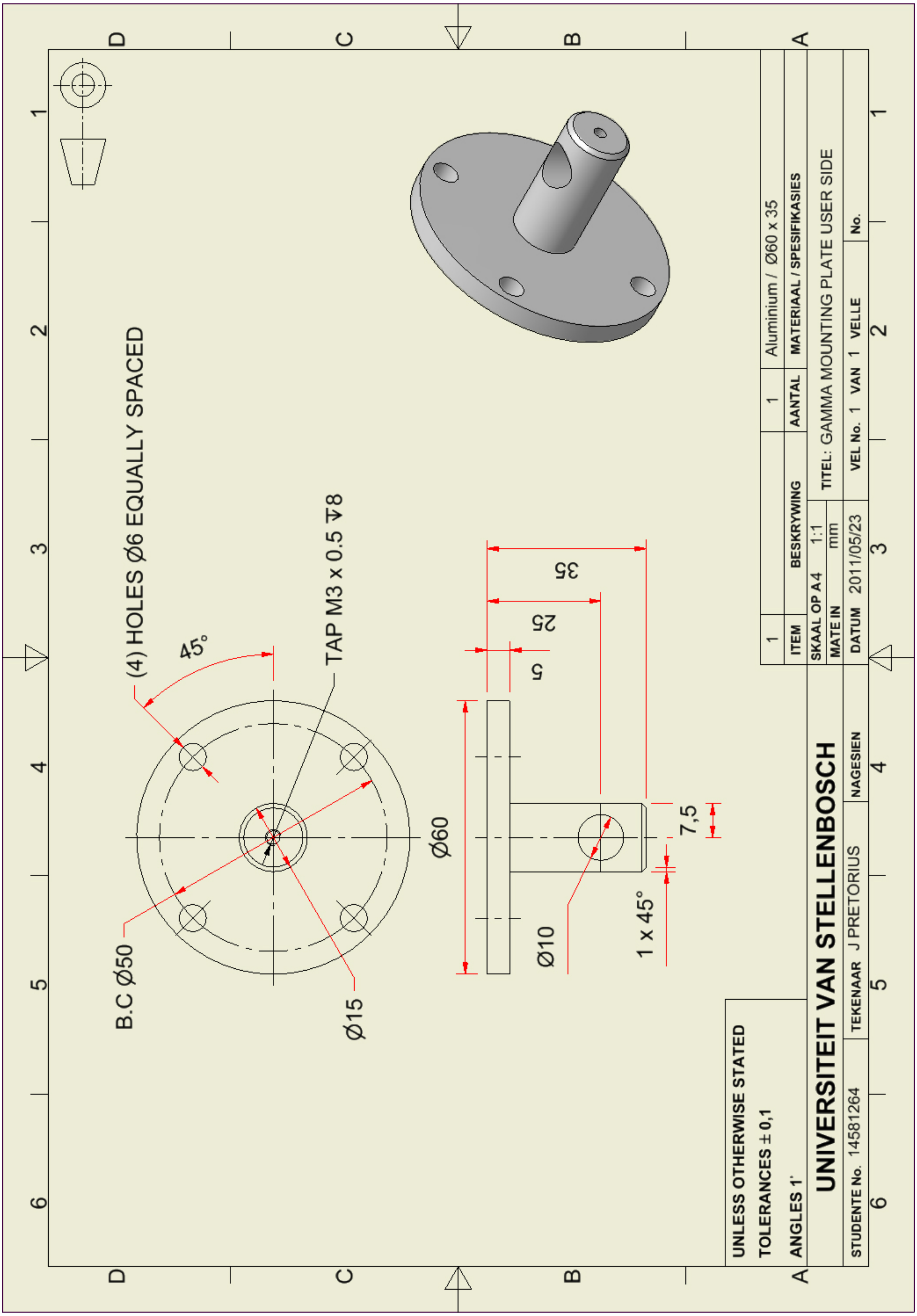


Figure F.1: Pencil Mounting Plate

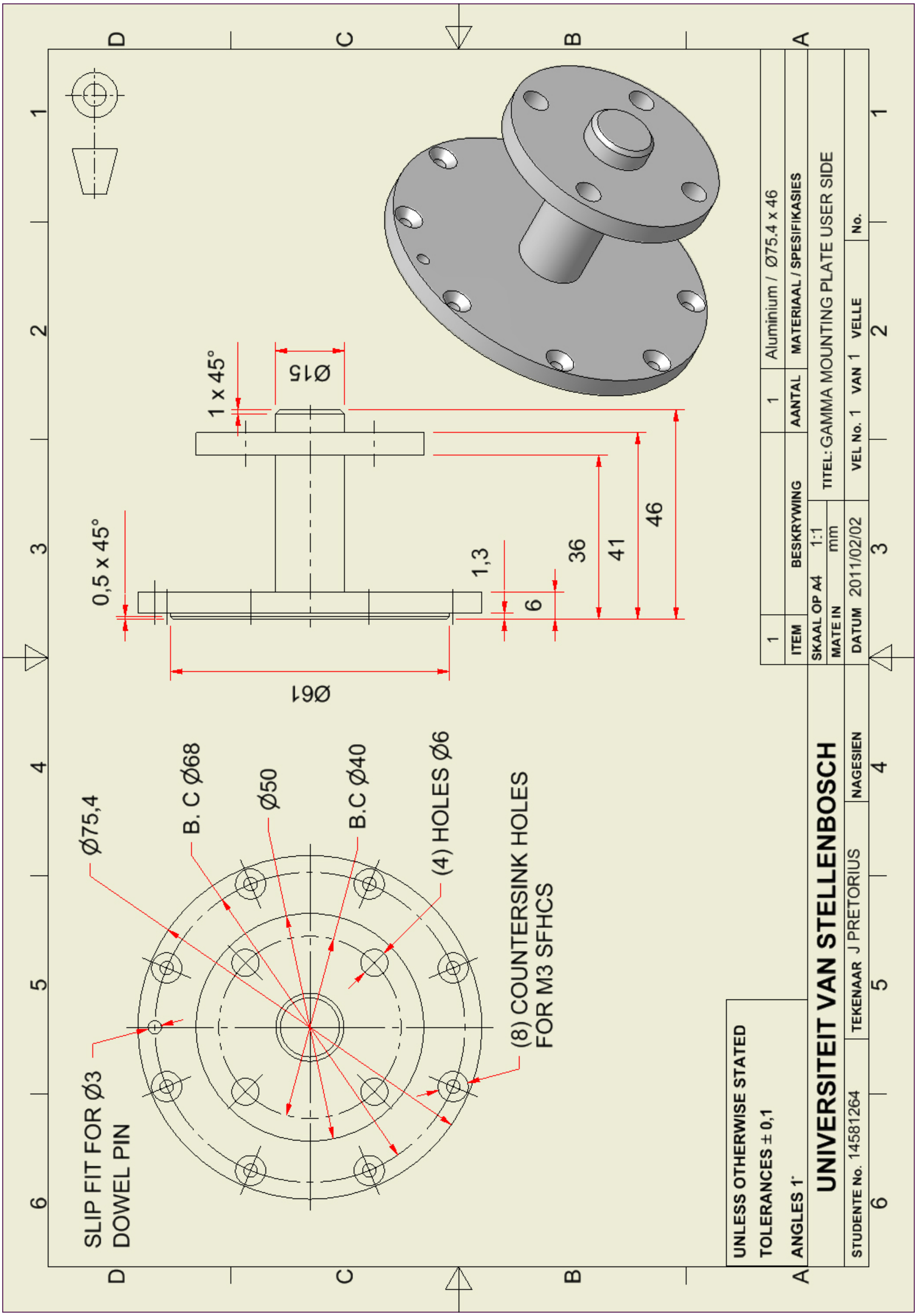


Figure F.2: UP6 Mounting Plate

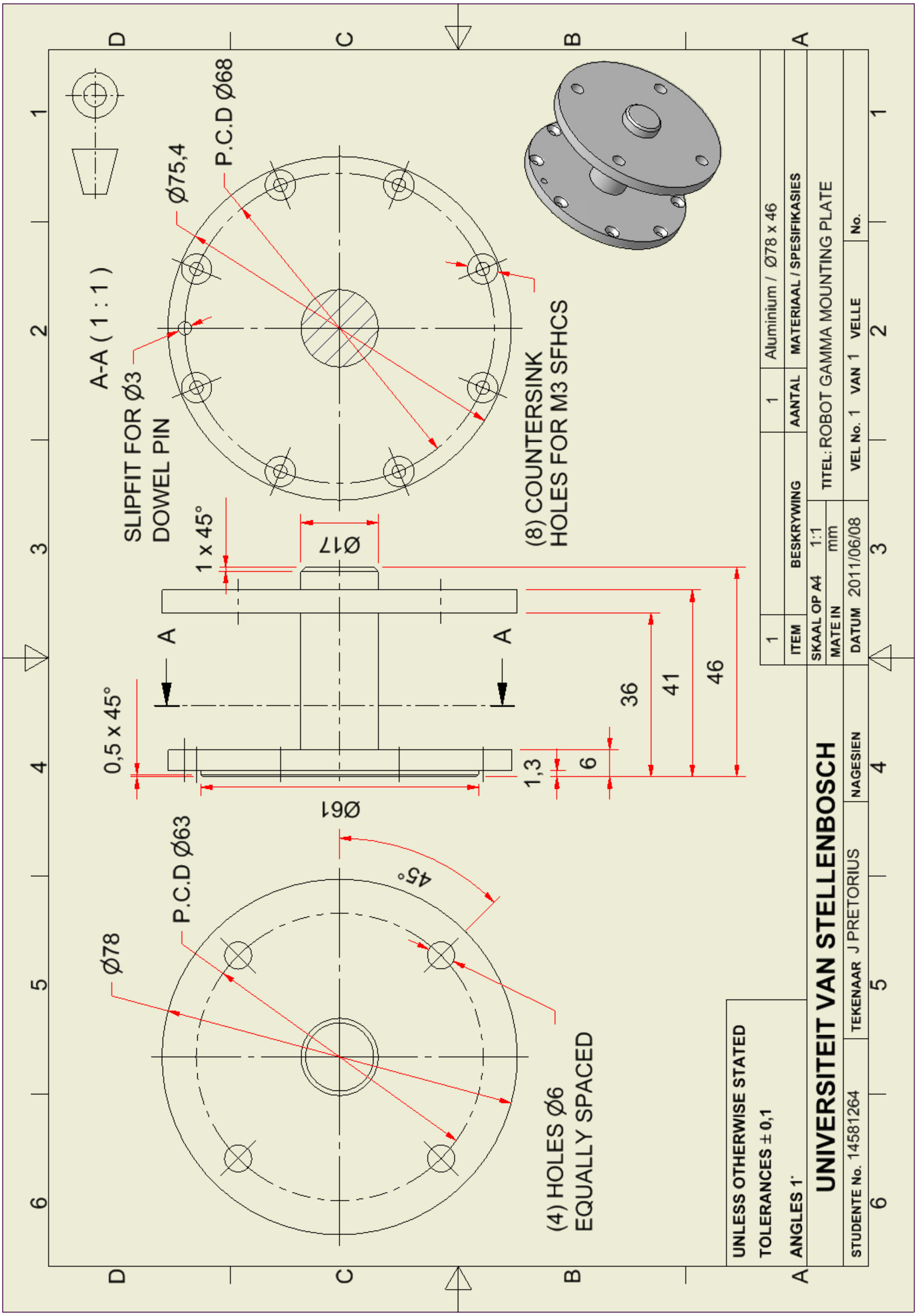


Figure F.3: SDA10D Mounting Plate

Appendix G

t-Distribution Probability Table

THE *t* DISTRIBUTION

Table entries are values of t corresponding to proportions in one tail or in two tails combined.

	PROPORTION IN ONE TAIL					
	0.25	0.10	0.05	0.025	0.01	0.005
	PROPORTION IN TWO TAILS COMBINED					
df	0.50	0.20	0.10	0.05	0.02	0.01
1	1.000	3.078	6.314	12.706	31.821	63.657
2	0.816	1.886	2.920	4.303	6.965	9.925
3	0.765	1.638	2.353	3.182	4.541	5.841
4	0.741	1.533	2.132	2.776	3.747	4.604
5	0.727	1.476	2.015	2.571	3.365	4.032
6	0.718	1.440	1.943	2.447	3.143	3.707
7	0.711	1.415	1.895	2.365	2.998	3.499
8	0.706	1.397	1.860	2.306	2.896	3.355
9	0.703	1.383	1.833	2.262	2.821	3.250
10	0.700	1.372	1.812	2.228	2.764	3.169
11	0.697	1.363	1.796	2.201	2.718	3.106
12	0.695	1.356	1.782	2.179	2.681	3.055
13	0.694	1.350	1.771	2.160	2.650	3.012
14	0.692	1.345	1.761	2.145	2.624	2.977
15	0.691	1.341	1.753	2.131	2.602	2.947
16	0.690	1.337	1.746	2.120	2.583	2.921
17	0.689	1.333	1.740	2.110	2.567	2.898
18	0.688	1.330	1.734	2.101	2.552	2.878
19	0.688	1.328	1.729	2.093	2.539	2.861
20	0.687	1.325	1.725	2.086	2.528	2.845
21	0.686	1.323	1.721	2.080	2.518	2.831
22	0.686	1.321	1.717	2.074	2.508	2.819
23	0.685	1.319	1.714	2.069	2.500	2.807
24	0.685	1.318	1.711	2.064	2.492	2.797
25	0.684	1.316	1.708	2.060	2.485	2.787
26	0.684	1.315	1.706	2.056	2.479	2.779
27	0.684	1.314	1.703	2.052	2.473	2.771
28	0.683	1.313	1.701	2.048	2.467	2.763
29	0.683	1.311	1.699	2.045	2.462	2.756
30	0.683	1.310	1.697	2.042	2.457	2.750
40	0.681	1.303	1.684	2.021	2.423	2.704
60	0.679	1.296	1.671	2.000	2.390	2.660
120	0.677	1.289	1.658	1.980	2.358	2.617
∞	0.674	1.282	1.645	1.960	2.326	2.576

Figure G.1: t-Distribution

Appendix H

Motoman IMOV Command

<Example>

Command MOVL 0, 500.0, 2, 123.1, 50.34, 10.8, 180.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Response 0000

- **IMOV**

Moves a manipulator from the current position for a specified coordinate incremental value in linear motion.

Command format : IMOV Data-1, Data-2, ····, Data-18
Data-1 = Motion speed selection (0 : V (speed), 1 : VR (posture speed))
Data-2 = Motion speed (0.1 to □□□.□□ mm/s, 0.1 to □□□.□° /s)
Data-3 = Coordinate specification
0 : Base coordinate
1 : Robot coordinate
2 : User coordinate 1
:
:
65 : User coordinate 64
66 : Tool coordinate
* The order varies depending on the number of robot's axes.

	6-axis robot	**7-axis robot**
Data-4	X coordinate incremental value (unit : mm, significant 3 decimal points)	X coordinate incremental value (unit : mm, significant 3 decimal points)
Data-5	Y coordinate incremental value (unit : mm, significant 3 decimal points)	Y coordinate incremental value (unit : mm, significant 3 decimal points)
Data-6	Z coordinate incremental value (unit : mm, significant 3 decimal points)	Z coordinate incremental value (unit : mm, significant 3 decimal points)
Data-7	Wrist angle Rx incremental value (unit : degree (°), significant 4 decimal points)	Wrist angle Rx incremental value (unit : degree (°), significant 4 decimal points)
Data-8	Wrist angle Ry incremental value (unit : degree (°), significant 4 decimal points)	Wrist angle Ry incremental value (unit : degree (°), significant 4 decimal points)
Data-9	Wrist angle Rz incremental value (unit : degree (°), significant 4 decimal points)	Wrist angle Rz incremental value (unit : degree (°), significant 4 decimal points)
Data-10	Reserved	Elbow angle Re incremental value (unit : degree (°), significant 4 decimal points)
Data-11	Tool No. (0 to 63)	Reserved
Data-12	Number of 7th axis pulses (for travel axis, mm)	Tool No. (0 to 63)
Data-13	Number of 8th axis pulses (for travel axis, mm)	Number of 7th axis pulses (for travel axis, mm)
Data-14	Number of 9th axis pulses (for travel axis, mm)	Number of 8th axis pulses (for travel axis, mm)
Data-15	Number of 10th axis pulses	Number of 9th axis pulses (for travel axis, mm)
Data-16	Number of 11th axis pulses	Number of 10th axis pulses

	6-axis robot	7-axis robot
Data-17	Number of 12th axis pulses	Number of 11th axis pulses
Data-18	-	Number of 12th axis pulses

- In a system without external axis, Data-12 to Data-17 (for 7-axis robots, Data-13 to Data-18) should be set to “0”.
- If a specified user coordinate is not defined, an error occurs.

Response format : 0000 or Error code

<Example>

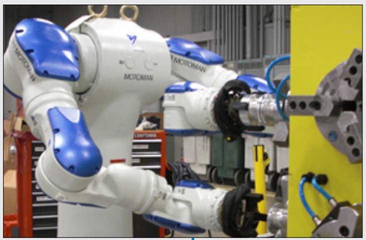
Command IMOV 0, 100.0, 2, 10.0, 10.0, 10.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Response 0000

Appendix I

Device Manuals

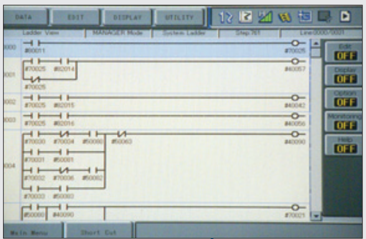
SOLUTIONS IN MOTION®

YASKAWA
MOTOMAN ROBOTICS



MACHINE TENDING

THRU-ARM CABLE AND HOSE ROUTING



LADDER EDITOR

TOP REASONS TO BUY

- Dexterity to perform complex tasks; dual 7-axis arms work together or independently
- Slim design optimizes space; provides "human-like" flexibility and range of motion, even in tight spaces
- Simplified tooling reduces cost
- Can be used in environments that are hazardous to humans
- Labor savings justifies capital investment

SDA10D

ASSEMBLY • PACKAGING • HANDLING • MACHINE TENDING • PART TRANSFER

Payload: 10 kg/arm

Slim, Dual-Arm Robot with "Human-Like" Flexibility

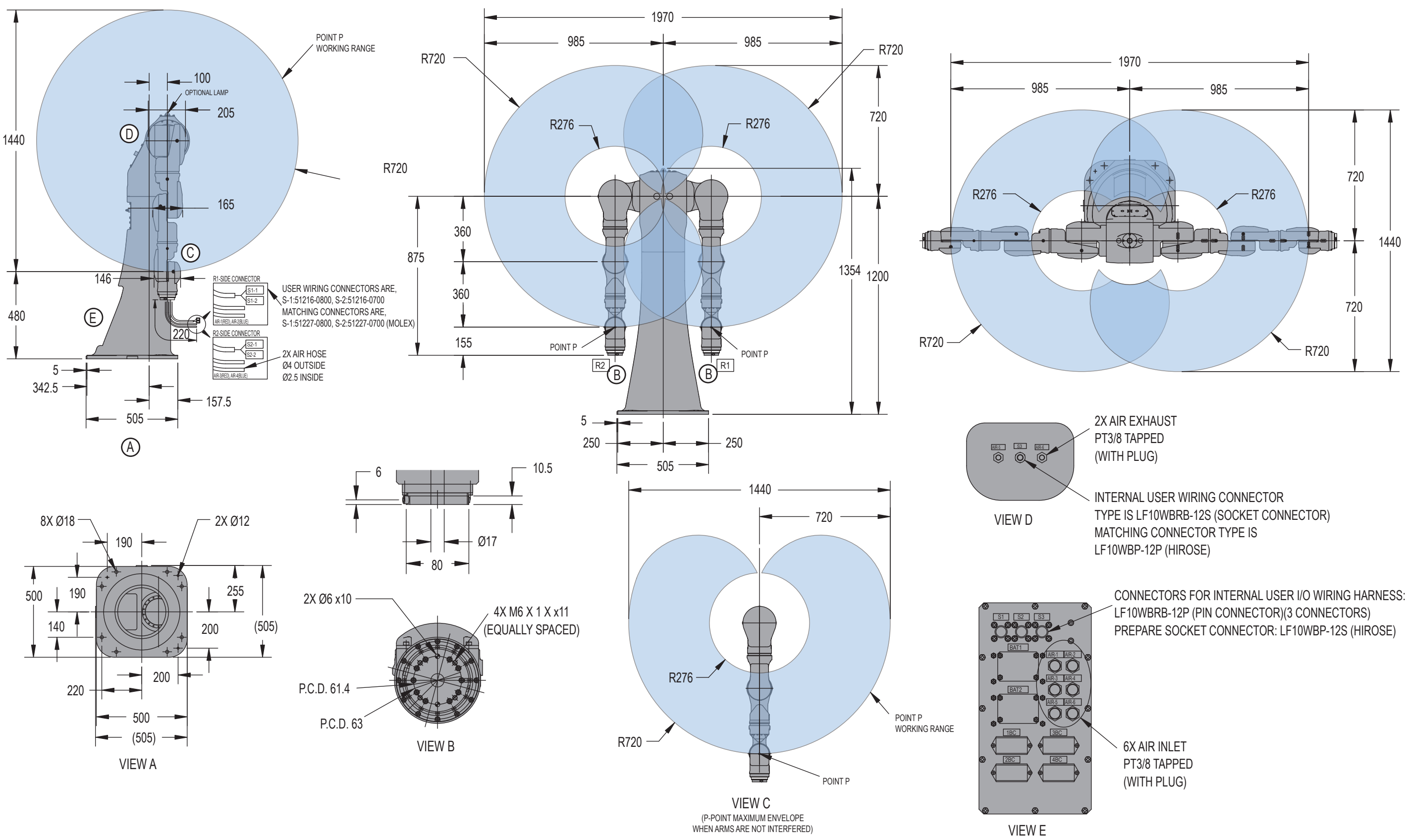
- Powerful actuator-based design provides "human-like" flexibility and fast acceleration.
- Superior dexterity and best-in-class wrist characteristics make slim, dual-arm robot ideally suited for assembly, part transfer, machine tending, packaging and other handling tasks that formerly could only be done by people.
- Highly flexible; 15 axes of motion (7 axes per arm, plus a single axis for base rotation).
- Internally routed cables and hoses (6 - air, 12 - electric) reduce interference and maintenance, and also make programming easier.
- 10 kg (22.1 lb) payload per arm; 720 mm (28.3") horizontal reach per arm; 1,440 mm (56.7") vertical reach per arm; ±0.1 mm (0.004") repeatability.
- Both robot arms can work together on one task to double the payload or handle heavy, unwieldy objects. Two manipulators can perform simultaneous independent operations.
- Ability to hold part with one arm while performing operations on it with other arm. Can transfer a part from one arm to the other with no need to set part down.

DX100 Controller

- Patented multiple robot control supports up to 8 robots/72 axes.
- Windows® CE programming pendant with color touch screen and USB interface.
- Faster processing speeds for smoother interpolation. Quicker I/O response. Accelerated Ethernet communication.
- Extensive I/O suite includes integral PLC and touch screen HMI, 2,048 I/O and graphical ladder editor.
- Supports all major fieldbus networks, including EtherNet/IP, DeviceNet, Profibus-DP and many others.
- Compliant to ANSI/RIA R15.06-1999 and other relevant ISO and CSA safety standards. Optional Category 3 functional safety unit.

SDA10D ROBOT

All dimensions are metric (mm) and for reference only. Please request detail drawings for all design/engineering requirements.



SDA10D SPECIFICATIONS

Structure		Articulated
Mounting		Floor
Controlled Axes		15 (7 axes per arm plus base rotation)
Payload		10 kg (22.1 lbs)/arm
Horizontal Reach per Arm		720 mm (28.3")
Horizontal Reach (P-point to P-point)		1,970 mm (77.6")
Vertical Reach		1,440 mm (56.7")
Repeatability		±0.1 mm (±0.004")
Maximum Motion Range	Rotation-Axis (Waist)	±170°
	S-Axis (Lifting)	±180°
	L-Axis (Lower Arm)	±110°
	E-Axis (Elbow)	±170°
	U-Axis (Upper Arm)	±135°
	R-Axis (Upper Arm Twist)	±180°
	B-Axis (Wrist Pitch/Yaw)	±110°
	T-Axis (Wrist Twist)	±180°
Maximum Speed	Rotation-Axis	130°/s
	S-Axis	170°/s
	L-Axis	170°/s
	E-Axis	170°/s
	U-Axis	170°/s
	R-Axis	200°/s
	B-Axis	200°/s
	T-Axis	400°/s
Approximate Mass		220 kg (485.1 lbs)
Power Consumption		2.7 kVA
Allowable Moment	R-Axis	31.4 N • m
	B-Axis	31.4 N • m
	T-Axis	19.6 N • m
Allowable Moment of Inertia	R-Axis	1 kg • m^2
	B-Axis	1 kg • m^2
	T-Axis	0.4 kg • m^2

DX100 CONTROLLER SPECIFICATIONS**

Dimensions (mm)	1,200 (w) x 1,000 (h) x 650 (d) 47.2" x 39.4" x 25.6")
Approximate Mass	250 kg max. (551.3 lbs)
Cooling System	Indirect cooling
Ambient Temperature	During operation: 0° to 45° C (32° to 113° F) During transit and storage: -10° to 60° C (14° to 140° F)
Relative Humidity	90% max. non-condensing
Primary Power Requirements	3-phase, 240/480/575 VAC at 50/60 Hz
Digital I/O NPN-Standard PNP-Optional	Standard I/O: 40 inputs/40 outputs consisting of 16 system inputs/ 16 system outputs, 24 user inputs/24 user outputs 32 Transistor Outputs; 8 Relay Outputs Max. I/O (optional): 2,048 inputs and 2,048 outputs
Position Feedback	By absolute encoder
Program Memory	JOB: 200,000 steps, 10,000 instructions CIO Ladder Standard: 15,000 steps Expanded: 20,000 steps
Pendant Dim. (mm)	169 (w) x 314.5 (h) x 50 (d) (6.7" x 12.4" x 2")
Pendant Weight	.998 kg (2.2 lbs)
Interface	One Compact Flash slot; One USB Port (1.1)
Pendant Playback Buttons	Teach/Play/Remote Keyswitch selector Servo On, Start, Hold, and Emergency Stop Buttons
Programming Language	INFORM III, menu-driven programming
Maintenance Functions	Displays troubleshooting for alarms, predicts reducer wear
Number of Robots/Axes	Up to 8 robots, 72 axes
Multi Tasking	Up to 16 concurrent jobs, 4 system jobs
Fieldbus	DeviceNet Master/Slave, AB RIO, Profibus, Interbus-S, M-Net, CC Link, EtherNet IP/Slave
Ethernet	10 Base T/100 Base TX
Safety	Dual-channel Emergency Stop Pushbuttons, 3-position Enable Switch, Manual Brake Release Meets ANSI/RIA R15.06-1999, ANSI/RIA/ISO 10218-1-2007 and CSA Z434-03

**See DX100 Controller data sheet (DS-399) for complete specifications

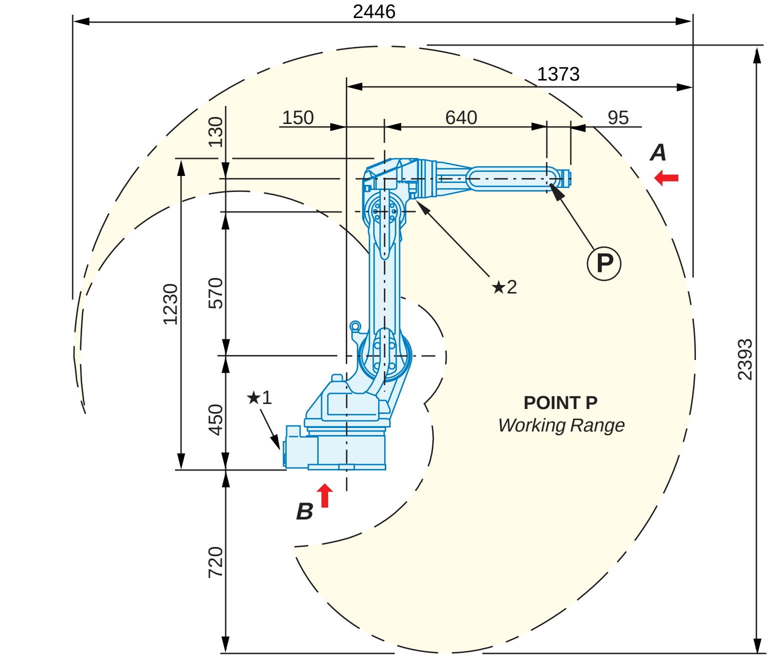
www.motoman.com

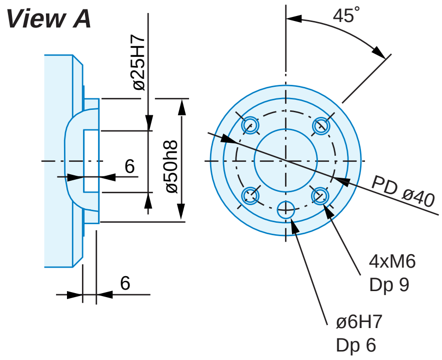
YASKAWA
MOTOMAN ROBOTICS

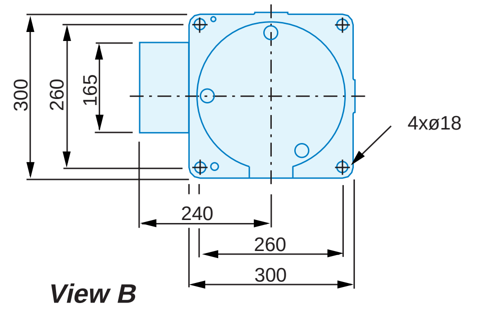
MOTOMAN ROBOTICS
805 LIBERTY LANE, WEST CARROLLTON, OHIO 45449
TEL: 937.847.6200 ■ FAX: 937.847.6277

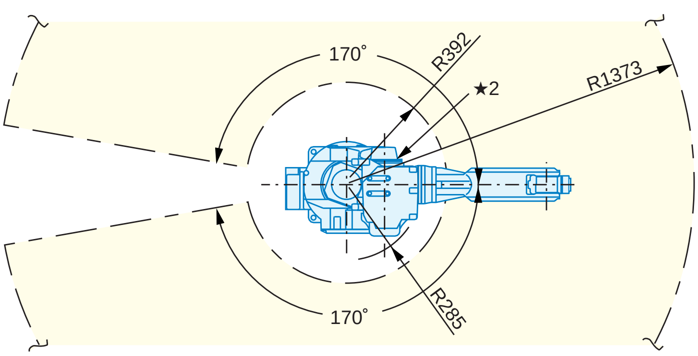
Industrial Robot

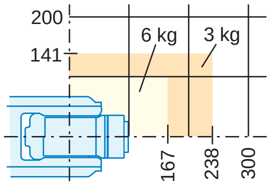
MOTOMAN-UP6











★	Air duct connector	Signal connector
1	1x PT 3/8, tap	16x wires + PE
2	1x PT 3/8, tap	16x wires + PE

Specifications

Controlled Axes		6
Payload		6 kg
Repetitive Positioning Accuracy		±0.08 mm
Motion Range	S-axis (turning)	±170°
	L-axis (lower arm)	+155°, -90°
	U-axis (upper arm)	+190°, -170°
	R-axis (wrist roll)	±180°
	B-axis (wrist pitch)	±135°
	T-axis (wrist twist)	±360°
Maximum Speed	S-axis	140°/s
	L-axis	160°/s
	U-axis	170°/s
	R-axis	335°/s
	B-axis	335°/s
	T-axis	500°/s

Weight		130 kgs
Power Supply		2 kVA
Allowable Moment	R-axis	11.8 Nm (1.2 kgf•m)
	B-axis	9.8 Nm (1.0 kgf•m)
	T-axis	5.9 Nm (0.6 kgf•m)
Allowable Inertia $GD^2/4$	R-axis	0.24 kg•m²
	B-axis	0.17 kg•m²
	T-axis	0.06 kg•m²
Ambient Conditions	Temperature	0 to 45°C
	Rel. Humidity	20 to 80% RH
	Vibration	0.5G or less
Others	• Free from excessive electrical noise. • Free from corrosive gas or liquid, or explosive gas.	

Reg 7Q-01-04-99GB

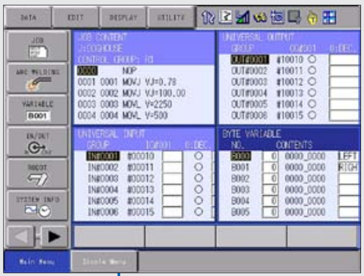
MOTOMAN ROBOTICS EUROPE AB
Box 504 • SE-385 25 Torsås • Sweden
Tel: +46-486-48800 • Fax: +46-486-41410
www.motoman.se

Group Companies: **France:** MOTOMAN Robotics, Nantes +33-2-40131919, **Germany:** MOTOMAN Robotec, Allershausen +49-8166-90-0 • MOTOMAN Robotec, Frankfurt +49-6173-6077-30, **Great Britain:** MOTOMAN Robotics, Banbury +44-1295-272755, **Italy:** MOTOMAN Robotics, Modena +39-59280496, **Netherlands:** MOTOMAN Benelux, Breda +31-76-5424278, **Slovenia:** RISTRO, Ribnica +386-61-861113, **Spain:** MOTOMAN Robotics, Barcelona +34-3-6303478, **Sweden:** MOTOMAN Robotics Europe, Torsås +46-486-48800 • MOTOMAN Mecatron Robotic Systems, Kalmar +46-480-444600

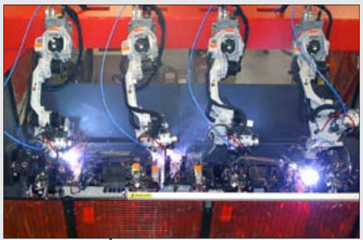
Distributors: **Czech Republic:** MGM Spol, Tabor +420-361-254571, **Denmark:** HN Automatic, Vejle +45-79428000, **Finland:** MOTOMAN-Robotit, Turku +358-22145600, **Greece:** Kouvalias Industrial Robots, Kallithea +30-1-95892436, **Norway:** Robot-Teknikk +47-32217830, **Portugal:** Electro-Arco, Amadora +351-14968160, **Switzerland:** Geiger Handling, Schwarzenburg +41-31-7343111 • Messer SAG, Dällikon +41-18471717

SOLUTIONS IN MOTION®

YASKAWA
MOTOMAN ROBOTICS



MULTIPLE WINDOW DISPLAY



MULTIPLE ROBOT CONTROL

DXM100 CONTROLLER

KEY FEATURES

- Patented multiple robot control (up to 8 robots/72 axes)
- Faster processing, high performance
- Integrated cell (system-level) control capabilities
- Open communication
- Energy savings
- Compliant to safety standards
- Controller connections through back of unit optimize floorspace

DX100

ROBOT/SYSTEM CONTROLLER

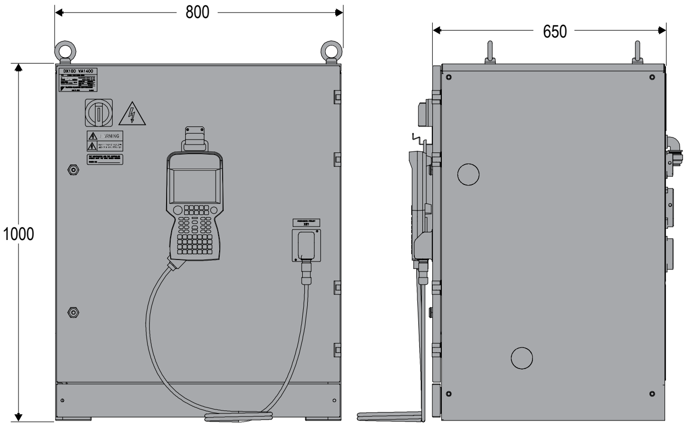
Option:
DXM100 Controller (smaller cabinet)

Dynamic Next-Generation Controller

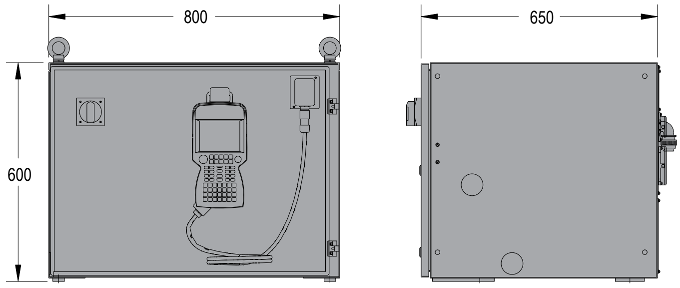
- Features robust PC architecture. Provides system-level control for robotic workcells.
- Patented multiple robot control (up to 8 robots/72 axes), as well as I/O devices and communication protocols. Dynamic interference zones protect robot arm and provide advanced collision avoidance.
- Fast processing speed provides smooth interpolation.
- Advanced Robot Motion (ARM) control provides high performance. Best-in-class path planning dramatically reduces teaching time.
- Small, lightweight Windows® CE programming pendant features color touch screen with multiple window display capability. Unique cross-shaped navigation cursor reduces teaching time. All operator controls are located on pendant. Program file names can be up to 32 characters long.
- Convenient compact flash slot and USB port facilitate memory backups.
- Conserves power during robot idle time, providing up to 25% energy savings.
- Highly flexible fieldbus support. Easy connection to information infrastructure through standard network options.
- Compliant to ANSI/RIA R15.06-1999 and other relevant ISO and CSA safety standards. Includes dual-channel E-Stop functionality, integrated speed monitoring and manual brake release for robot. Optional Category 3 functional safety unit.
- Often eliminates need for separate PLC and human machine interface (HMI). Delivers significant cost savings at system level, while decreasing workcell complexity and improving overall reliability.
- Connections to controller cabinet are made through the back of the unit, optimizing floorspace.
- DX100 control cabinet allows for up to three external axes and can be remote-mounted. DXM100 supports up to two external axes. Top- or side-mount expansion options available for DX100 controller only.
- Easy maintenance with reducer status check function, enhanced troubleshooting and alarm recovery, and 20% improvement in MTTR.

DX100 ROBOT CONTROLLER

DX100 Controller



DXM100 Controller



All dimensions are metric (mm) and for reference only.
Please request detail drawings for all design/engineering requirements.

Standard I/O - NPN

Forty optically isolated inputs, 32 transistor outputs, 8 relay contact outputs (configured to optimize each application), and four break-out cards are provided as standard. For arc welding applications, one YEW01 welder interface board is installed in the DX100 cabinet as standard (not available in DXM100).

I/O Expansion - DX100

The DX100 supports I/O expansion via:

- EtherNet/IP
- DeviceNet
- Profibus-DP
- Mechatrolink II
- CC-Link
- Remote I/O
- Discrete I/O, NPN or PNP
- Analog I/O
- Other networks available

I/O Expansion - DXM100

The DXM100 supports I/O expansion via:

- EtherNet/IP
- DeviceNet
- Remote I/O
- CC-Link

DX100 ROBOT CONTROLLER SPECIFICATIONS

CONTROLLER

Dimensions	DX100: 800 (w) x 1000 (h) x 650 (d) (31.5" x 39.4" x 25.6") DXM100: 800 (w) x 600 (h) x 650 (d) (31.5" x 23.6" x 25.6")
Approximate Mass	150-250 kg (330.8-551.3 lbs.)
Cooling System	Indirect cooling
Ambient Temperature	During operation: 0° to 45° C (32° to 113° F) During transport and storage: -10° to 60° C (14° to 140° F)
Relative Humidity	90% max. non-condensing
Primary Power Requirements	3-phase, 240/480/575 VAC at 50/60 Hz
Digital I/O NPN - Standard PNP - Optional	Standard I/O: 40 inputs/40 outputs consisting of 16 system inputs/16 system outputs, 24 user inputs/24 user outputs 32 Transistor Outputs; 8 Relay Outputs Max. I/O (optional): 2,048 inputs and 2,048 outputs
Position Feedback	Absolute encoder
Program Memory	JOB: 200,000 steps, 10,000 instructions CIO Ladder Standard: 15,000 steps Expanded: 20,000 steps
Interface	Ethernet, RS-232C
Multiple Robot Control	Ability to control up to 8 robots/72 axes

SAFETY FEATURES

Safety Specs	Controller Dual-Channel Emergency Stop & Safety Gate user interface. Programming Pendant includes: Dual-channel Emergency Stop Pushbutton, 3-Position Enable Switch with key-lock and Manual Brake Release built into programming pendant. Meets ANSI/RIA R15.06-1999, ANSI/RIA/ISO 10218-1-2007 and CSA Z434-03
Collision Avoidance	Collision avoidance zones and radial interference zones
Collision Detection	Protects robot by monitoring torque levels on manipulator
Machine Lock	Permits testing of peripheral devices without robot operation
Safety Interlock	Prevents robot operation while safety circuit is open

PENDANT

Pendant Dimensions	169 (w) x 314.5 (h) x 50 (d) (6.6" x 12.4" x 2")
Pendant Display	5.7-inch full-color touch screen, 640 x 480 (VGA)
Pendant Languages	English, German, Japanese, Spanish, Chinese
Pendant Weight	.998 kg (2.2 lbs)
Coordinate System	Joint, rectangular, cylindrical, tool, 24 user-coordinate frames
Windows® Menu-Driven Interface	User-selectable touch-screen menu, Multiple windows supported One Compact Flash slot; One USB port (1.1)
Pendant O/S	Windows® CE
Protection Rating	IP65

PROGRAMMING

Programming Language	INFORM III, menu-driven programming
Robot Motion Control	Joint motion, linear, circular, spline interpolation
Speed Adjustment	Percentage of maximum for joint motion; mm/sec, cm/min, in/min for displacement; °/sec for orientation
Device Instructions	Application-specific (ARCON, ARCOFF, LASERON, LASEROFF, HANDON, HANDOFF)
I/O Instructions	Discrete I/O, 4-bit and 8-bit manipulation, analog output, analog input, analog scaling, sloping
Operation	Up to 5 levels of undo/redo

MAINTENANCE

Maintenance Functions	System monitor, internal maintenance clocks
Self-Diagnostics	Classifies errors and major/minor alarms and displays data
User Alarm Display	Displays alarm messages for peripheral devices
Alarm Display	Alarm messages and alarm history
I/O Diagnosis	Permits simulated enabled/disabled input/output
TCP Calibration	Automatically calibrates parameters for end-effectors, optional TCP recovery function
Tool Weight Calibration	Automatically calibrates total weight of tool, center of gravity and inertia for peak performance

www.motoman.com

YASKAWA
MOTOMAN ROBOTICS

MOTOMAN ROBOTICS
100 AUTOMATION WAY, MIAMISBURG, OHIO 45342
TEL: 937.847.6200 ■ FAX: 937.847.6277

Technical data

Controller

Configuration	Free-standing, Enclosed type
Dimensions	800(W)x900(H)x650(D) mm
Weight	Approx. 170 kg
Cooling system	Indirect cooling
Ambient temperature	During operation 0° to +45° During transport -10° to +60°
Relative humidity	Max. 90% (non-condensing)
Power supply	3 x 400/415/440V AC, 50/60 Hz
Grounding	Less than 100 ohm
Digital I/O	Specialised signal (hardware) 12 inputs and 3 outputs General signals (standard) 40 inputs and 40 outputs 4 direct inputs
Positioning system	Absolute encoder / Serial interface
Drive units	Servopacks for AC servomotor
Accel / Decel	Software servo control
Programming capacity (standard)	5.000 steps and 3.000 instructions 1.500 ladder steps

Safety features

Personal safety	3-position "dead-man´s handle" Low speed in teaching mode
Teach Lock Mode	Prohibits operation from operator´s panel
Collision proof frames	Doughnut-sector frame, cubic frame
Machine lock	Test-run peripherial device without robot motion
Self-diagnosis	Alarm and error messages displayed together with explanation
User Alarm display	Possible to display alarm messages for peripheral devices

Operator´s panel

Buttons provided	Mode, Start, Hold, Emergency stop Servo power ON

Programming pendant

Material	Reinforced thermoplastic enclosure
Dimensions	211(W)x382(H)x71(D) mm
Weight	1.2 kg
Display	5.7 inch, 40 characters x 12 lines
Safety feature	3-position "dead-man´s handle"
Interface	RS-232C

Programming functions

Coordinate system	Joint, rectangular/cylindrical, tool, user coordinates
Robot Motion Control	Joint coordinates, linear/circular, interpolation, tool coordinates
Speed setting	Percentage for joint coordinates, 0.1 mm/s units for interpolations, angular velocity for T.C.P. fixed motion
Program Control Instructions	Jump, call, timer, robot stop, execution of some instructions during robot motion
Modification of teaching point	Adding, deleting, correcting (robot axes and/or external axes)
Position control	Manually forwards and backwards in the job (even circular)
Speed adjustment	Fine adjustment possible
I/O-function	Discrete I/O control, pattern I/O processing
Programming Language	Interactive programming Robot language: INFORM II
Display text	English, Swedish, French, Spanish, Italian, Finnish, German
Tool Centre Point	Max. 24 and up to 24 external
TCP-calibration	TCP´s Automatically calibrates parameters for end effectors using master jig

Maintenance functions

Software time usage meters	Control power-ON time, servo power-ON time, playback time, work time and operation time displayed
Alarm display	Alarm messages and previous alarm records
I/O-diagnosis	Simulated enable/disabled output possible

Options

Digital I/O	I/O-boards, total max. 256/256 MIO02: 32 inputs and outputs MIO03: 16 inputs and outputs
Analog output	12 channel (MEW/XEW-board)
Memory expansion (up to...)	Max 60.000 points for 6 axes and 20.000 instructions 3.000 ladder steps
External axis	Total max. 27 axes
Enclosure classification	IP54, by add on kit

MOTOMAN®

XRC Robot Controller

Software functions

Example of general functions

- ARM control
- Vibration control
- Station coordinated motion control
- P-start function
- Coordinated motion 2 manipulators
- Coordinated motion 3 manipulators
- Coordinated motion 3 manipulators and ex-axis
- Twin drive function
- Welding condition slope up/down
- Interrupt job function
- Search function
- Servo float function
- Linear servo float function
- T-axis endless rotation
- External axis endless rotation
- External reference point control function
- PMT function
- Start point search function
- High speed start point search function
- General purpose sensor function
- Data tansmission function
- Relative job function
- Parallel shift function
- PAM function
- TCP function
- Weaving function
- Pause weaving function
- Online tool modification function
- Independant control function (6 tasks)
- External storage function PC-Card
- Analogue output function
- Analogue output function related to speed

Examples of special functions

- COMARC arc welding control
- Multi-layer welding function
- Conveyor synchonous function
- Press synchronous function
- Laser sensor function
- Pitch control for spot welding
- Ethernet communication
- Fieldbus communication

Available optional boards

- 16/16 IO-board
- 32/32 IO-board
- MEW-Welding interface board
- XEW-Welding interface board
- Ethernet communication board
- Fieldbus communication board
- Sensor board

Software applications

- Arc welding application
- Handling application
- Spot welding application
- General purpose application

MOTOMAN ROBOTICS EUROPE AB
Box 504 • SE-385 25 Torsås • Sweden
Tel: +46-486-48800 • Fax: +46-486-41410
www.motoman.se

Reg 7P-01-04-99GB

Group Companies: **France:** MOTOMAN Robotics, Nantes +33-2-40131919, **Germany:** MOTOMAN Robotec, Allershausen +49-8166-90-0 • MOTOMAN Robotec, Frankfurt +49-6173-6077-30, **Great Britain:** MOTOMAN Robotics, Banbury +44-1295-272755, **Italy:** MOTOMAN Robotics, Modena +39-59280496, **Netherlands:** MOTOMAN Benelux, Breda +31-76-5424278, **Slovenia:** RISTRO, Ribnica +386-61-861113, **Spain:** MOTOMAN Robotics, Barcelona +34-3-6303478, **Sweden:** MOTOMAN Robotics Europe, Torsås +46-486-48800 • MOTOMAN Mecatron Robotic Systems, Kalmar +46-480-444600

Distributors: **Czech Republic:** MGM Spol, Tabor +420-361-254571, **Denmark:** HN Automatic, Vejle +45-79428000, **Finland:** MOTOMAN-Robotit, Turku +358-22145600, **Greece:** Kouvalias Industrial Robots, Kallithea +30-1-95892436, **Norway:** Robot-Teknikk +47-32217830, **Portugal:** Electro-Arco, Amadora +351-14968160, **Switzerland:** Geiger Handling, Schwarzenburg +41-31-7343111 • Messer SAG, Dällikon +41-18471717

MOTOMAN XRC Robot Controller

MOTOMAN XRC will be the premier robot controller.
It is physically minimised but optimised for functions, simple operations and expandability.

Outlines of the XRC

Built-in transformer
Simple installation and setup by plug-connections.
Used for robot in UP-series and SK-X, SP-X and SV-X-series.
INFORM programming language
Expandable slots for optional boards
-Sensor function by MSL-board
-Welding interface MEW and XEW-board
-I/O-interface MIO-board
-Ethernet and Fieldbus board
XRC cabinets may be put on top of each other or docked side by side
Fulfils the EC-directive incl. EMC and LVD
Made in Sweden

Advanced control system

Powerful 32-bit micro processor for rapid data processing.
PLC-programs for different applications, for example: arc welding and handling.
Advanced PLC-unit in the control system with expanded program capacity and new logical functions e.g. for controlling pneumatic functions in fixtures.
Multi tasking makes it possible to run several jobs simultaneously.
Digital servopacks with software controlled speed and position feedback.

Shock detection function

This function stops the robot when it comes into contact with an object. Robot or peripherical devices can be protected in case of accidental collision.

Maintenance functions

Remote maintenance function monitors remotely robot status through ethernet network. Production processing monitoring and quality control can be managed from a central control room.
Maintains history of when and who changed robot job instructions.

Programming pendant

All programming functions are in one unit with clear and large 12-line screen. Optimal programming time is achieved while all functions are available during programming. The menu text can be switched between several different European languages by a simple key operation. The programming pendant is equipped with a 3-position "dead-man´s handle".

Communication

XRC is like the previous control system for MOTOMAN robots a data technical solution that is PC-compatible. Therefore common computer tools may be used for creating, reading and editing jobs in a PC.
The flexibility of the control system makes it possible to communicate with other systems such as PLC´s, host computers, vision system, etc. through serial links, Ethernet, digital I/O, fieldbus. Controller memory is also accessable through PC-card.

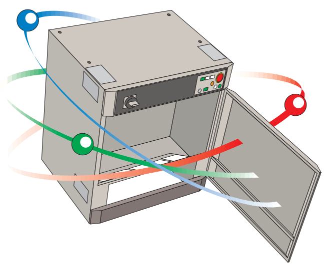
Path control

Robot working path is controlled to perform consistently precision and quality. Even at high speed, the robot will not vary from the target execution line.
XRC improves high speed welding, cutting and sealing more than ever.

ARM control

Advanced Robot Motion Control improves quality and productivity by improved servo response for the motion control. Optimised acceleration/deceleration and vibration control obtains correct path and cycle time. Speed is automatically optimised for circles and corners. Collision between robot, robot tool and other devices is immediately detected to stop robot and protect from damages.

Controller features



LCD-display
Large LCD-display with back-lighting. High contrast for day-light operation.

Cursor operation
Cross shape cursor button for instruction basic operations. Operates by icons and pull-down menues.

Lightweight
Lightweight thermoplastic p-pendant reduces fatigue.

Key operation
Key allocation is based on human engineering research. The number of keys are minimised for required functions only.

3 Robots
CPU can control up to 3 robots simultaneously.

27 Axes
CPU can control up to 27 axes simultaneously.

6 Tasks
Multi-task CPU can handle up to 6 tasks simultaneously.

Easy placing
Cables and air vent only at back. Optional cables, IO's, etc. are accessed behinde the front door.

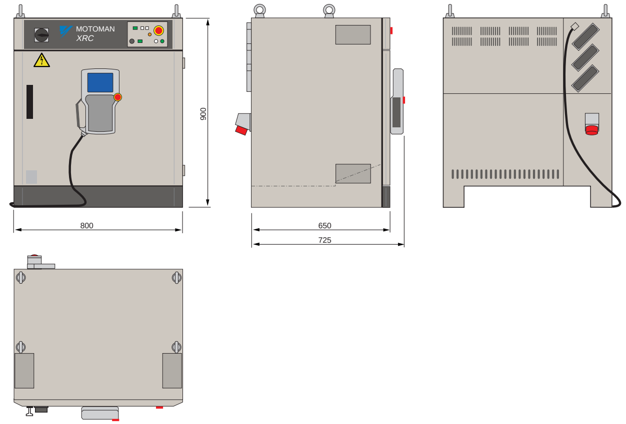
Easy installation
Cables for power, robots and IO´s are connected by plugs.

Expansion units
Optional units such as external axis servopacks may be placed in optional cabinets. These cabinets are docked to the controller cabinet by means of ports on top and on the sides. Which means tidy cable installation.

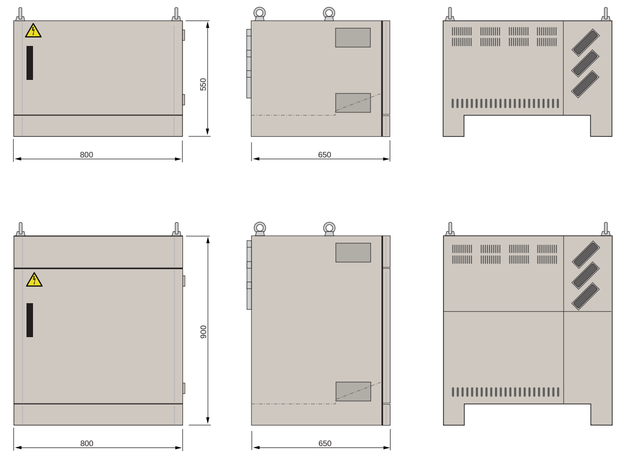
PC-Card
PC-Card (PCMCIA) is used for expansion interface. It simplifies expanding functions and speeds up backing up data.

Network
Standard field networks are applicable: Ethernet, Profibus, Interbus-S, Device-NET, etc. Your manufacturing line can be integrally controlled through your existing network.

Controller layout



External axis cabinet layout



Net F/T

Six-Axis Force/Torque Sensor for Ethernet and EtherNet/IP™

ATI INDUSTRIAL AUTOMATION
ISO 9001 Registered

Product Description

The Network Force/Torque (Net F/T) sensor system measures six components of force and torque (Fx, Fy, Fz, Tx, Ty, Tz). The Net F/T provides an EtherNet/IP and CAN bus communication interface and is compatible with standard Ethernet. The Net F/T system is available with all of our transducer models.

Product Features

- Fully ODVA™-compliant EtherNet/IP interface.
- Environmental sealing is available. The Net F/T interface (Net Box) is sealed to IP65 (water splash-proof), most Net F/T transducers and cables can be sealed to IP68 (submergibility in fresh water, to a depth of 10 meters).
- Powered by Power over Ethernet (PoE) or by an external power supply (11VDC to 24VDC).
- Multiple transducer calibrations can be permanently stored in the system and can be selected by the user.

Product Advantages

Multiple Interfaces: Ethernet, EtherNet/IP and CAN bus, to support a wide range of automation and research applications.

LAN Connectivity: The Net F/T can be easily connected to your Local Area Network (LAN) allowing for easy remote operation and monitoring.

High-speed output: Output rates up to 7000 Hz for six axes of measurement over Ethernet (using UDP).

Web-based Configuration and Demo: The Net F/T is configurable via a web interface, allowing the user to view and change system settings using a web browser. An easily accessible Java™ demonstration application provides a graphical view of six-axis measurements in real time.

Programmable Thresholding: Allows monitoring of force and torque conditions via a relay output or status bits to allow immediate action in case of critical process conditions.

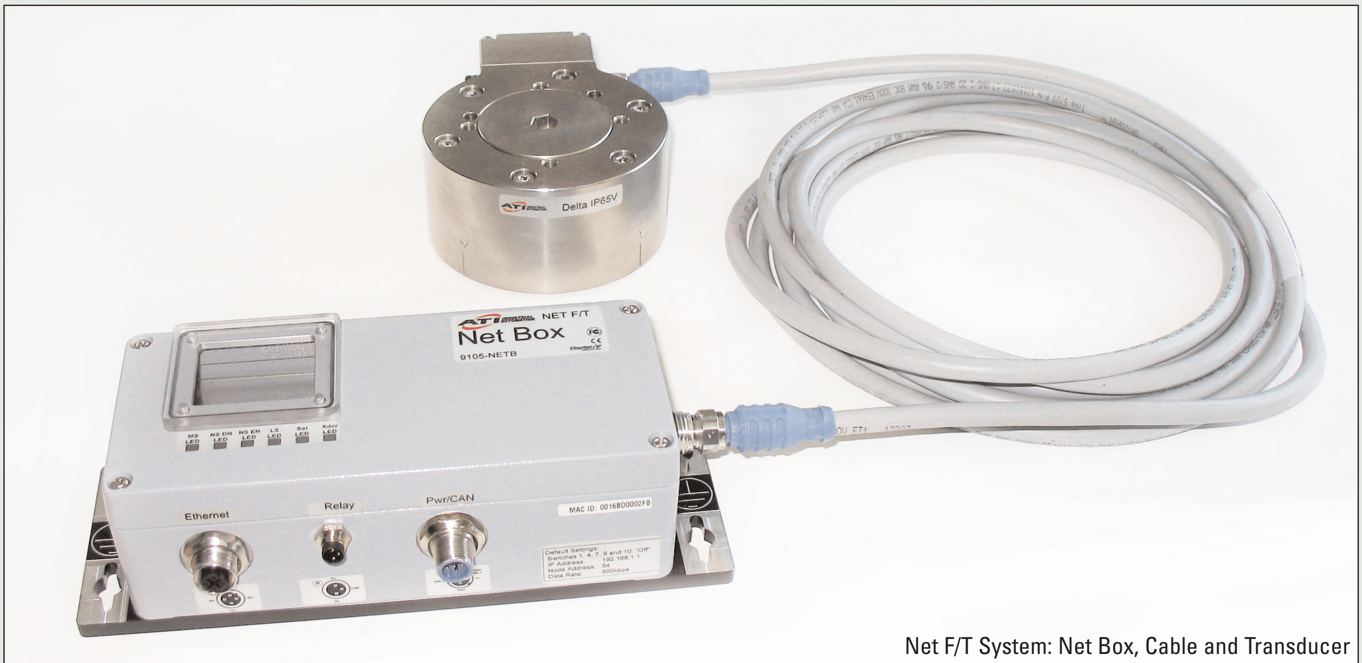
Built-in Status Indicators: LEDs in the Net Box alert the user to connection and load saturation status.

Overload protection: The F/T transducer is extremely rugged and durable. The transducer's factor of safety can be as high as 4080%, depending on model and calibration.

Tool Transformations: Tool transformations can be used to translate and rotate the F/T point of origin to make measurements relative to the work at hand.

Temperature Compensation: Each F/T transducer features hardware temperature compensation to stabilize its sensitivity over temperature. This compensation method optimizes the transducer's accuracy over a range approximately ±25°C from room temperature.

High signal-to-noise ratio: Silicon strain gages provide a signal 75-times stronger than conventional foil gages, reducing the need for noise-increasing gain.

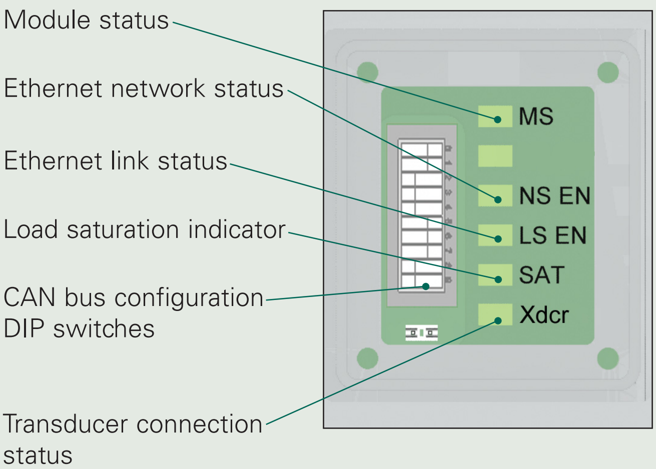


Net F/T System: Net Box, Cable and Transducer

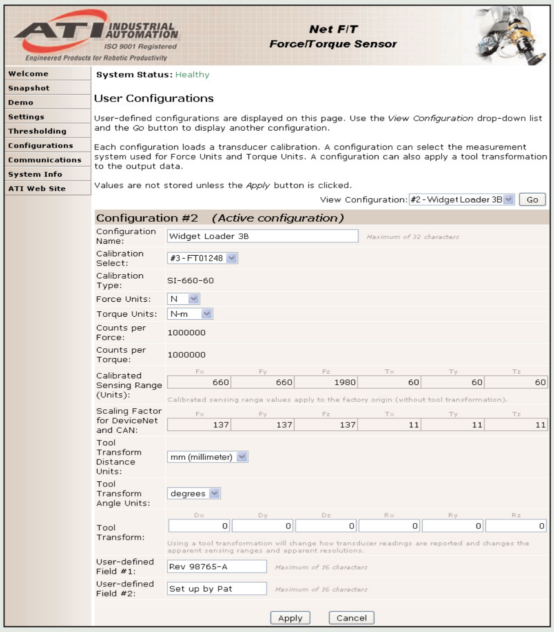
User Interface Overview

Sensor system setup and status monitoring can be done through the status LEDs and DIP switches or via an Internet browser. Advanced functions and the demonstration application are also available via a browser

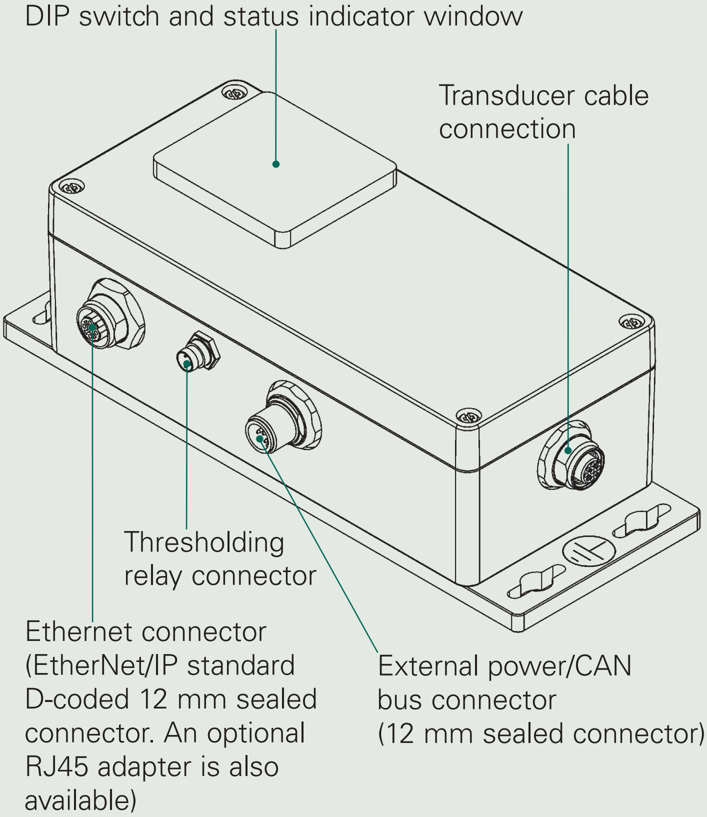
Indicator Window Functions



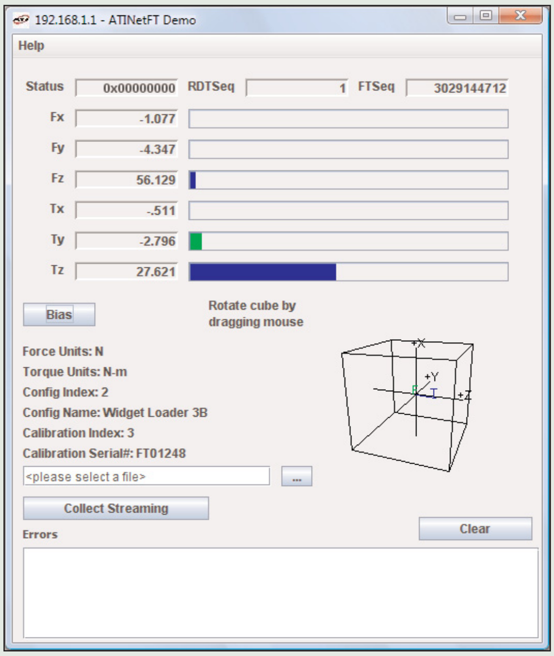
Easily-Accessible Configuration Settings



User I/O



Self-Contained Demonstration Application



ATI INDUSTRIAL AUTOMATION
ISO 9001 Registered
Engineered Products for Robotic Productivity

Pinnacle Park
1031 Goodworth Drive
Apex, NC 27539 USA

+1 919.772.0115
+1 919.772.8259 fax
E-mail: info@ati-ia.com
www.ati-ia.com

9205-05-1016-03 November, 2008

BARIONET FAMILY

Programmable I/O device server with two serial ports, Dallas 1-wire support, SNMP, Modbus/ TCP and web interfaces via built-in 10/100 Ethernet port.

The Barionet is a fully programmable I/O device server that supports standard interfaces such as SNMP, CGI, HTTP, and Modbus/TCP to access local I/O and program functions. Support for a broad variety of standard protocols and the built-in 10/100 Ethernet port makes the Barionet devices ideal for automation and monitoring applications in buildings, industry and IT systems.

Barionet family common hardware features:

- 10/100 Mbit Ethernet interface
- Two serial ports (RS-232 and RS-485)
- Very low power operation (typically 1.5 watts with no relays activated, 4 watts maximum)
- Wide supply voltage input range: 9 - 30V
- Relay outputs and digital inputs
- Dallas 1-wire interface

Software applications available for the Barionet devices:

The Barionet comes pre-loaded with a web configuration and user interface application for monitoring and control of all on-board I/O functions. In addition, full custom applications can be developed using the built-in BCL interpretter (Barix Control Language, which is very similar to BASIC). A demo application, including source code is also included that provides a useful serial tunnelling function, but also serves as a BCL programming example.

The standard software provides the following functionality:

- Browser-based configuration and I/O monitoring pages built-in
- Support for custom dynamic (DHTML) web pages.
- TCP, UDP, CGI, Modbus/TCP, SNMP protocol support
- All I/O functions can be monitored or controlled via built-in standard protocols or using custom DHTML pages and BCL software (Barix Control Language, which is very similar to BASIC).
- Support for 50 1-wire temperature sensors and a 1-wire Real Time Clock
- Example Serial gateway/tunnel BCL program with source code pre-loaded provides serial „tunnelling" across an IP network, and also serves as a BCL programming example.

Available BCL applications for the Barionet:

- Digital I/O tunnel (forwarding of contact closures over the network)
- Example BCL program for controlling and monitoring Barix Modbus expansion modules (R6, IO12, and X8).
- General alarming/monitoring/reporting application for security, monitoring, and supervision applications
- RTP replicator/rebroadcaster
- home automation master
- weather station interface
- NTP server
- simple access controller for card readers with Wiegand interface
- sauna controller
- energy counter/usage reporting

Barix Barionet (100)

Programmable I/O device server with 2 serial ports, relay and digital outputs, digital and analog inputs and Dallas 1-wire support., SNMP, Modbus/TCP and web interfaces via built-in 10/100 Ethernet interface. UL listed.

The Barionet 100 is a fully programmable I/O device server that supports standard interfaces such as SNMP, CGI, HTTP, and Modbus/TCP to monitor and control a variety of built-in I/O functions. Four analog inputs, four contact closure inputs, four open collector digital outputs, two relay outputs. a 1-wire interface and two serial ports can all be monitored and controlled via several built-in protocols. In addition, all I/O functions and interfaces can be controlled and monitored using custom DHTML web pages and BCL (Barix Control Language) applications loaded into the device

Applications

- Facility management system
- Building automation
- Remote control and monitoring of machines
- Data acquisition, storage and transmission in any technical system

Barionet 100 hardware specific features:

- 4 contact closure inputs with programmable pull-up resistor
- 4 analog inputs, 0-5V range, 12 bit resolution, programmable pull-up (can be used as contact closure inputs as well)
- Contact closure inputs also support use as Wiegand reader interface
- 4 open collector (digital) outputs, current limited
- 2 relay outputs, switching up to 240VAC, 5A