DESIGN AND ANALYSIS OF A ROBOTIC SURGICAL MANIPULATOR FOR COCHLEOSTOMY

Buddhi Arjuna Mendis

(108859E)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

June 2015

STUDENT DECLARATION

"I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters dissertation under my supervision.

Signature of the supervisor: Date

Dr. D. P. Chandima

The above candidate has carried out research for the Masters dissertation under my supervision.

Signature of the supervisor: Date

Dr. A. D. K. S. N. Yasawardena

ABSTRACT

Robot assisted surgery is proven to be useful in surgeries, proven to be complex in conventional form in terms of accessibility anatomical complexity and small scale, required precision and accuracy. Cochleostomy procedure in cochlear implantation surgery is one such procedure, proven to be a complex practice even for the most experienced surgeon.

In this thesis, the drilling processes involved in conventional cochleostomy are looked at. Due to dexterity and precision robotics offer, it is deemed the efficiency of the in situ drilling procedure of the cochleostomy can be greatly increased with the use of a robotic manipulator tool.

Despite commercial success of general robotic platforms, practical use in task specific microsurgery is still challenging, due to considerable levels of accuracy required at sub-millimeter scales, limited visualization, degrees of freedom, range of motion, large footprint and constrained visual and tool accessibility, under operation microscopes. The proposed task specific surgical manipulator addresses the drawbacks of existing surgical manipulators and other apparatus for the purpose of cochleostomy. The proposed tool: a six degrees of freedom manipulator, is a micromanipulator that is attached to the surgical microscope boom. The surgeon is able to use the manipulator as conventional surgical drill tool for drilling and clearing of bone.

The thesis looks at the development of the introduced surgical manipulator; from concept, theory to a proof of concept prototype. The theoretical analysis, theoretically formulates the concepts, which are the basis of the manipulator design. The theoretical study includes a study of manipulator kinematics, manipulator singularities, analysis of the systems dynamic parameters and the controller design in joint space. Methods of localization and trajectory generation are briefly discussed and validated using simulation.

A simple prototype is developed based on the developed concepts and theoretical formulation. The prototype development includes design of mechanical linkages, drive actuators, a robot controller and software. Simple tests are conducted using the developed prototype to validate required motion control.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. D. P. Chandima from the department of electrical engineering, University of Moratuwa for his patience, advice and constant encouragement, and my external supervisor Dr. A. D. K. S. Yasawardena, ENT surgeon at the Lady Ridgeway Hospital, for his expert advice, and access to observe surgical procedures.

I thank Mr. Tilak Dissanayake for his advice & support.

I would like to acknowledge and appreciate the support given by J. M. Wickramarachchi & Co., local partner for Cochlear Ltd., with information and correspondence.

I am truly grateful for the support the Electrical department at the university has offed me, which enabled me to complete this research.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC – Analog to Digital Converter

AMC – Auxiliary Microprocessor

ANSI – American National Standards Institute (of USA)

CAD – Computer Aided Design

CT– Computer Tomography

DAC – Digital to Analog Converter

DC – Direct Current

DOF – Degrees of Freedom

EE – End Effector

ENT – Ear, Nose, Throat

FDA – Food and Drugs Administration (of USA)

GUI – Graphical User Interface

IC – Integrated Circuit

ICP – Iterative Center Point (algorithm)

I2C – Inter-Inter-Connected (bus)

JC – Joint Controller

JCM – Joint Controller Microcontroller

LRH – Lady Ridgeway Hospital

MC – Master Controller

MDH – Modified Denavit-Hartenburg

PC – Personal Computer

PCS – PC side Software

PMDC – Permanent Magnet Direct Current (Motor)

RISC – Reduced Instruction Set Computing

SPI – Serial Peripheral Interface

SVD - Singular Value Decomposition

TCP - Tool Center Point

USB – Universal Serial Bus

1 INTRODUCTION

In otolaryngology, cochleostomy is a process which requires precise drilling of the temporal bone. A number of functionally important organs, nerves, blood vessels and structure e.g. facial nerve and the chorda tympani, are embedded in the temporal bone at the vicinity of drilling, increasing procedure complexity. An experienced surgeon needs to be cautious, while using the precision hand held surgical drill, as damage caused to these structures would result in complications for the patient.

This research investigates the prospect of developing an electro-mechanical solution for easing the cochleostomy procedure, specifically a task specific robotic surgical micromanipulator for in situ drilling of the temporal bone required for the cochleostomy procedure. The resultant design, analysis and a prototype evaluation for the investigation is presented in this thesis.

The aims of the following sections are to provide the background information on the thesis research. Section 1.1 provides a description and shortcoming of the surgical procedure. Section 1.2 is a review of literature on concurrent study, their solutions and drawbacks. Section 1.3 proposes a new surgical manipulator design and changes to conventional the surgical process as a solution to the shortcomings of previous solutions. Section 1.4 provides an outline of the impending chapters of the thesis.

1.1 Surgical Procedure and Cochleostomy Requirement

Cochleostomy, the process of opening of the cochlear, is a requirement for the cochlear implant surgery, a treatment for severe to profound loss of hearing. At a cochlear implant surgery, an electrode array is implanted into the inner ear. This device bypasses the functionality of the outer and the middle ear, and directly stimulates the nerve cells in the inner ear. The current surgical procedure begins with the mastoidectomy procedure, in which an open cavity of 35[mm] depth is carved out of the temporal bone, in the area behind the ear using a high speed hand held surgical drill. The procedure exposes a number of functionally critical and sensitive

anatomical structures, such as organs and nerves in the temporal bone area. These features need to be identified and preserved, as the drill burr passes in very close proximity less than 1[mm]. Identifications of these sensitive features are a requirement for the location of the cochleostomy.

The exposed features are; the facial recess between the canal of the facial nerve and the chorda tympani. After tympanotomy at this position, the promontory, the round window niche and the stapes are identified. The cochleostomy is performed either at the round window niche or anterior to the round window [1].

The facial nerve and the chorda tympani, are separated by approximately 2[mm] at the facial recess. Damage to the facial nerve results in ipsilateral facial paralysis while damage to the chorda tympani results in ipsilateral loss of taste in the tongue. The drill burr must pass through these two nerves during the cochleostomy process.

For the cochlear implant process, a drilled hole of diameter 0.5 to 1[mm] is used for inserting the implant array, which is directed into the scala tympani in the cochlear. An ideally positioned cochleostomy should be placed within a deviation of 0.3 [mm], in order to prevent damages to the internal structure and tissue of the inner ear [2, 3]. Divergence causes perforation of the basilary membrane and dislocation of the electrode into the scala vestibuli, which results in loss of residual hearing.

In order to conduct successful cochleostomy and cochlear implantation, surgeon relies on anatomical knowledge and experience in order to correctly identify subsurface features relating to the anatomical structure. In order to expose sensitive features without accidently causing damage, the surgeon has to rely on hand-eye coordination. Accidently damaging the said nerves or encroaching on the ear canal, can lead to chronic infection. Perforation of the endosteal membrane by the drill may result in contamination of the endolymph and perilymph with bone dust; increase the risk of postoperative infection. Thus in order to enhance patient safety and reduce trauma, it is desirable to use robotics for automating this surgical procedure, thus removing reliance on human hand–eye coordination and spatial reasoning.

The hand held surgical drill uses micro-burrs with a coating of diamond dust for abrasive erosion of bone. The surgeon clears the bone with minimal force, while moving the drill in circular fashion. If the drill is kept stationary for too long, it would cause tissue to burn.

The surgeon relies on anatomical knowledge and experience. A computer tomography (CT) scan of the patient is acquired preoperatively is used for evaluating organ defects and patient specific uniqueness.

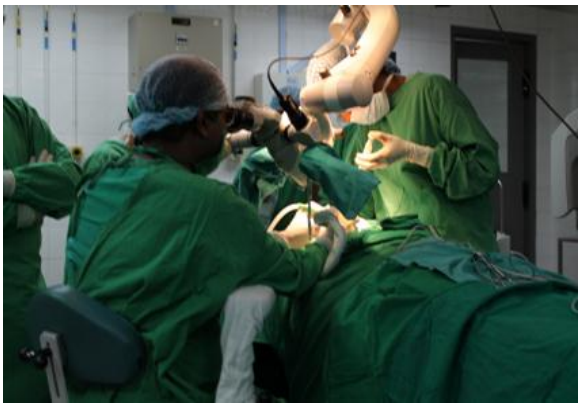

Figure 1.1: Cochlear implant surgery at the LRH. (Original in color)

Figure 1.2: Preparation for cochleostomy. (Original in color)

Figure 1.1 and Figure 1.2 shows the surgical theater during a typical cochleostomy surgery. Figure 1.1, Shows a surgeon seated performs a drilling process on a patient looking through a surgical microscope, while a supporting surgeon overlooks, two others are present to support tasks such as removal of debris, cleaning. Figure 1.2 shows preparation for cochleostomy and Figure 1.3 shows typical cochleostomy procedure. Figure 1.1 and Figure 1.2 are taken from surgeries the author observed at the Lady Ridgeway Hospital (LRH).

Figure 1.3: Typical cochleostomy procedure. (Original in color)
Source: Cochlear Ltd.

1.2 Literature Review

Use of robotics is becoming increasingly popular in surgery in the recent past [4], resulting in an increase in exploration for utilizing and integrating robotics for the enhancement of surgical procedures and integration of robots in the clinical workflow.

In initial experiments in robotics surgery assistance, industrial robots were used for experimenting with surgical purposes. The first use of an industrial robot, Puma 560 for neurosurgical biopsies was recorded in 1985. These followed the introduction of task specific robotic systems. The ROBODOC was the first task specific medical robot system, which was certified by the FDA. The physical size of general surgical systems and industrial systems user at early stages in surgery are large in footprint and has cumbersome arms, requiring large booms in the surgical theater.

To this date, a vast number of medical robotic systems exist in clinical use or in clinical trials, but cochleostomy specific robotics exist only in research and are in clinical test phases, none in the clinical use phase.

Initial experiments in use of robotic devices for the drilling part of the cochlear surgery were conducted by a number of independent groups. The research was pioneered by Lenarz et al. [2], a minimally invasive approach is proposed, in which the drilling creates an access canal to the inner ear, and perform cochleostomy using

a general purpose industrial robotic manipulator arm KUKA KR3. A minimal invasive approach, the surgery is conducted without performing mastoidectomy and exposing critical anatomical structure. The system uses a camera and special markers to perform localization, pose estimation. Preoperative planning using a CT image was used for optimal drilling trajectory calculation, away from the critical features. The research is presented as a proof of concept of the adaptation of a robotic manipulator for the drilling procedure. The targeting errors were not low enough for clinical development. Industrial robots provided an excellent test bed for preliminary study, though by nature not meant for operation theatre environments, mainly due to size and stiffness [4],[5].

Minimally invasive robots for general surgery, e.g. DaVinci, KineMedic, although kinematically compatible with industrial robots, no report or results of use for cochleostomy have been published to date. Most popular robotic surgical platforms are specifically created for minimally invasive telerobotics. A drawback of using general purpose robotic surgical manipulators is the fact that they consume considerable amount of space in the surgical theater, thus at a given time, either the surgeon or the manipulator arm can occupy the surgical workspace. Thus if such a manipulator is used, for the drilling procedure of cochleostomy, the manipulator arm can be only used for autonomous mode operations.

Brett et al. [6] developed an autonomous micro drill system, specifically for the cochleostomy procedure. The system consists of a micro drill mounted on a linear guide, attached to a passive robot arm. The surgeon moves the arm to the required position and orientation of the desired drilling trajectory, following which the arm is locked, letting the drill autonomously create a hole, leaving the endosteal membrane intact. The drilling system analyses the forces and torques and identifies if breakthrough is about to occur in order to stop the drilling. This system proves to be efficient in terms of identifying breakthrough point and cease drilling. A drawback of this system was the manual setup operation of the passive robotic arm, as the positioning could be inaccurate. Manual setup also consumes considerable time.

An automated microstereostatic image-guided (AIM) frame prototype was studied by Labadie et al [7]. A miniature Stewart platform mounted on a patient specific frame is anchored on to the temporal bone by screws. While solving the localizing

problem accurately, the system obstructs vision, and the ability to use other tools etc. for cleaning the workspace. Another drawback is the complication of fabrication and attachment of the microstereostatic frame, a procedure which deviates from the clinical workflow.

Promising research on surgical robotics is carried out on miniature form of snake robots, Snake like manipulators enable high dexterity and speed using direct drive or cable actuators. A negative aspect is that snake robots are hard to control. CadioARM [8] is a cardiac surgical robot belonging to the group of snake-like manipulators, CardioARM has 102 joints is a highly dexterous complex manipulator.

Control of medical robots, are similar in nature to the control of industrial robots. Control of robotic micro-manipulators falls into two main categories: Joint space control and operational space control. Joint space control schema does not influence the operational space variables, which are controlled in an open loop fashion through the manipulator control structure. Any uncertainty of the structure: construction tolerance, lack of calibration, gear backlash, elasticity etc. can cause a loss of accuracy on the operational space variables.

Due to the nature of surgical micro-manipulators, with low inertia and gravity loading, high gearing ratios requiring relatively slow movement, joint space methods are deems to be sufficient, as non-linear effects are minimal[9].

Joint-space control is found in decentralized and centralized forms. In decentralized control, also known as independent joint control; each of the joints, are treated as independent joints. This, greatly simplify the control mechanism. In centralized control, the controller is model based, in which the manipulator dynamics model is a coupled differential equation. In all forms of centralized control, the manipulators dynamic equations of motion plays a greater role than in decentralized control schemas. This in general increases the computational requirements in comparison to decentralized control. There are a number of common Joint space centralized control schemas.

PD controller with gravity compensation: this is a common controller found in most industrial robots. PD with gravity compensation is adequate for most slow moving robots that do not use direct drive actuators. [9] Thus PD controller with gravity compensation is used as a benchmark for geared articulated manipulators.

Other general forms are: inverse dynamics, robust control – in the case of imperfect compensation and presence of disturbance factors, Adaptive control- adaptive control is also useful as a calibration procedure for robust control.

Operational space controller designs require greater algorithmic complexity. But the ability to act directly on operating space variables proves to be an advantage over joint space schemas.

1.3 Proposed Tool and Surgical Workflow Change

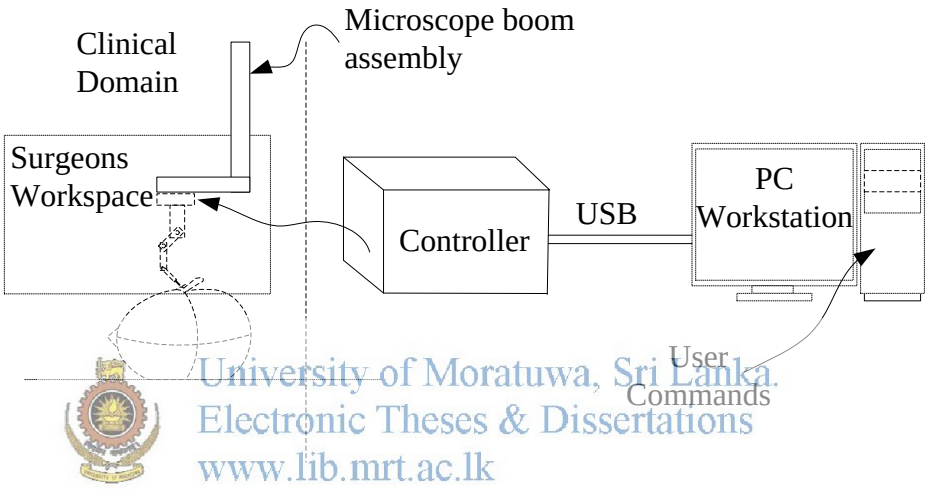

Figure 1.4: Conceptual diagram of the surgical manipulator tool

In this research, a new tool is proposed to ease the drilling and cleaning process in cochleostomy. The purposed tool: a robotic micromanipulator with six degrees of freedom is introduced into the conventional surgical procedure at the drilling scenario. The micro-manipulator is attached to a larger manipulator or positioning tool, for example: (preferably) the surgeons ENT microscope. Introducing a new tool requires a change in the conventional workflow. The said manipulator requires registration of the patient/surgical workspace in order to be used effectively for autonomous drilling.

For the task of patient workspace registration, the same computer tomography scan of the patient acquired preoperatively in a conventional cochlear implant surgery is also used for a 3-dimensional reconstruction of the anatomy. The anatomy model

created from the tomography is used for registration, surgical planning and intra-operative navigation.

Similar to conventional surgery, Surgeon uses unique landmarks and anatomical features of the anatomy, in order to localize. The identified landmarks marked on the three dimensional reconstruction, thereafter by moving the manipulator's end effector, the corresponding positions are marked on the patient physically. The software registers the workspace and correlates the virtual reconstruction of the surgical envelope to the physical surgical envelope. The robotic manipulator can now be positioned; the surgeon may either set the drilling path, or load a pre-operative clearing path plan before commencing the drilling. The system presented is semi-autonomous, thus the surgeon is able to alter or stop the manipulator during any drilling process.

The proposed tool is depicted in

Figure 1.4. Once the manipulator, attached to the surgical microscope is set in the vicinity of the workspace; localizing, and manipulation of the drill along a path can be done by giving user commands on the workstation PC. The surgeon can still observe the progress through the surgical microscope.

1.4 Thesis Overview

This thesis describes the process of developing a robotic surgical manipulator from concept to a simple prototype, as a possible solution to the limitations of conventional cochleostomy. The aim of this section is to give an outline of the thesis structure.

Chapter 1 begins with an introduction to the thesis. Chapter 1.1 describes the cochleostomy process, in which the research problem is specified. A literature review follows in Chapter 1.2, identifying presented solutions and their drawbacks. A solution to the research problem is proposed in Chapter 1.3

Chapter 2 describes the design of the robotic surgical manipulator. An important section of this is the concept development in section 2.1. The following sections focus on the theoretical design aspects of a robotic arm manipulator, in league with the concepts described in section 2.1. Section 2.8 looks at the dynamic effects of the robotic manipulator, the results of these critical in the development of the physical

prototype. The theoretical manipulator designed in chapter 2, are validated using the Robotics toolbox, and MATLAB.

In chapter 3, the localizing procedure for the surgical manipulator is described. The localizing component was an auxiliary requirement added to the research. Due to the fact that localizing is require to help validate the manipulator.

Chapter 4 focuses on controller design; a set of requirements for the manipulator controller is formulated and a simple controller design is presented that satisfies the formulated requirements.

Chapter 5 describes implementation of a prototype. The section describes the implementation of a simple prototype of the robotic manipulator, which satisfies the functionality of the manipulator design in concept. The major components of the chapter are: linkage, drive, electronics, software, and firmware design.

Chapter 6, describes the functionality of the robotic manipulator, an example use case is looked at, at the end of the chapter.

The final chapter, Chapter 7 concludes the work presented in the thesis.

2 DESIGN OF THE SURGICAL MANIPULATOR

2.1 Manipulator Design Considerations

By observation of the cochleostomy procedure at a cochlear implant surgery, and with recommendations of the surgeon, the basic characteristics and properties of the manipulator tool were speculated.

On workspace: The workspace that the end effector needs to maneuver along is constrained to a maximum envelope of 40[mm] x 40[mm] x 30[mm]. The tool would preferably operate only in the +z direction. (The system operates in inverted +z-axis.)

On usability: The tool should be of minimum hindrance to the surgeon, during the surgery. A tool could be of hindrance in a number of ways:

1) Obstruction of vision: due to the size and complexity and physical appearance of the tool, it could obstruct the view of the surgeon. The surgeon uses a microscope for detail viewing of the workspace. If there are multiple surgeons present, they too need to see. Depending on the physical appearance, the tool may be required to be completely out of the workspace for a clear view.
2) Limit access: due to the size and complexity, the tool could not allow any other tool to be used in the workspace. During a surgery, there may be many people present, with supporting hand tools and equipment may need to access the workspace for lubricating and cleaning blood, debris etc. In common practice, usually two surgeons operating on the patient and two other supporting personal for lubrication and cleaning.
3) Transportation and storage: due to its physical size and complexity, the tool requires special attention or process for moving to and from the workspace. This could mean waste of time, and may need support personal

The clearing procedures were done using abrasion, rather than by force. The drill burr is covered with a diamond dust coating for this purpose. The surgeon uses minimal force. The surgical tool is moved slowly about the area of cleaning, in circular motions. A speed of about 1~2 [$rads^{-1}$] was speculated.

On risk: The surgeon should be able to control the tool movement, and in an emergency, should be able to stop. Accidental contact with the tool should have

minimal impact. Care should be taken not to have the tool interfere with other tools and equipment electromagnetically.

2.2 Manipulator Concept Design

Selection of the manipulator configuration is the most critical part of the design, the work-envelope, limits, size, weight, accuracy etc. would depend directly on the configuration. According the criteria discussed in the previous section, the design was focused on down scaling of the physical size of the manipulator, with the object of achieving manipulability at the required work envelop of the operational point. In order to reduce the manipulator size: firstly, a kinematic configuration was adapted in which the weight distribution of drives, allow the links to be smaller: By using a light-weight robot configuration [10, 11] where three axis intersect at the base/shoulder of the manipulator, a shoulder manipulator in comparison to a wrist manipulator; the majority of actuators are to the base of the manipulator than at the end, thus reducing the size of the links at the end of the manipulator as the weight is distributed more towards the base. This also reduces the congestion caused by the manipulator joints at the end of the manipulator. The manipulator should not disrupt the vision of the surgeon and the others at the surgery.

Secondly, the physical size of joints and links were scaled down, while accommodating the drive actuators and sensors. As the sizes of the links were reduced and structure made compact, smaller actuator drives with the required torque were chosen. Reduction of size also reduces the inheriting weight of the manipulator, resulting in reduced inertia of the robot, this is advantages as it would cause less damage to the workspace if collision occurs by accident, reduction in inertia also simplifies the design of drive and control system.

Another advantage of the reduced inertia is that, the drives require less torque to move a joint. Thus the current carried for each electric motor in a joint is reduced. This enables an overall transmission of a small current to the manipulator, via short conductors, as the overall length of the manipulator is reduced. This means less electromagnetic interference. Reduction of electromagnet interference is critical as it may interfere with other medical devices and instruments in the surgical theater.

A drilling tool is attached to the 6th link. The end effector is transformed by $R_x(90)$. This tool replaces the high speed precision surgical hand drill (usually a Medtronic Skeeter ™ drill) used in conventional cochlear implant etc. surgery. The drilling tool uses burrs with diamond dust in order to drill through bone.

2.3 Definition of Manipulator Reference Frames

Figure 2.1, depicts the standard frames of reference for the manipulator.

The location of the tool is given by

$$ {}^{S}_{T}T = {}^{B}_{S}T^{-1}\,{}^{B}_{W}T\,{}^{W}_{T}T \qquad (1) $$

Where, ${}^{S}_{T}T$ is the tool frame $\{T\}$ relative to the station frame $\{S\}$. The station frame contains the workspace and all goal frames are taken relative to the station frame, this is given by the transform ${}^{S}_{G}T$. The frame $\{0\}$ of the robot is considered to be the base frame. All frames of the robot manipulator are taken with reference to the base frame. The origin of the last joint of the robot is considered to be the wrist frame $\{W\}$. Wrist frame is defined relative to the base frame. It is given by the transform ${}^{B}_{W}T$. Tool frame $\{T\}$ is a fixed transform applied to the wrist frame, given by the transform ${}^{W}_{T}T$.

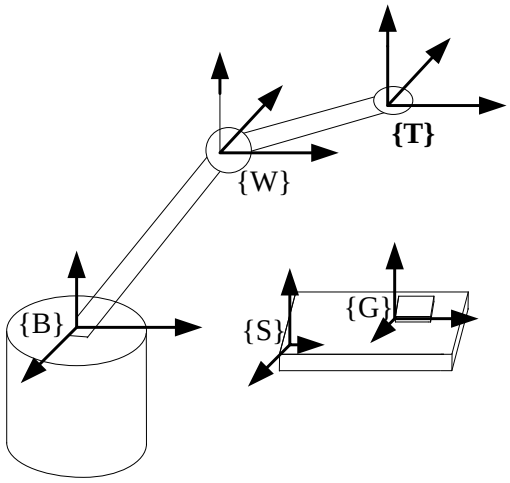

Figure 2.1: Diagram of manipulator frames of reference

2.4 Manipulator Kinematic Configuration

Solvability of the manipulator and ease of implementation was of principle concern in manipulator design. A fully actuated, 6 DOF serial link configuration was selected, considering the existence of a closed form solution in general. Presence of closed from solution for inverse kinematics is generally advantageous as joint angles could be found accurately with less computational effort in comparison to solving Jacobian inverse or other wrench based iterative numerical method for joint velocity which provide the nearest solution. At the same time, a 6 DOF configuration is advantages as the resultant Jacobian is of a square matrix form, which results in simplification of analysis for singularities, manipulability measures etc. Choosing a redundant kinematic configuration may prove advantageous in terms of dexterity, but would result in increase of complexity. Work envelope was chosen at a region with no inherited singularities, causing reduction of degrees of freedom.

The manipulator arm design consists of a shoulder (base), elbow and wrist. In order to limit the manipulator to a closed form solution, three joint axis at the shoulder were set to intersect[12]. An anthropomorphic characteristic is obtained by setting one link length from shoulder to elbow and one link offset from elbow to wrist. Drill tool attached to the wrist has dimensions analogues to a hand.

Joint offsets and link twists were selected to get the maximum workspace

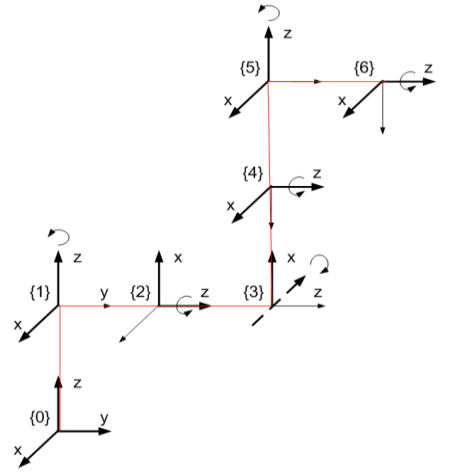

Figure 2.2: Kinematic configuration layout

envelope, simultaneously, maintaining offsets and twists at zero or $\pm\frac{\pi}{2}$, the link length and joint displacement offsets at zero, for kinematic simplicity. The resultant design of the kinematic chain parameters are given in the modified Denavit – Hartenberg (MDH) convention [13] in Table 2.1.

Table 2.1: Denavit - Hartenberg Parameters

Joint *i*	**Link Parameters**			
	α_{i-1}	a_{i-1}	d_i	θ_i
	[rad]	[mm]	[mm]	[rad]
1	0			θ_1
2	$-\pi/2$			$\theta_2 - \pi/2$
3	$\pi/2$			θ_3
4	$-\pi/2$	50		$\theta_4 + \pi/2$
5	$\pi/2$	35		θ_5
6	$-\pi/2$			θ_6

A tool transform is added to the last link, by $\frac{\pi}{2}$ rotation in the x direction, using an orthonormal matrix. Link offsets of $\pm\frac{\pi}{2}$ are applied to link 2 and link 4.

2.5 Manipulator Forward Kinematics

Kinematic equation for the manipulator was obtained using the MDH parameters are given in Table 2.1.

2.5.1 Derivation of Forward Kinematic Equations

The common kinematic shorthand notation: $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$;

Using the link transformations, in the notation of Craig[13].

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

Transformation matrix of operational point with respect to the base is given as:

$$ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T \tag{3} $$

$$ {}^{0}_{6}T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

With internal elements of the matrix:

$$ R_{11} = c_6\left(c_5\left(s_4\left(s_1s_3 - c_1c_3s_2\right) + c_1c_2c_4\right) - s_5\left(c_3s_1 + c_1s_2s_3\right)\right) $$
$$ + s_6\left(c_4\left(s_1s_3 - c_1c_3s_2\right) - c_1c_2s_4\right) $$
$$ R_{21} = -c_6\left(c_5\left(s_4\left(c_1s_3 + c_3s_1s_2\right) - c_2c_4s_1\right) - s_5\left(c_1c_3 - s_1s_2s_3\right)\right) $$
$$ - s_6\left(c_4\left(c_1s_3 + c_3s_1s_2\right) + c_2s_1s_4\right) $$
$$ R_{31} = s_6\left(s_2s_4 - c_2c_3c_4\right) - c_6\left(c_5\left(c_4s_2 + c_2c_3s_4\right) + c_2s_3s_5\right) $$
$$ R_{12} = -s_6\left(c_5\left(s_4\left(s_1s_3 - c_1c_3s_2\right) + c_1c_2c_4\right) - s_5\left(c_3s_1 + c_1s_2s_3\right)\right) $$
$$ + c_6\left(c_4\left(s_1s_3 - c_1c_3s_2\right) - c_1c_2s_4\right) $$
$$ R_{22} = s_6\left(c_5\left(s_4\left(c_1s_3 + c_3s_1s_2\right) - c_2c_4s_1\right) - s_5\left(c_1c_3 - s_1s_2s_3\right)\right) $$
$$ - c_6\left(c_4\left(c_1s_3 + c_3s_1s_2\right) + c_2s_1s_4\right) $$
$$ R_{32} = s_6\left(c_5\left(c_4s_2 + c_2c_3s_4\right) + c_2s_3s_5\right) + c_6\left(s_2s_4 - c_2c_3c_4\right) $$
$$ R_{13} = -s_5\left(s_4\left(s_1s_3 - c_1c_3s_2\right) + c_1c_2c_4\right) - c_5\left(c_3s_1 + c_1s_2s_3\right) $$
$$ R_{23} = s_5\left(s_4\left(c_1s_3 + c_3s_1s_2\right) - c_2c_4s_1\right) + c_5\left(c_1c_3 - s_1s_2s_3\right) $$
$$ R_{33} = s_5\left(c_4s_2 + c_2c_3s_4\right) - c_2c_5s_3 $$
$$ P_x = -a_3\left(s_1s_3 - c_1c_3s_2\right) - d_5\left(c_4\left(s_1s_3 - c_1c_3s_2\right) - c_1c_2s_4\right) $$
$$ P_y = a_3\left(c_1s_3 + c_3s_1s_2\right) + d_5\left(c_4\left(c_1s_3 + c_3s_1s_2\right) + c_2s_1s_4\right) $$
$$ P_z = a_3c_2c_3 - d_5\left(s_2s_4 - c_2c_3c_4\right) $$

The forward kinematics equations are utilized for a number of tasks: for workspace determination, Jacobian analysis and manual move of the end effector in the prototype.

2.6 Manipulator Inverse Kinematics

The inverse kinematic solution analytically in the closed form, is obtained using kinematic decoupling of position and orientation. Both algebraic and geometric properties of the model are observed. Using the geomantic properties of the first three and the last three sets of joints are analyzed independently.

2.6.1 Derivation of Inverse Kinematic Equations

According to Pieper [12], kinematics of position and orientation can be decoupled at the three axis intersection point. Using the methodology followed in [14], Goal position is given by ${}^{0}_{6}T$, position component of ${}^{0}_{6}T$ is dependent on ${}^{3}_{6}T$, angles $\theta_{4,5,6}$. The inverse of the ${}^{0}_{6}T$; ${}^{6}_{0}T$ can be written as a function of $\theta_{4,5,6}$ using forward kinematics

Goal position given by ${}^{6}_{0}P$ of ${}^{6}_{0}T$.

$$ {}^{6}_{0}P = \begin{bmatrix} Px & Py & Pz \end{bmatrix} \tag{5} $$

Using forward kinematics based on $\theta_{4,5,6}$ for the inverse position of ${}^{3}_{6}P$; ${}^{6}_{3}P$

$$ {}^{6}_{3}P = \begin{bmatrix} d_5 s_6 + a_3 c_4 s_6 + a_3 s_4 c_5 c_6 \\ d_5 c_6 + a_3 c_4 c_6 - a_3 c_5 s_4 s_6 \\ -a_3 s_4 s_5 \end{bmatrix} \tag{6} $$

By equating ${}^{6}_{0}P$ of goal and ${}^{6}_{3}P$ of the forward kinematics

$$ \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} = \begin{bmatrix} d_5 s_6 + a_3 c_4 s_6 + a_3 s_4 c_5 c_6 \\ d_5 c_6 + a_3 c_4 c_6 - a_3 c_5 s_4 s_6 \\ -a_3 s_4 s_5 \end{bmatrix} \tag{7} $$

Let $d_5 + a_3 c_4 = k_1$ and $a_3 s_4 = k_2$

$$ \begin{aligned} s_6 k_1 + c_5 c_6 k_2 &= Px \\ c_6 k_1 - c_5 s_6 k_2 &= Py \\ -k_2 s_5 &= Pz \end{aligned} \tag{8} $$

$$Px^2 + Py^2 + Pz^2 = k_1^2 + k_2^2 \tag{9}$$

let $p = Px^2 + Py^2 + Pz^2$

$$p = k_1^2 + k_2^2$$

$$p = \left(d_5 + a_3 c_4\right)^2 + \left(a_3 s_4\right)^2$$

$$c_4 = \frac{p - {d_5}^2 - {a_3}^2}{2 d_5 a_3} \tag{10}$$

$$\because \theta_4 = \pm \cos^{-1}\left(\frac{p - {d_5}^2 - {a_3}^2}{2 d_5 a_3}\right) \tag{11}$$

for $-1 \le \left(\frac{p - {d_5}^2 - {a_3}^2}{2 d_5 a_3}\right) \le 1$

But using c_4 and converting it to Atan2,

$$\theta_4 = 2\ \text{atan2}\left(\pm\frac{\sqrt{1 - {c_4}^2}}{1 + c_4}\right) \tag{12}$$

Instead of using sine or cosine terms, we make use of the two argument arctangent function *Atan2(y,x)*, which has a range of $-\pi < \theta < \pi$ for a given angle θ. This is a preferred method of inverse as it is computationally well behaved [15]

$$Pz = -a_3 s_4 s_5 \tag{13}$$

$$s_5 = \frac{Pz}{-a_3 s_4} \tag{14}$$

$$\theta_5 = \sin^{-1}\frac{Pz}{-a_3 s_4} \text{ and } \theta_5 = \pi - \sin^{-1}\frac{Pz}{-a_3 s_4} \tag{15}$$

Let $d_5 + a_3 c_4 = k_1$ and $a_3 s_4 c_5 = k_2$

$$Px = s_6 k_1 + c_6 k_2 \tag{16}$$

$$Py = c_6 k_1 - s_6 k_2 \tag{17}$$

$$Pxs_6 + Pyc_6 = s_6^2 k_1 + c_6 s_6 k_2 + s_6^2 k_1 - s_6 c_6 k_2$$

$$Pxs_6 + Pyc_6 = k_1 \tag{18}$$

Let $r \sin\phi = Py$ and $r \cos\phi = Px$

$$r = \pm\sqrt{Py^2 + Px^2} \text{ and } \phi = \tan^{-1}\left(\frac{Py}{Px}\right) \tag{19}$$

$$\cos\phi\sin\theta_6 + \sin\phi\cos\theta_6 = \frac{k_1}{r} \tag{20}$$

$$\sin\left(\phi + \theta_6\right) = \frac{k_1}{r}$$

$$\left(\phi + \theta_6\right) = \sin^{-1}\frac{k_1}{r}$$

$$\theta_6 = \sin^{-1}\frac{k_1}{r} - \phi$$

$$\theta_6 = \sin^{-1}\left(\frac{k_1}{\sqrt{Py^2 + Px^2}}\right) - \tan^{-1}\left(\frac{Py}{Px}\right)$$

$$\tan^{-1}\left(\frac{Py}{Px}\right) = A\tan 2\left(Px, Py\right) \tag{21}$$

$$\theta_6 = A\tan 2\left(Px, Py\right) - A\tan 2\left(k_2, k_1\right)$$

With ${}^6_3R$ written as a function of $\theta_{4,5,6}$;

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \left({}^0_6R\,{}^6_3R\right)^{-1} \tag{22}$$

Using forward kinematics for ${}^3_0R$;

$${}^3_0R = \begin{bmatrix} c_1s_2c_3 - s_1s_3 & s_1s_2c_3 + c_1s_3 & c_2c_3 \\ -c_1s_2s_3 - s_1c_3 & -s_1s_2s_3 + c_1c_3 & -c_2s_3 \\ -c_1c_2 & -s_1c_2 & s_2 \end{bmatrix} \tag{23}$$

By equating(22) and (23);

First three:

$$\theta_2 = atan2\left(R_{33}, \pm\sqrt{R_{31}^2 + R_{32}^2}\right) \tag{24}$$

$$\theta_3 = atan2\left(\frac{R_{23}}{-c_2}, \frac{R_{13}}{c_2}\right) \tag{25}$$

$$\theta_1 = atan2\left(\frac{R_{32}}{-c_2}, \frac{R_{31}}{-c_2}\right) \tag{26}$$

The inverse kinematics derivation, results in eight sets of solution for the robotic manipulator. All solutions are not used at operation; Best configurations are selected, based on the workspace, and joint limits.

2.7 Manipulator Singularities

Dynamic Analysis of kinematic singularities of the robotic manipulator is essential in order to ensure reachability of the end effector. Two forms of singularities are determined. Boundary singularities are caused by maximum stretching or retraction of a link chain and internal singularities, caused by alignment of two or more axis of motion or due to particular end-effector configurations.

For the computation of internal singularities, singularity decoupling is used to break the manipulator into two problems. In the case of a 6 DOF manipulator, the Jacobian is partitioned into (3×3) blocks;

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{13} & J_{22} \end{bmatrix} \tag{27}$$

Singularities are typical of the mechanical structure and do not depend on frame chosen to describe kinematics. Choosing origin as the intersection the 3-axis leads to:

$$J_{12} = [0 \quad 0 \quad 0] \tag{28}$$

$$J = \begin{bmatrix} J_{11} & 0 \\ J_{13} & J_{22} \end{bmatrix} \tag{29}$$

The determinant is taken from the product of the diagonal element:

$$\det(J) = \det(J_{11})\det(J_{22}) \tag{30}$$

Where, $\det(J_{11})$ are the lower 3 intersecting joints, and $\det(J_{22})$ is the upper three intersecting joints.

$$q_2 = \frac{\pi}{2}\ ,\ q_3 = \frac{\pi}{2} \tag{31}$$

$$q_4 = \pi \tag{32}$$

$$q_2 = \frac{\pi}{2}\ ,\ q_6 = \pi \tag{33}$$

$$q_5 = \frac{\pi}{2} \ , \ q_6 = \pi \tag{34}$$

Based on this result a value of [15, 30, 0, 120 0, 60] [Degrees] was chosen to be the ready pose.

2.8 Manipulator Dynamics Analysis

2.8.1 Link Mass Distribution

The distribution of mass of a rigid body in relation to a reference frame is given by the mass moments of inertia.

An inertia tensor is defined as

$$^{A}I = \begin{bmatrix} I_{xx} & -I_{-xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{35}$$

Elements I_{xx}, I_{yy}, I_{zz} are the mass moments of inertia and the other elements are mass products of inertia.

Link mass and center of mass properties given in Table 2.2 and inertia tensor properties given in Table 2.3 for each link is found using, the manipulators CAD model using the Solidworks® software

Table 2.2: Table of Link Mass Properties

Link	**Mass [g]**	**Center of Mass [mm]**		
		x	***y***	***z***
1	12.85	0	34.3	11.27
2	12.1	0	-22.61	-12.45
3	11.6	31.84	-26.39	-12.33
4	10.55	4.66	-3.21	7.98
5	11.72	0	17.1	12.21
6	10	10	0	-6

With the mass distribution, and kinematic information for each link, forces and moments for each joint and the manipulator can be found using the dynamic equations of motion. The calculation can be done either by using closed form

derivation of dynamic equations, or by using a numerical method. In this study, iterative Newton – Euler method is used for dynamic simulation.

Table 2.3: Table of Link Inertia Properties

Link	**Moments of Inertia [g.mm^2]**								
	Lxx	*Lxy*	*Lxz*	*Lyx*	***Lyy***	*Lyz*	*Lzx*	*Lzy*	***Lzz***
1	2603.49	0	0	0	904.06	0	0	0	1954.09
2	1661.78	0	0	0	934.98	0	0	0	964.49
3	1571.57	-542.1	-334.1	-542.1	1455.68	161.59	-334.1	161.59	1739.63
4	627.83	44.77	-1.85	44.77	550.66	-17.65	-1.85	-17.65	913.65
5	1392.34	0	0	0	774.45	72.46	0	72.46	849.91
6	235.01	0	0	0	1423.1	0	0	0	1423.1

2.8.2 Dynamic Equations of Motion

The dynamic equation for a series of links is given by the coupled differential equation.

For a series of rigid body link, the dynamic equations takes the matrix form:

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^{T} g \quad (36)$$

Where, $q, \dot{q}$ and $\ddot{q}$ are joint angle, velocity and acceleration. Q is the generalized joint forces and moments. M is the joint space inertia. C is the Coriolis and centripetal coupling. F is the friction force. G is the gravity loading. g is a wrench applied at the end-effector and J is the manipulator Jacobian. [13], [16], [17]

2.8.3 Analysis of Gravity Loading on Manipulator

The most dominant loading effect on the surgical robotic manipulator is the gravity term. The effect of gravity can be looked at by solving the dynamic equation setting acceleration and velocity to zero, as the gravity term is only dependent on joint angle, the torque inserted on a joint due to gravity depends on the robot's pose.

The second link has the highest gravity torque, in the manipulator configuration. The variation of gravity torques for the joints 2 and 5 are plotted in Figure 2.3, for the

angles from $\pm 2\frac{\pi}{3}$. The plot shows a maximum torque of 33.9[mN.m.] For joint 2 as the arm is reached out horizontally. Joint 5 has a maximum torque of 0.88 [mN.m] as this information is useful for determining the required torque for the motors.

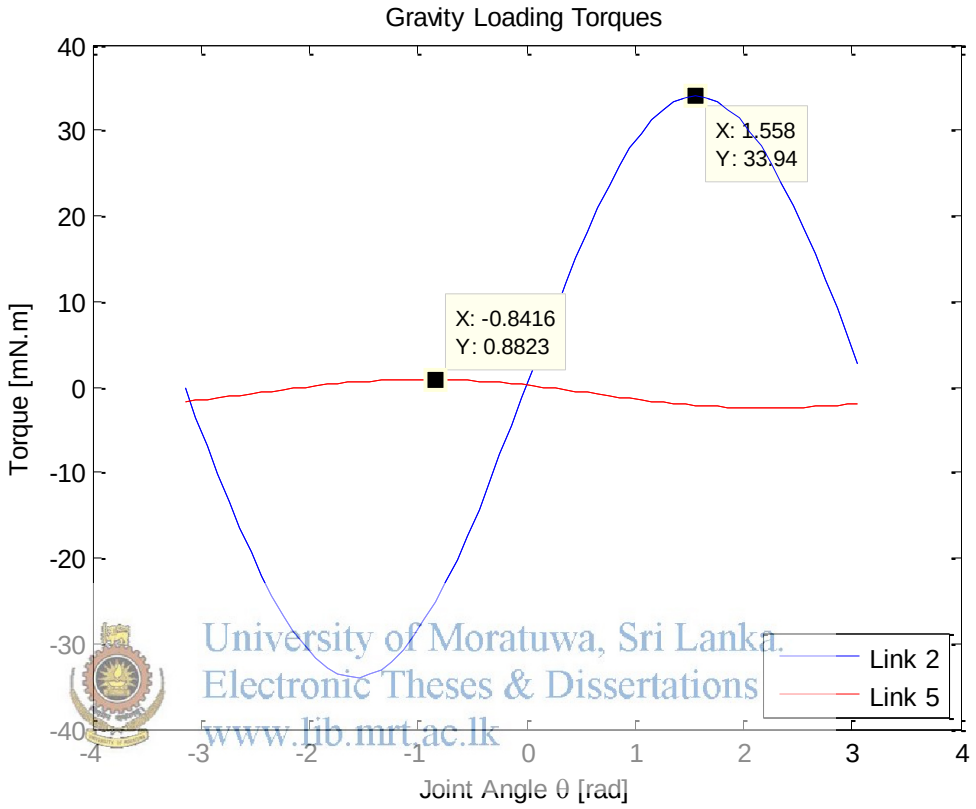

Figure 2.3: Plot of gravity loading torques

2.8.4 Analysis of Inertia Loading on Manipulator

In the robotic surgical manipulator, for a fixed maximum motor torque, inertia sets the limits of acceleration. The amount of inertia variation is important in the manipulator drive design, as the gear ratio of the drive has to be set such that it reduces the overall inertia seen by the motor. The link size and mass accumulates as it reaches the base, thus the lower portion of the joints tend to have a higher moment of inertia. Inertia variations for joint 1 and 2 as functions of joint angle 2 and 3 are plotted in Figure 2.4. According to the plot, Inertia variation for joint 1 is 3.12×10^{5} $[\,g.mm^{2}\,]$ and variation factor is 1.9860

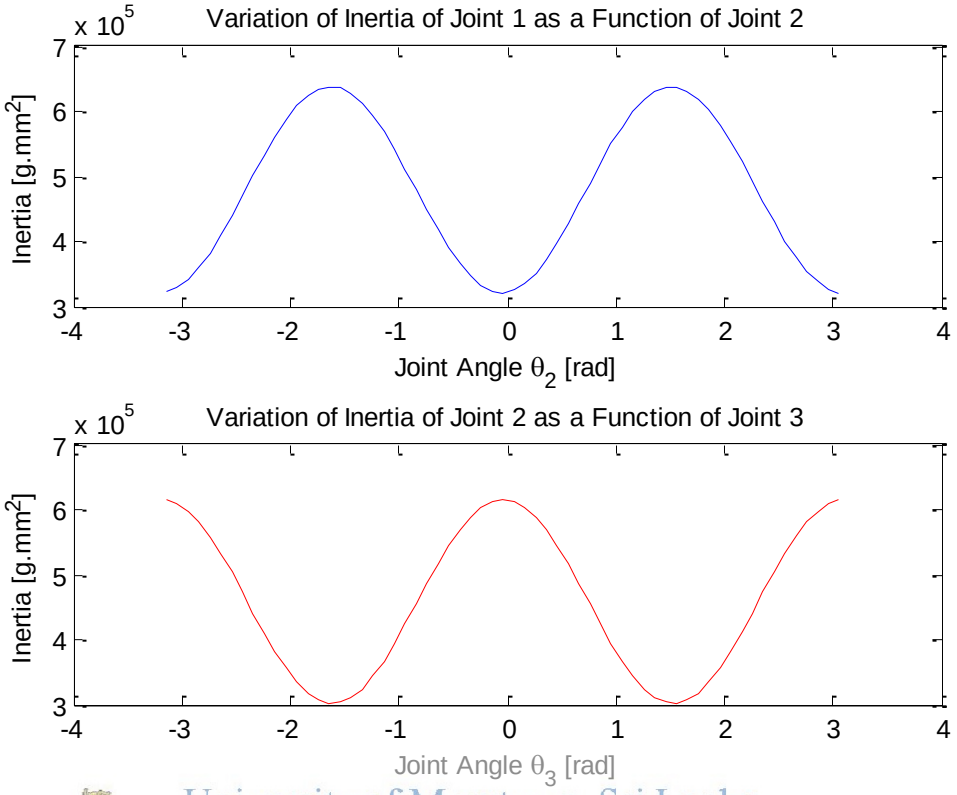

Figure 2.4: Plot of Inertia loading

2.8.5 Analysis of Payload Loading Torques

The payload attached to the robotic manipulator is a surgical micro-drill, with a fixed maximum mass of 30 [g]. The payload at the end of the manipulator chain will cause the joints to see an increase in inertia. Added mass also means that the drives must have the capacity to support, without exceeding the torque rating. The effect of payload on the inertia variation is plotted in Figure 2.5

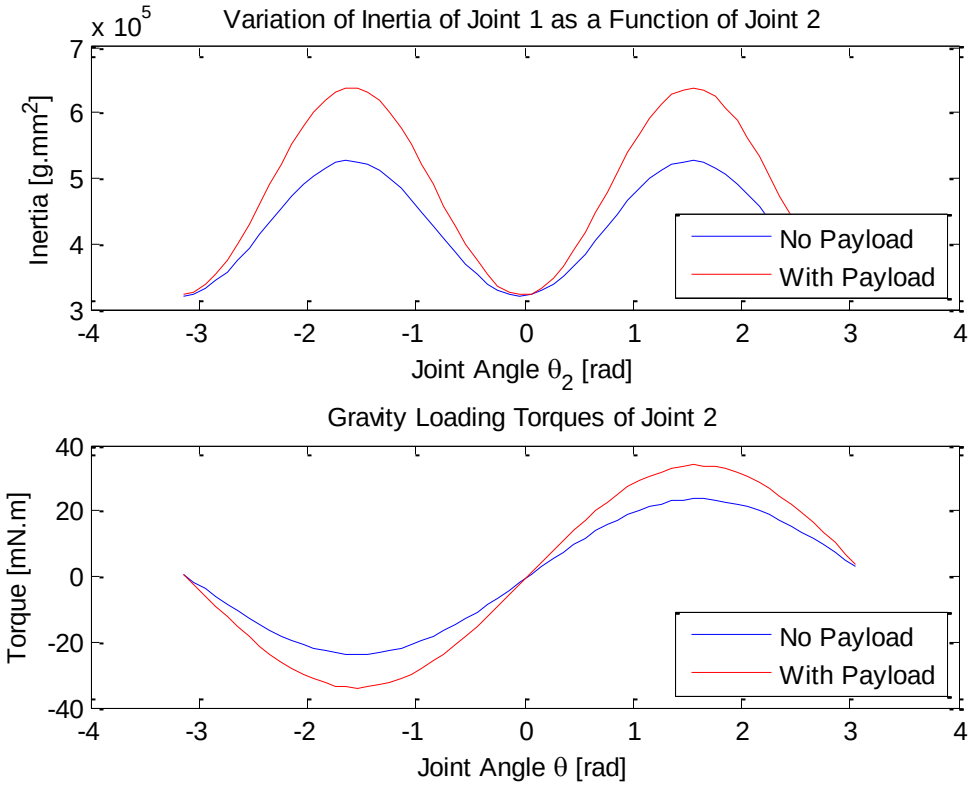

Figure 2.5: Inertia loading with payload

Choosing a light-weight configuration is beneficial, with regard to gravity component and the inertia component; in a lightweight configuration, the distribution of mass is more to the base of the manipulator, these results in less power requirement in motors, which results in smaller motor drives. This also mitigates decoupling non-linearity effects.

2.8.6 Analysis of Frictional Forces

Friction is a significant component in a typical robot manipulator, as it has high gear ratios, friction is mainly of two components: Viscos friction and Coulomb friction. Viscous friction is a function of velocity and Coulomb friction is a constant. The friction parameters for the drive were measured by experiment, and are given in Table 5.1. The friction nonlinearities for the linkages are not considered in the analysis of this study, as it is difficult to obtain link friction parameters.

2.9 Motion Trajectory Generation

In the robotic surgical manipulator, a joint space schema of trajectory generation is implemented. Given starting and end points, the trajectory generator generates $\theta, \dot{\theta}$ and $\ddot{\theta}$. additionally, via-points can be specified between starting and end points. A cubic function is used for interpolating between points. Cubic function has four coefficients, that is able to accommodate four constrains on joint angle θ, these being initial and final angles, and initial and final velocity.

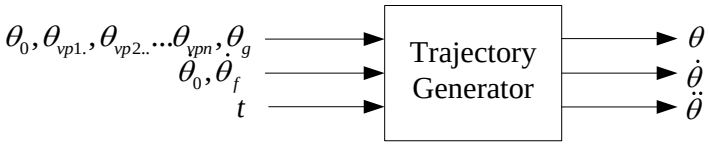

Figure 2.6: Diagram of the trajectory generator

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{37}$$

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{38}$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \tag{39}$$

$$a_0 = \theta_0 \tag{40}$$

$$a_1 = \dot{\theta}_0 \tag{41}$$

$$a_2 = \frac{3}{t_f^2}\left(\theta_f - \theta_0\right) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f \tag{42}$$

$$a_3 = -\frac{2}{t_f^3}\left(\theta_f - \theta_0\right) - \frac{1}{t_f^2}\left(\dot{\theta}_f - \dot{\theta}_0\right) \tag{43}$$

3 PATIENT WORKSPACE REGISTRATION

For the purpose of patient registration, a workflow is purposed, in which the manipulator is moved manually into positions of identified landmarks

The objective of workspace registration is to correlate the coordinate systems of the physical workspace to a modeled virtual workspace, using anatomical landmarks or artificial landmarks. This is achieved using the Iterative Center Point algorithm as described in references [18],[19], where a transformation, is found using the algorithm, that which describes the relationship between the two coordinate frames

3.1 Iterative Closest Point Algorithm Description

The objective of the algorithm is to find the optimal rotation and translation rule, between two data sets. Using the corresponding two datasets of the model P_m and target P_t, assuming minimal variation between datasets and optimal in terms of least square error

Let P_m and P_t be model and target position datasets

Where P_x is the Cartesian position vector $[x, y, z]^T$

Let CP_m and CP_t be the centroids of each point cloud

$$CP_x = \frac{1}{n}\sum_{i=1}^{n} P^i{}_x \tag{44}$$

Change origin of the datasets, removing translation, in order to isolate orientation

Let H be a covariance matrix

$$H = \sum_{i=1}^{n}\left(P^i{}_m - CP_m\right)\left(P^i{}_t - CP_t\right)^T \tag{45}$$

Using singular value decomposition

$$[U, D, V] = SVD(H) \tag{46}$$

Rotation matrix is calculated as

$$R = VU^T \tag{47}$$

Translation component is calculated as

$$P = \left[PC_t - RPC_m\right]^T \tag{48}$$

The correlation between the coordinate systems of dataset P_m and P_t are

$$P_t = RP_m + P \tag{49}$$

To further create a transformation matrix

$$T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{50}$$

Thus transformation between datasets m and t can be done using

$$t_i^T = Tm_i^T \tag{51}$$

3.2 Patient Workspace Registration Process

For the process of patient registration, once the manipulator is in the vicinity of the patient workspace, the manipulator end effector needs to be moved manually to each of the prior identified landmark positions, which are the P_t^i target landmarks in the point cloud dataset. For the task of end effector movement either forward kinematics or inverse kinematics can be used. Once the manipulator end effector is moved into position, transformation matrices are obtained from forward kinematics. This process is recorded by the software (presented in Chapter 5.6), by using the algorithm presented in Chapter 3.1, the software obtains the transformation T_t^i from the forward kinematics of the manipulator arm. The resultant transformation matrix is used for positioning for transferring position information from the model to the physical workspace

4 CONTROLLER DESIGN

The basic purpose of a robot controller is to make the end-effector follow a desired trajectory. The end-effector Cartesian path trajectory is allowed by setting manipulators joints to follow a specific joint space trajectory.

The inputs for the controller are from a trajectory generator, such as the one looked at in chapter 2.9. The trajectory generator computes position, velocity and acceleration states for a given time period. The output of the controller is to the plant; in this case the plant is a DC motor. The input to the dc motor is in the form of voltage. The voltage control signal manipulates a power amplifier using pulsed width modulation (PWM) scheme which proportionally provides power to the motor, by varying the duty cycle of PWM.

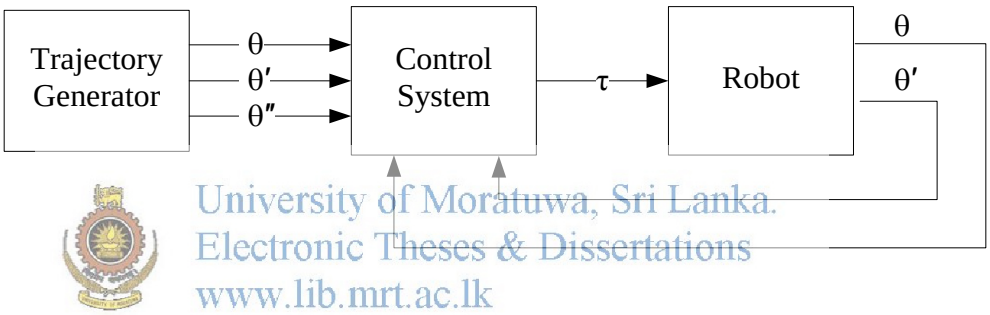

Figure 4.1: Diagram of the control system outline

4.1 Controller Design Requirements

The purpose of this section is to derive some initial requirements for the controllers. These requirements are based on the information gained on the analysis and are only preliminary.

It is important that the position controller has the capability to hold given position in the steady state. Thus for accurate steady state the steady state error should be small as possible. (Approximately ≈ 0). The velocity controller should also have a small steady state values as possible. In terms of overshoot, it is acceptable for the velocity to overshoot, as long as the settling times are well below the required.

The maximum speed for the drive is less approximately 8 [rad/s]. The maximum displacement for the each joint angle is $\pi/2$ [rad]. The encoder maximum resolution is 0.05 [degrees]. Assuming the encoder is noise free, and then the minimal joint step size is approximately 8.7×10^{-4} [rad]. If the drive is commanded to reach the minimum step size at maximum velocity 8 [rad/s] the time taken would be approximately 109 [μs]

In order to determine numerical values, the joint controller microprocessor was looked at. The joint controller has a microprocessor loop latency dictates the finale sample period. A maximum period of 600~800 [us] latency is caused by the main loop of the microprocessor. Thus the sampling period is taken as worse case value of 800[us] or 1.25[kHz]. It is recommended to have at least 6 samples before rise time period.

For the velocity controller, which is in the inner loop of the cascaded controller, the sampling period is taken as the loop latency time.

$$T_{sv} = 800[\mu s] \tag{52}$$

The rise time is six times of the sampling period:

$$t_r = 6 \cdot T_{sv} = 4.8[ms] \tag{53}$$

And the rise time is defined as 2% of the settling time

$$t_s = \frac{t_r}{0.2} = 24.0[ms] \tag{54}$$

Natural frequency ω_n is approximated as:

$$\omega_{n(outer)} = \frac{1.8}{t_r} = 375[rad/s] \tag{55}$$

With overshoot of:

$$M_p = 0.2 \tag{56}$$

The overshoot defines the damping ratio:

$$\zeta = \zeta(M_p) = 0.457 \tag{57}$$

And decay defined by:

$$\sigma = \frac{\ln(1\%)}{t_s} = 192 \tag{58}$$

For the position controller, the outer loop of the cascaded controller, the bandwidth $\approx \omega_n$ needs to be smaller by a factor between 4~10.

A gain factor of 4 was selected. Thus the bandwidth for the outer loop is approximated as approximated as $\frac{1}{4}\omega_{n(inner)}$ of inner loop.

$$\omega_{n(outer)} = \frac{375}{4} = 93.75[rad/s] \tag{59}$$

Thus, rise time is taken as:

$$t_r = \frac{1.8}{\omega_n} = 19.2[ms] \tag{60}$$

With a settling time of:

$$t_s = \frac{t_r}{0.2} = 96.0[\text{ms}] \tag{61}$$

Since no overshoot the damping ratio was taken as 1. Thus:

$$\zeta = 1 \tag{62}$$

With a decay of

$$\sigma = \frac{\ln(1\%)}{t_s} = 47.9 \tag{63}$$

These requirements are to be met in the controller design.

4.1.1 Manipulator Drive Model

In order to create a controller with acceptable performance, the primary task is to model the drive. The drive for the robotic actuator is a geared PMDC motor, with an angle position sensor for feedback.

4.1.1.1 DC Motor Model

Electric dynamics of the system defined as:

$$v_a(t) = R_a \cdot i_a(t) + L_a \frac{di_a(t)}{dt} + v_b \tag{64}$$

Where v_a, R_a, i_a, L_a and v_b are armature voltage, resistance, current, inductance and back-electromotive force (back-EMF).

Mechanical dynamics of the motor are defined as:

$$J \cdot \frac{d\omega_m(t)}{dt} = \tau_m(t) - B\omega_m(t) \quad (65)$$

Where J, B, τ_m and ω_m are the moment of inertia of the motor, friction coefficient, motor torque and motor speed.

Electro mechanical coupling is defined as:

$$\tau_m = k_t \cdot i_a(t) \quad (66)$$

$$v_b(t) = k_v \omega_m(t) \quad (67)$$

Where k_t is the motor constant and k_v is the back-EMF constant. Numerically, $k_t = k_v$ under SI units.

Let

$$k_m = k_t = k_v \quad (68)$$

Considering armature voltage and velocity of the motor; using Laplace transforms:

$$V_a(s) = R_a I_a(s) + sL_a I_a(s) + V_b(s) \quad (69)$$

$$sJ\Omega_m(s) = C_m(s) - B\Omega_m(s) \quad (70)$$

$$C_m(s) = k_m I_a(s) \quad (71)$$

$$V_b(s) = k_m \Omega_m(s) \quad (72)$$

The electrical and mechanical dynamics are expressed as two transfer functions:

$$I_a(s) = \frac{V_a(s) - V_b(s)}{R_a + s \cdot L_a} \quad (73)$$

$$\frac{\Omega_m(s)}{C(s)} = \frac{1}{B + s \cdot J} \quad (74)$$

Combining both transfer functions:

$$\Omega_m(s) = \frac{k_m}{(B + sJ)(R_a + sL_a) + {k_m}^2} \cdot V_a(s) \quad (75)$$

$$G_\omega(s) = \frac{\Omega_m(s)}{V_a(s)} = \frac{k_m}{(L_a J)s^2 + (R_a J + BL_a)s + (BR_a + {k_m}^2)} \quad (76)$$

4.1.1.2 Driveline Model

Using the energy conservation laws, in an ideal gearbox, power at the input of the gearbox equals power at the output.

$$\tau_m(t)\omega_m(t) = \tau_c(t)\omega_c(t) \quad \text{with } (\omega_c < \omega_m) \tag{77}$$

$$\omega_c = \frac{\omega_m}{K_r}, \qquad \tau_c = K_r\tau_m \tag{78}$$

At the load side, the mechanical system is described by:

$$sJ_c\omega_c(s) = \tau_c(s) - B_c\omega_c(s) \tag{79}$$

At the motor side,

$$\tau_{c,m} = \frac{\tau_c}{K_r} = \frac{1}{K_r^2}(B_c + sJ_c)\omega_m(s) \tag{80}$$

Considering motor and the load together,

$$\tau(s) = \tau_m(s) + \tau_{c,m}(s) \tag{81}$$

$$\tau(s) = B_t\omega_m(s) + s \cdot J_t\omega_m(s) \tag{82}$$

Thus,

$$\omega_m(s) = \frac{\tau(s)}{B_t + sJ_t} \tag{83}$$

Where,

$$B_t = B + \frac{B_c}{K_r^2}, \text{ and } J_t = J + \frac{J_c}{K_r^2}$$

Thus the complete model of the drive is given by:

$$G_\omega(s) = \frac{\Omega_d(s)}{V_a(s)} = \frac{\frac{k_m}{K_r}}{(L_aJ)s^2 + (R_aJ + BL_a)s + (BR_a + {k_m}^2)} \tag{84}$$

The use of very high gear ratios tend to linearize the nonlinear behavior of the drive system

$G_\omega(s)$ can be expressed as a second order equation in canonical form, with $K_g = \frac{1}{K_r}$

$$G_{\omega}(s)=\frac{\Omega_d(s)}{V_a(s)}=\frac{\dfrac{k_m K_g}{(L_a J)}}{s^2+\left(\dfrac{R_a}{L_a}+\dfrac{B}{J}\right)s+\dfrac{\left(BR_a+{k_m}^2\right)}{(L_a J)}} \tag{85}$$

4.1.2 Power Amplifier Model

The servo system contains a power amplifier, which does the task of modulating the control signal using pulsed width modulation. The amplifier takes a fraction of the power available at the supply, proportional to the control signal. The amplifier gain is given by k_a

The input to the controller is modeled as:

$$v_a = v_{in} \cdot k_a \tag{86}$$

The complete motor model is given in Figure 4.2., which includes an integrator after the velocity term for derivation of position.

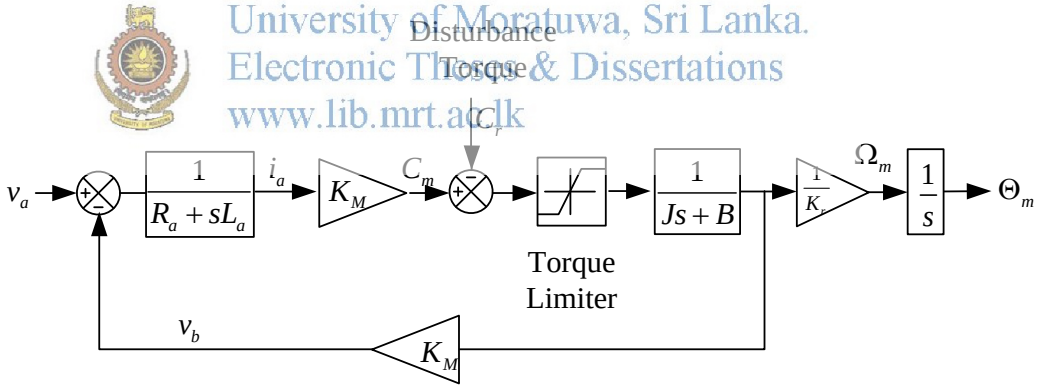

Figure 4.2: Diagram of the drive model

4.1.3 Open Loop Plant Analysis

Using the largest link for its mean inertia,

$$J_t = J + \frac{M_{22}}{{K_r}^2}$$

Where M_{22} is the inertia matrix of a joint of the manipulator.

Using Table 5.1, for the drive parameters;

Motor inertia $J : 25\times10^{-9} kg\cdot m^2$

Gear ratio: $K_r : 214.28$

Largest link mean inertia of link 2, M_{22} $2.3918\times10^{-3} kg\cdot m^2$

From our previous analysis of system dynamics, the mean inertia of link 2 inertia to be $2.3918\times10^{-3} kg\cdot m^2$;

$$J_t = 25\times10^{-9} + \frac{2.3918\times10^{-3}}{214.28^2} = \ 25\times10^{-9} + 52.091\times10^{-9} = 77.09\times10^{-9} kg\cdot m^2$$

The numerical values for the combined transfer function is G_ω

$$G_\omega = \frac{9.204\times10^6}{s^2 + 7.266\times10^5 s + 4.768\times10^6} \tag{87}$$

The open loop transfer function shows the characteristics of a damped second order function. A unit step input to the transfer function gives a rise time of 0.338[s] a settling time of 0.6 [s] and a steady state value of 1.93[rad/s] for an input of 1[v]. Inertia is the major factor that increases or decreases the settling time.

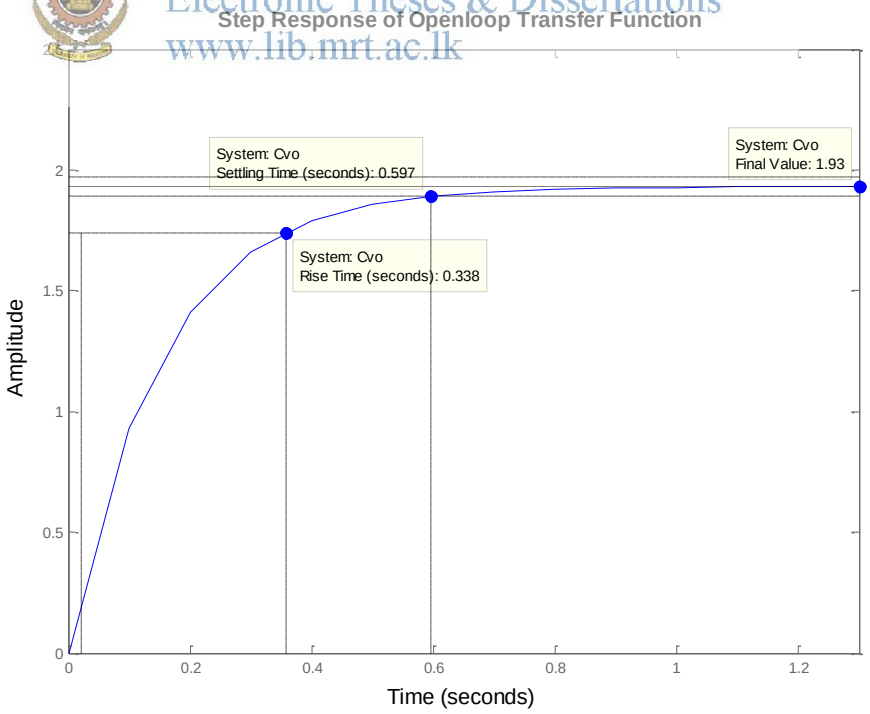

Figure 4.3: Open loop step response of drive

Measurement data for the drive are plotted in Figure 4.4. The figure shows output of the system for a voltage input of 5[v]. The signals in solid lines depict the system response taken at data acquisition sampling intervals from 500~900[μs]. The dotted line shows the modeled system in comparison. The modeled system is not ideal, but assumed to be close in approximation to the physical system. It is also noteworthy that the notches, seen at the beginning of the solid lines are due to friction (stiction). These aspects are not seen in the simulated model as nonlinearities such as dry-friction and stiction are not accommodated in the model.

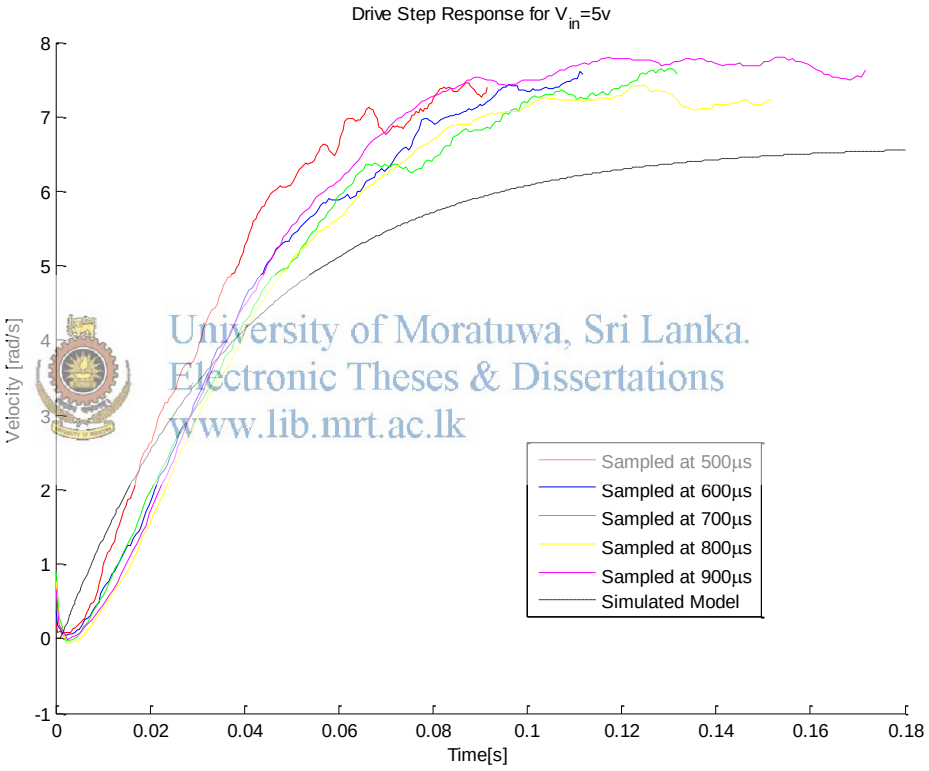

Figure 4.4: Drive step responses

Theoretically, the system poles are given by:

$$p_{1,2} = \frac{-R_a J \pm \sqrt{(R_a J)^2 - 4L_a J k_m^{\ 2}}}{2L_a J} \tag{88}$$

Since L_a is sufficiently small such that $(R_a J)^2 - 4L_a J k_m{}^2 > 0$ then, the two poles are real and negative.

Thus, $L_a \ll \frac{R_a{}^2 J}{k_m{}^2}$ then the electro mechanical pole $p_1 \approx -\frac{k_m{}^2}{R_a J}$ and electrical pole $p_2 \approx -\frac{R_a}{L_a}$

$$G_\omega(s) = \frac{\frac{k_m}{L_a J}}{\left(s + \frac{k_m{}^2}{R_a J}\right)\left(s + \frac{R_a}{L_a}\right)} = \frac{1/k_m}{(1+s\tau_m)(1+s\tau_e)} \tag{89}$$

Electro mechanical and electrical poles given as:

$$\tau_m = \frac{R_a J}{k_m{}^2} \approx \frac{J}{B} \tag{90}$$

$$\tau_e = \frac{L_a}{R_a} \tag{91}$$

The resulting τ_m and τ_e are given as: $\tau_m = 2.6955$, and $\tau_e = 1.3763\text{e-}006$. The resulting τ_m is much larger than τ_e.

4.2 Joint Space Control Design

There are two main approaches to joint space control. These being: independent joint control and model-based control. Independent or decentralized joint control is a simple form of joint space control, in which each joint in a n joint robotic manipulator is treated as an independent control system of the single-input-single-output form. Each of the joints independently (decoupled from other joints) follows their own joint angle trajectory, contributing to the Cartesian trajectory path of the end-effector. Coupling effects internal to the system and forces external to the system (gravity, velocity, acceleration, coupling forces, friction) are treated as external disturbance inputs.

A simple implementation of independent joint control is in the form of a nested control loop. In which an inner loop maintains the velocity of each joint as demanded by an outer loop, which in turn maintains position and determines velocity of each

joint as to minimize position error. In order to increase accuracy of the controller, in superior control schemas, a form of compensation is achieved by feeding in recalculated dynamics in terms of torque, such as by computing model dynamics using recursive Norton – Euler algorithm. An example system is shown in Figure 4.5. In this research dynamic compensation is not implemented.

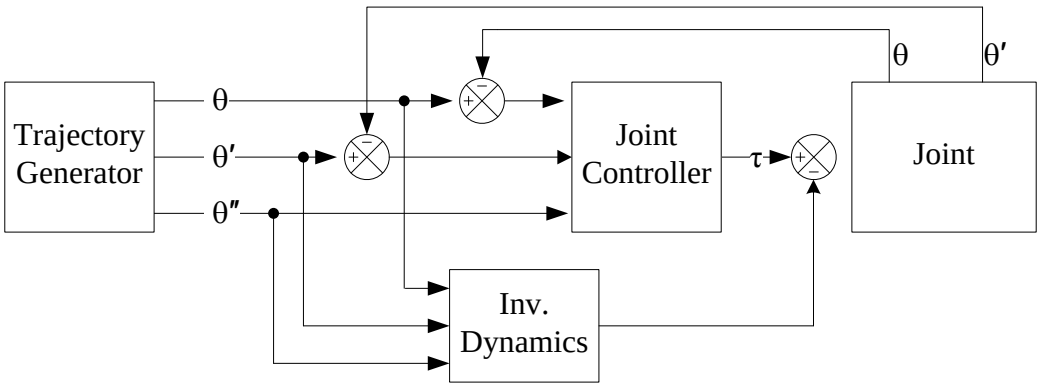

Figure 4.5: Control system with dynamic compensation

4.2.1 Cascaded Controller Design

A cascaded position and velocity controller was designed using root locus design methods, with the help of MATLAB Control Toolbox *sisotool*. The velocity and position models are created as two loops. Where velocity model P_1 is in the inner loop and position model P_2 is in the outer loop. Two controllers are created for the inner loop C_2 and the outer loop C_1. The inner loop consists of a Lag compensator and the outer loop consists of a proportional controller.

4.2.1.1 Design of Inner Control Loop

The inner loop controls the velocity of a joint. The reference velocity is set by the demand of the outer loop. The system is responses settling times; heavily depend on motor parameters and the link inertia. (τ_m is much larger than τ_e)

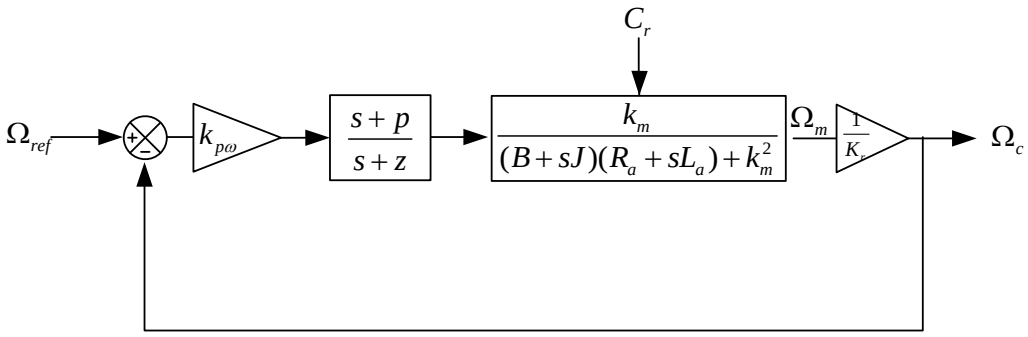

Figure 4.6: Diagram of the velocity control loop

The openloop equation of the drive:

$$G_\omega = \frac{9.204\times10^6}{s^2 + 7.266\times10^5 s + 4.768\times10^6} \tag{92}$$

System contains two negative poles at -7.26×10^5 , and -4.67, thus contains no oscillations or overshoots in the step response. Since, one pole is much negative in magnitude than the other, the slower pole at -4.67 will determine the dynamics of the motor speed.

By observation of pole zero positions on a pole zero plot, the open loop transfer function contains two poles at -7.26×10^5 and -6.5624, one pole is greater in magnitude than the other. The slower pole dictates the system dynamics, while the higher frequency pole has minimal impact on the system dynamics. Thus the system can be reduced to a first order approximation to simplify the design.

The transfer high frequency pole is removed while keeping the dc gain fixed. This gives the first order approximation

$$G_\omega = \frac{12.67}{s + 6.562} \tag{93}$$

The approximation can be verified using a bode plot:

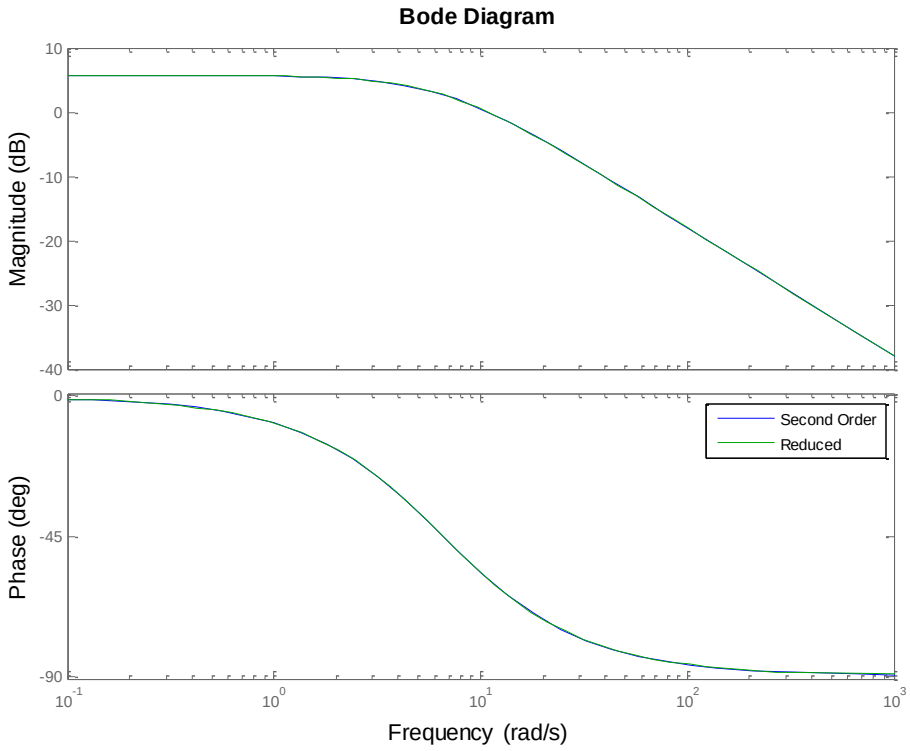

Figure 4.7: Bode plot of original and reduced transfer functions

Using the design requirements described in section 4.1 for the velocity control loop, which are:

$t_s = 24.0[ms]$, $t_r(2\%) = 4.8[ms]$, $M_p < 20\%$ which gives $\zeta = .46$, $\omega_n = 375$, decay of $\sigma = 192$ and zero steady state error. $0 \approx Er_{ss}$

Using the root locus methods, initially a proportional controller was tested, with the requirements constrained in the root locus plot.

The shaded area in the plot marks constrains. The pole must be located outside it. Initially, the gain is increased as to take the poll outside, but this does not fulfill the requirements. Thus the gain is increased till a gain of 35.9. This gives a settling time of 8.5[ms], and a rise time of 4.7[ms], satisfying the rise time and the settling time constrains. But it does not fulfill the zero steady state error, these is a steady state error of 0.02[rad/s]

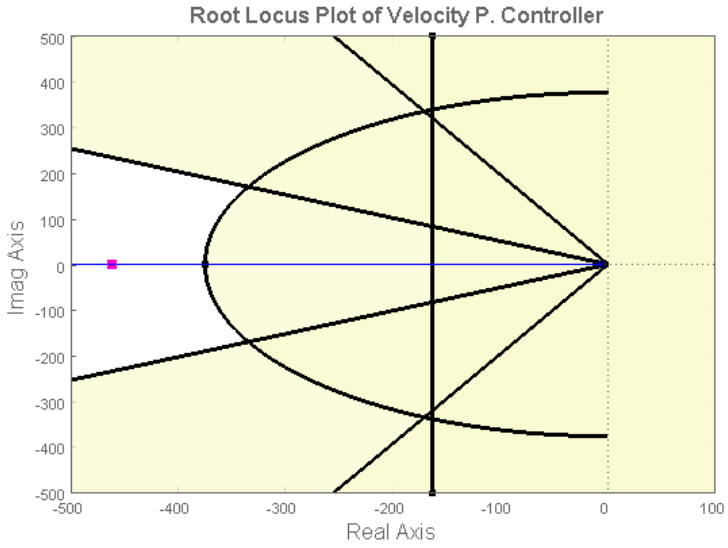

Figure 4.8: Root locus plot of proportional velocity controller

In order to achieve the steady state requirement, a lag compensator was introduced to the system. A lag compensator adds a pole and a zero to the dynamic system in order to reduce the steady state error. Since the objective is to zero the steady state error, ideally a pole and a zero are to be placed at 0. but this would result in a rise time of 5.2[ms] and a settling time of 11.5[ms], failing to satisfy the rise time constraint.

The lag controller is defined by:

$$C_{Lag} = \frac{s+z}{s+p} \text{ with } z \geq p \text{ and } p, z \ll \omega_n \tag{94}$$

Where p, z are pole and zero, of the lag compensator.

Once again using root locus methods, the ideal pole zero locations and the gain was found to be:

$$C_1 = G_1 \cdot C_{Lag} \tag{95}$$

$$C_1 = 110 \cdot \frac{s+60}{s+6} \tag{96}$$

The root locus plot and the step response are shown in Figure 4.9 and Figure 4.10. The system response shows a settling time of ~5[ms], a raise time of 1.5[ms], and zero steady state error. The response also shows an overshoot of 2.75% which is well

below the stated maximum overshoot of 20%. Thus the lag compensator satisfies all of the design requirements.

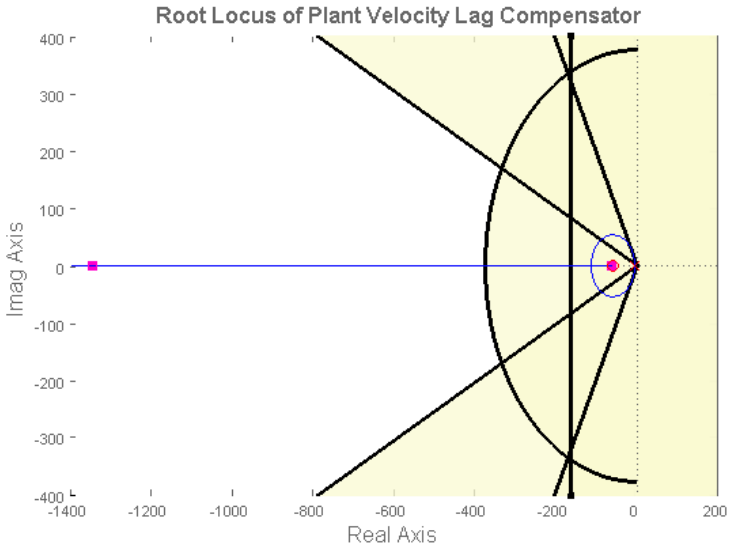

Figure 4.9: Root locus of plant lag compensator

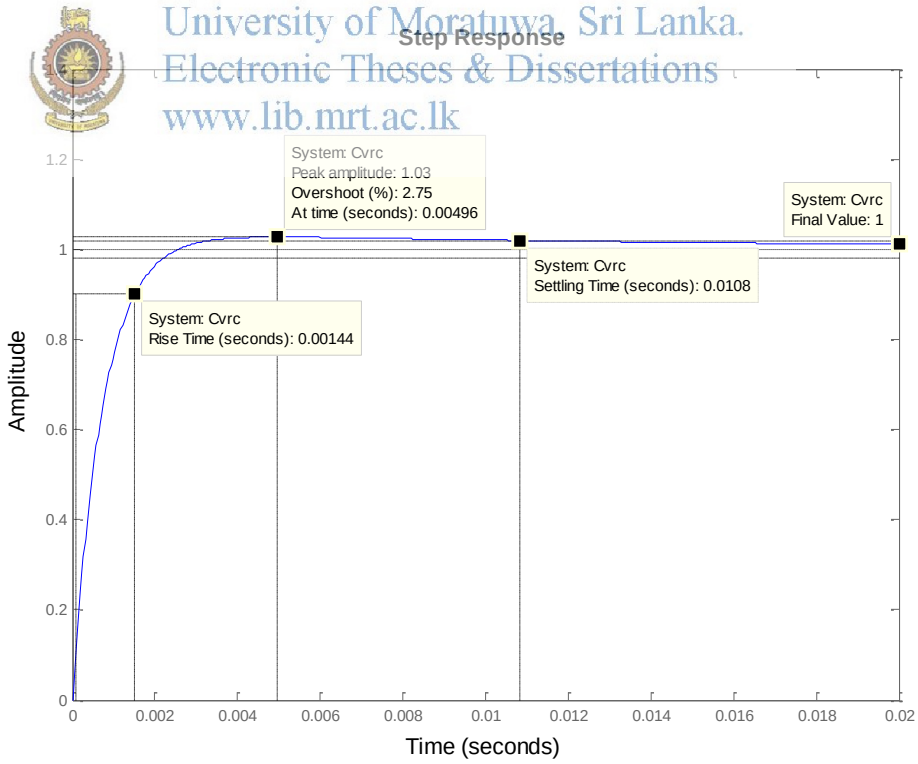

Figure 4.10: Step response of velocity lag compensator

4.2.1.2 Designing the Outer Control Loop

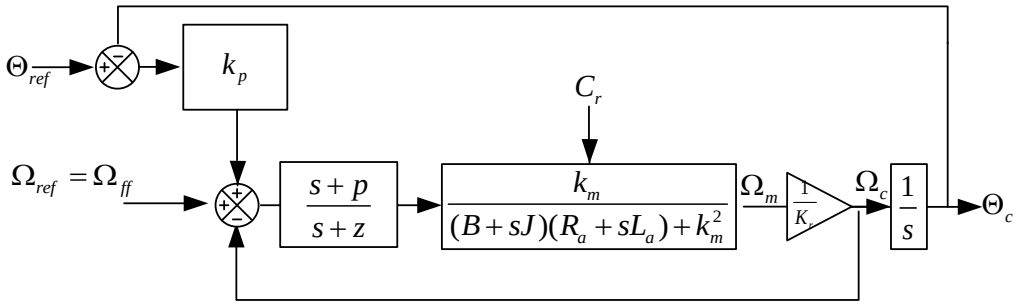

Figure 4.11: Diagram of position controller

In the cascaded loop, the outer loop maintains position; the loop has an open loop transfer function given by:

$$G_\theta = \frac{1393s + 8.361\times10^4}{s^3 + 1406s^2 + 8.365\times10^4 s} \tag{97}$$

The closed loop plot of the system

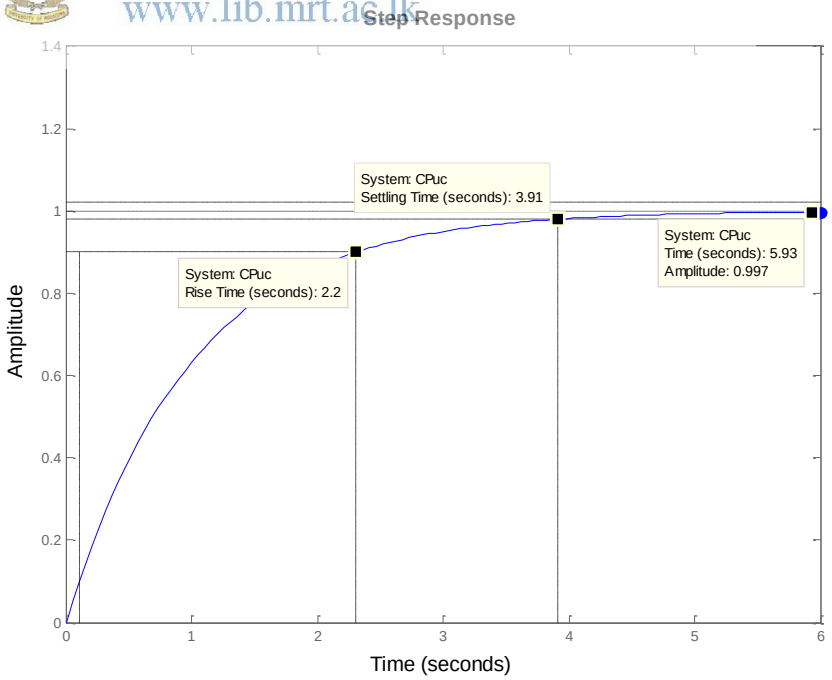

Figure 4.12: Outer loop, closed-loop step response

Once again referring to the design parameters given in section 4.1:

$t_s = 96[ms]$, $t_r(2\%) = 19.2[ms]$, $M_p = 0\%$ which gives $\zeta = 1$, $\omega_n = 93.75$, decay of $\sigma = 47.97$ and zero steady state error. $0 \approx Er_{ss}$

As stated in section 4.1, the bandwidth requirement for the inner loop is much higher than that of the outer loop. The outer loop bandwidth is chosen with a ratio of 4:1, it is better to have a higher ratio in terms of performance; but this will result in infeasibility high steady state times.

A proportional compensator is added to the closed loop. A unit step plot of the closed loop system shows the system to be stable, but with longer rise and settling times of 2.2[s] and 3.91[s]. With the required requirements marked on the root locus plot. A proportional gain is found to be adequate, in order to satisfy the said requirements.

The gain was found to be:

$$G_2 = 112.26 \tag{98}$$

With the said gain the resultant outcome was: settling time of 34[ms], rise time of 17[ms] and zero steady state error.

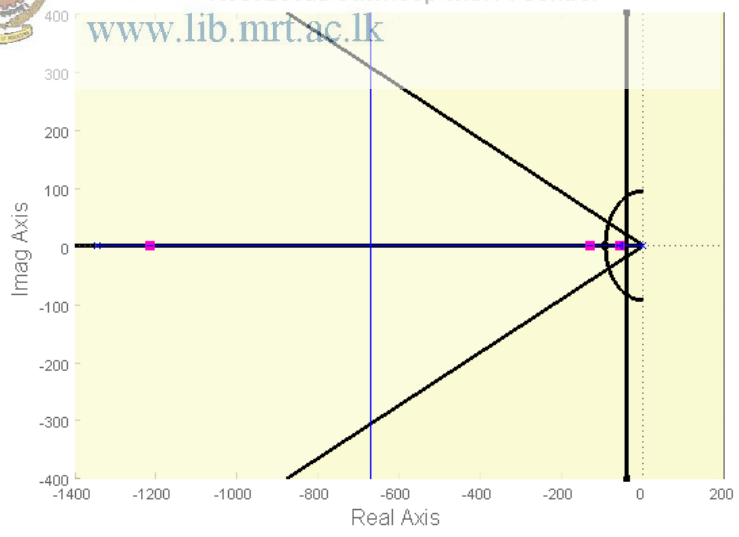

Figure 4.13:Root locus plot of the outer loop controller

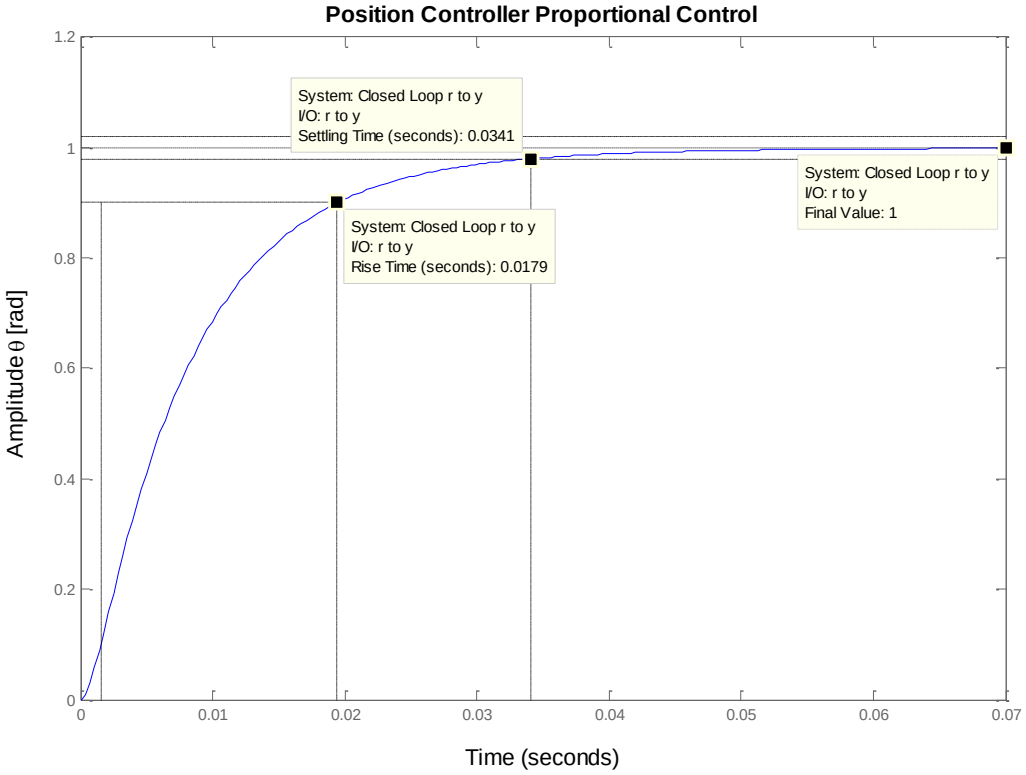

Figure 4.14, Position controller step response

4.2.2 Digitizing the Controller

In order for the controllers to be implemented in a microcontroller, the derived continuous time Lag Compensator was converted to discrete time. For the conversion zero-order hold (ZOH) is used, assuming the control inputs are piecewise constant over the sampling period T_s. The outer loop controller containing only a gain does not require conversion.

Two sampling rates are used for the inner and the outer loops. Given by the requirements, at least 6 samples should be encountered before passing the rise time interval.

For inner loop controller, the lag compensator is converted as:

$$C_1 = 110 \cdot \frac{s+60}{s+6} \Rightarrow D_1 = 110 \cdot \frac{(z\text{-}0.9521)}{(z\text{-}0.9952)} \tag{99}$$

With a sampling time of $T_s = 800[\mu s]$

and the outer loop controller, is converted as:

$$C_2 = 112.26 \Rightarrow D_2 = 112.26 \quad (100)$$

With a sampling time of $T_s = 3.2[\mathrm{m}s]$

The conversion of the Lag compensator is validated using a bode plot as shown in Figure 4.15. The figure shows identical shapes for the continuous time and the discrete controller. In the plot

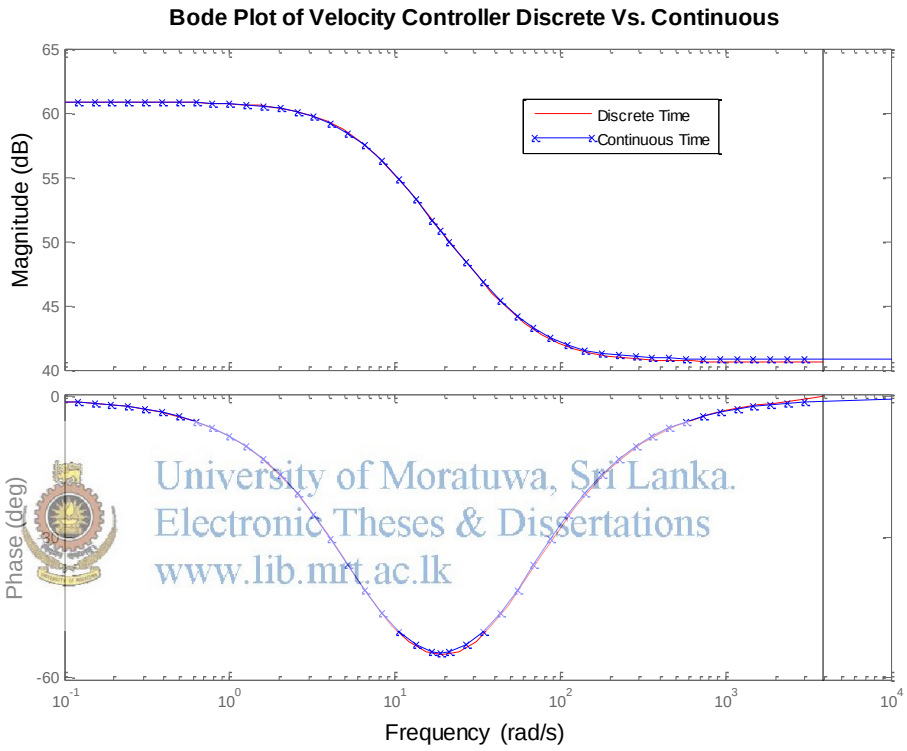

Figure 4.15: Bode plot of continuous vs. discrete controller

The controller performance is validated with the requirements in section 4.1 in discrete time. The results for the inner loop- velocity controller are shown in Figure 4.16 and the results for the outer loop- position controller are shown in Figure 4.17. In both plots, the discrete time signal is plotted in blue, while the continuous time signal is plotted in green.

The plot of outer loop position controller in discrete time, with a rise time of 14[ms] and settling time of 26[ms] closely resembles the continuous time plot, which has a

17[ms] rise time and a 34[ms] settling time, still below the maximum rated in the design requirements.

The inner loop – velocity controller in discrete time, has a settling time of 1[ms], rise time of 0.5[ms] and an overshoot of 11.2%, in comparison to the continuous time plot which shows a settling time of 5, rise time of 1.5 and an overshoot of 2.75%. The shape of the discrete time curve is slightly different at the start of the step input due to the minimal number of sampling points; this is mostly visible in the change of overshoot from 2.75% to 11.2%. Since the overshoot is still less than the specified 20%, no tuning was done to reduce it.

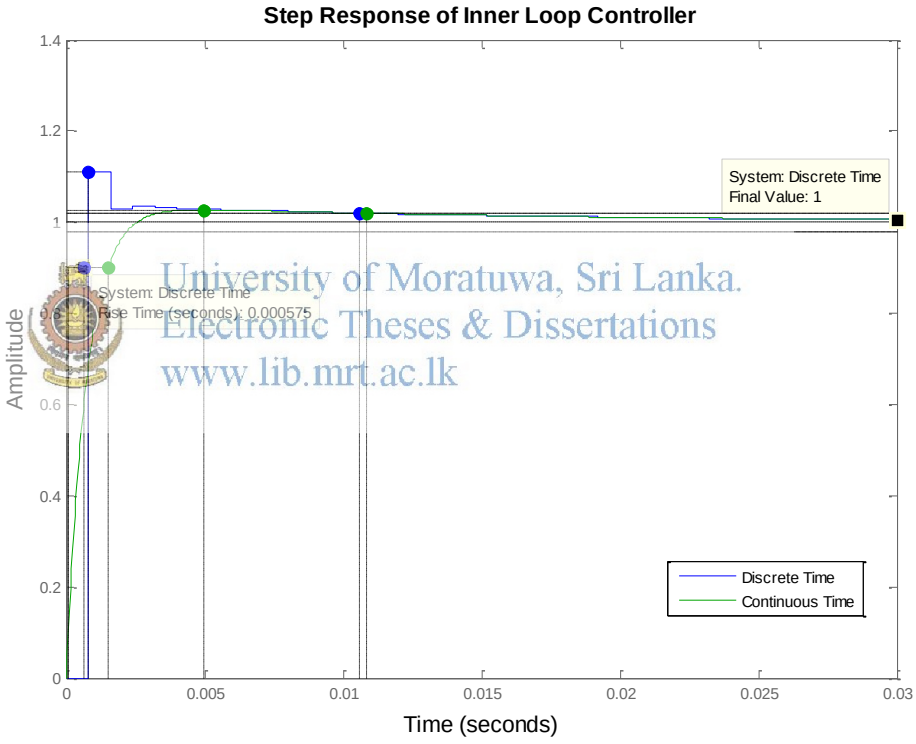

Figure 4.16: Inner loop controller step response

Once the discretization is done, the controllers are converted to difference equations for the microcontroller implementation.

The inner loop is converted as:

$$D_1 = 110 \cdot \frac{(z\text{-}0.9521)}{(z\text{-}0.9952)}$$

$$\Downarrow \tag{101}$$

$$u[k] = 110(e[k] - 0.9521 \cdot e[k-1]) + 0.9952 \cdot u[k-1]$$

The outer loop being a gain, this is converted as:

$$D_2 = 112.26$$

$$\Downarrow \tag{102}$$

$$u[k] = 112.26 \cdot e[k]$$

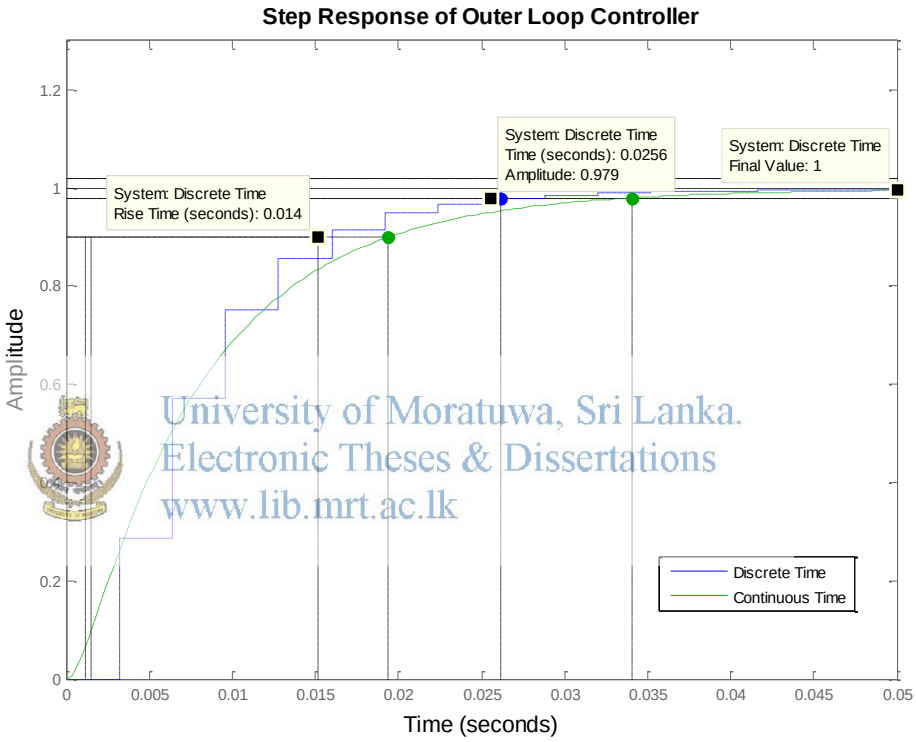

Figure 4.17: Outer loop controller step response

5 PROTOTYPE IMPLEMENTATION

This chapter describes creating a simple prototype of the design and concepts described in the previous chapter. The prototype creation includes creating a physical model, controller electronics and firmware and a software client to operate the robot manipulator. The high level architecture of the robotic manipulator is depicted in Figure 5.1

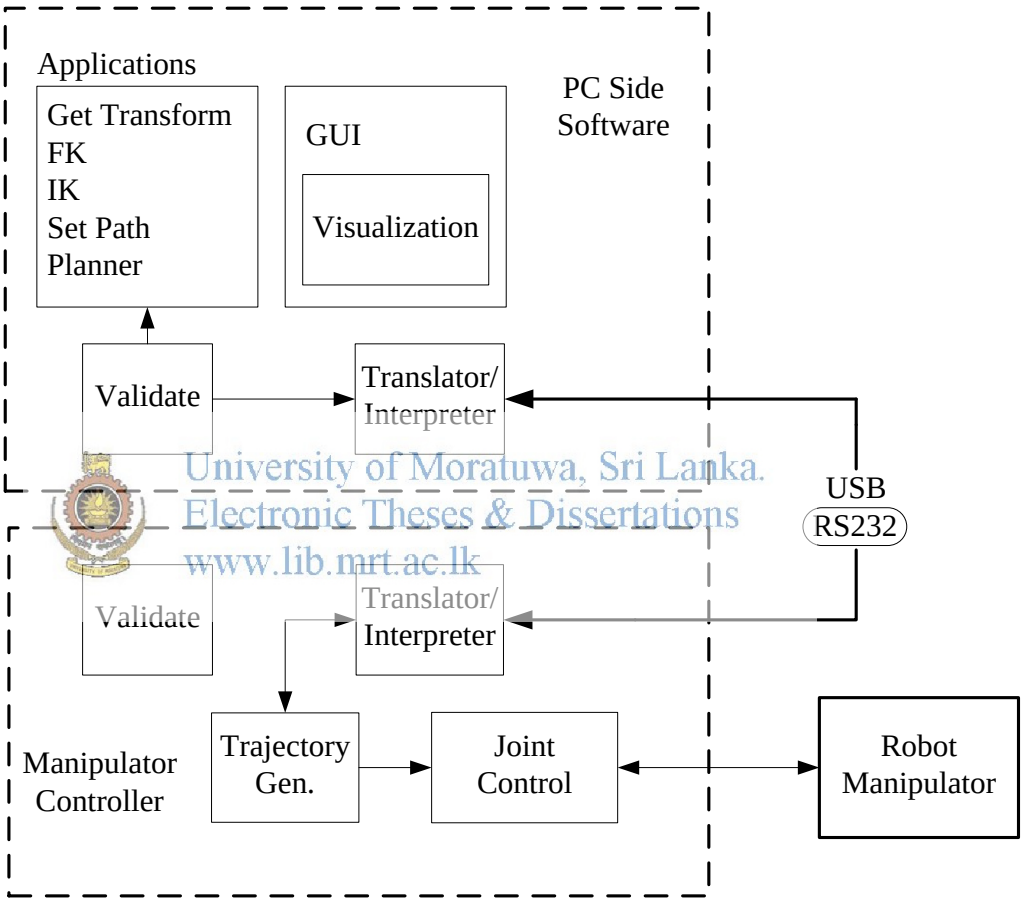

Figure 5.1: High level architecture of the manipulator controller

5.1 Manipulator Linkage Implementation

The manipulator contains two complicated sets of joints. At the base; the origins of the first three joints intersect on one axis point, creating a three degree of freedom joint. Similarly, the last two joints near the end effector have its axis intersecting,

creating a two degree of freedom joint. Link three has significant link length that separates the three axis point of the base. Similarly, length of link four separates the two axis point of the end effector from joint axis four. Thus there are only two displacements that create the entire manipulator length. The links mainly consist of the drive and mounting brackets. The last link has a mounting bracket for a surgical drill.

Figure 5.2: Manipulator in CAD vs. physical implementation

5.2 Joint Actuator Implementation

The joint actuators are simply geared DC machines with position and velocity servo control. The drive unit is near the vicinity of the base of the next joint, coupled through a shaft to the reduction gearbox. The drive unit contains of rare earth permanent magnet DC (PMDC) brushed motor, the limiting factor of deliverable torque is the overheating of motor windings. Use of rare earth magnets in the PMDC

motor enables high peak torque. For servo control, position feedback is acquired from the actuator shaft, using a co-located position sensor.

The drive parameters are given Table 5.1

Table 5.1: Drive and Motor Parameters.

Parameter	Symbol	Value	Unit
Motor torque constant	k_m	2.28E-03	V.s.
Motor inertia *	J	25.0E-09	Kg.m^2
Drive viscous friction	B_D	7.70E-09	Nm.s
Drive dry friction	τ_{Ddry}	4.02E-05	Nm.
Motor viscos friction	B_M	2.09E-08	Nm.s
Motor dry friction	τ_{MDry}	2.49E-05	Nm
Gear ratio	K_r	214.28	
Motor inductance	L_a	15.002	µH
Motor resistance	R_a	1.09E+01	Ohm
Motor stiction	τ_{MStic}	1.36E-04	Nm
Drive stiction	τ_{DStic}	1.92E-04	Nm
Total dry friction	τ_{Dry}	6.52E-05	Nm
Total viscous friction	B_t	2.86E-08	Nm.s
Max drive speed	ω_{Max}	7.60E+00	rad/s
Max encoder resolution		0.05	Degrees

**note Motor inertia is an estimated value*

Drive parameters were found using experimental results and estimation, except for that of Motor inertia. Motor inertia is an assumed / estimated value based on similar motor datasheets. The measurement journals for the motor and drive parameter measurement are found in the appendix

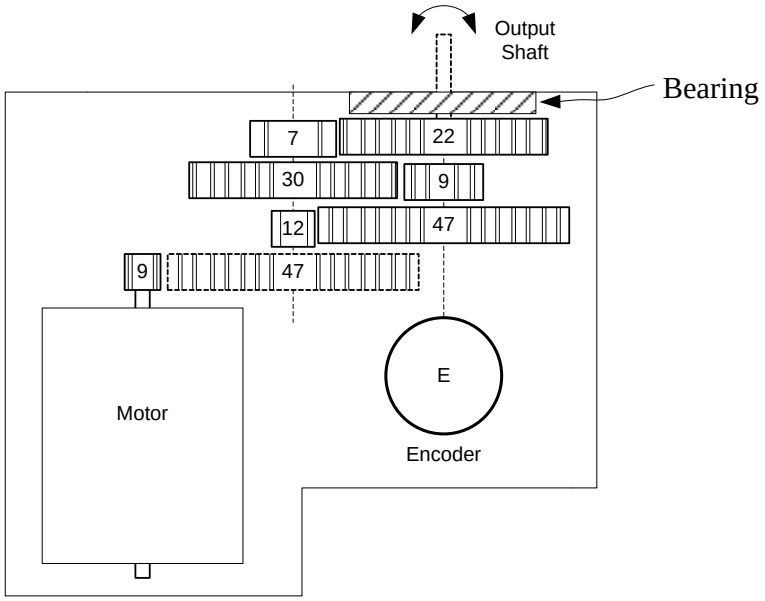

Figure 5.3: Diagram of the drive layout.

5.3 Implementation of Controller Circuitry

The electronics of the robotic manipulator consists; power regulation, control electronics, data acquisition, motor control and communication.

An Atmel ATMega 328 microcontroller serves as the master controller (MC), passing commands to three Atmel ATMega 328 microcontrollers acting as joint controllers (JC). Each joint controller microcontroller (JCM) controls two independent joint actuators. The JCM has a fixed loop time of 800 [μs]. The controller is designed such that, servo controllers and algorithm processors can operate at maximum processor loop time, which is the constraining factor in this board design. The MC and JCMs are connected by an Inter-Interconnected (I2C) bus. Each joint is equipped with an absolute position sensor. Two position sensors corresponding to two joints are connected to each JCM via Serial Peripheral Interface (SPI) bus. In order to command torques to the DC motors, the JCMs use its digital to analog converter (DAC) to command motor driver circuitry. The current flowing through the motor is controlled by adjusting the voltage across the armature as required. Voltage and current feedback are monitored using the JCM's 10-bit

analog to digital converter (ADC). The motor driver circuitry includes a solid state motor driver IC, capable of handling 0.6A per phase. The MC handles translation and high level operations; including interpreting command input from a personal computer (PC) through the MCs USB bridged serial input. An auxiliary microprocessor (AMC) ATMega 128 is similarly connected to the MC using the I2C bus. This controller is for additional processing requirements (e.g. resolved rates calculation). Power regulation is required for proper operation. The drives are powered by a 5V supply, capable of providing 4A. Control electronics require a supply of 5V with 1A. The position sensors require a noise filtered supply of 3.3V, of 100mA

Ideally a field bus should be used for connecting the sensors, actuators to the controller. CAN bus is a commonly used field bus, mostly used for automotive and industrial applications. A specific variation of CAN bus is also available for medical applications. More complex field buses used for medical devices include Spacewire etc. In this design the distance from the sensors, actuators, to the controller is considerably short, and the servo controllers are not in the manipulator itself. This enables the controller to use SPI or I2C instead of field bus.

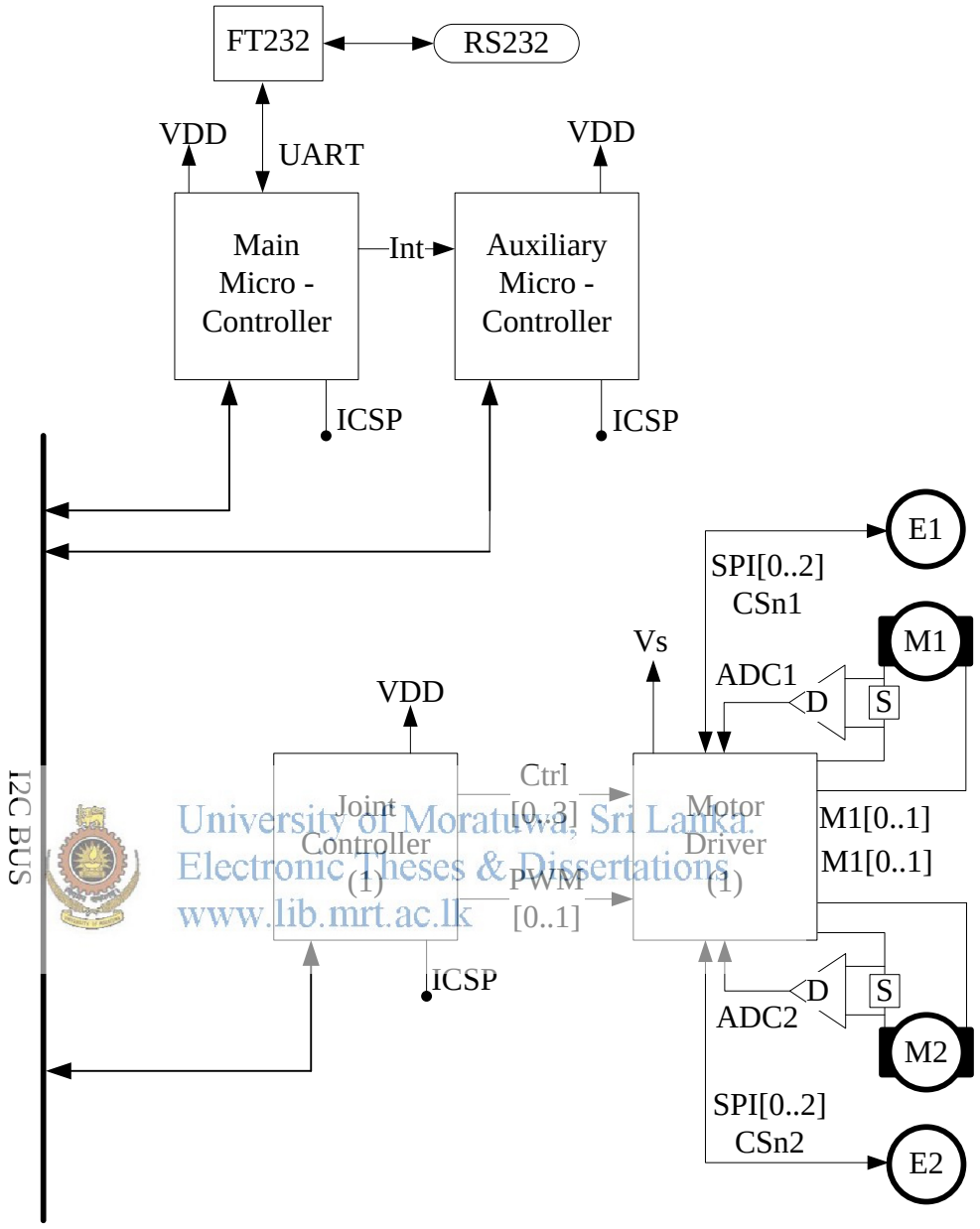

Figure 5.4: Diagram of controller electronics layout

5.4 Robot Controller Firmware Implementation

The electronic design uses a number of embedded microprocessors: MC, JMCs and AMC, which require firmware. All the processors are of the 8-bit RISC AVR architecture. The microcontrollers source code, can be complied either; directly using

AVR® assembler, AVR C, which is a subset of ANSI C language or by using a wrapper library such as Wiring® or Arduino® that translates to AVR C. All firmware used in this research, uses the Arduino wrapper library.

An SPI based magnetic position sensor is used for acquiring joint angles. The sensor requires parity checks, during each transmission, diagnostic features in the sensor check if the acquired data is correct. If there are transmission errors, the sensor connection needs to reset. Velocity is calculated by differentiating time sampled position information, this appears to be noisy. Thus a low pass filter and a moving average filter are used for countering this.

There are a number of limitations with the selected microcontroller, one of them is that the ATMega microcontroller family does not have multi-threading. Thus functions such as the UART use the main loop for data processing, causing delays in the main loop.

The firmware is posed to be minimalistic and fast, contains the controller, communication and validation features such as joint limit validation.

5.5 Communication Interface

The robot manipulator can be commanded using a serial (RS232) interface at 57600[bps]. The robot controller acts as a command interpreter. The command format is similar to the G-Code format. A complete set of commands are listed in the appendix.

5.6 PC-side Software Implementation

The PC side software (PCS) is the graphical user interface (GUI) and client to the serial host of robotic manipulator, enabling the user to command and run a set of application on the manipulator. The functionality of the PCS include: direct positioning of manipulator, setting trajectories, position acquiring and a debugging consol. A user guide is on the appendix.

PC side software was implemented using the DOT NET libraries of the Microsoft Visual Studio. Manipulator specific mathematical functions were created in MATLAB and later complied as application extensions, to be used inside the PCS.

6 RESULTS

6.1 Robotic Manipulator Design

6.1.1 Design validation

Validation of the design was conducted using MATLAB and the Robotic toolbox by Peter Croke [16]

6.1.2 Workspace

The current prototype of the robotic manipulator only uses the fixed workspace, bounded by the manipulators reachable space, between the joint angles of the first octant defined by joint angles $[0,0,0,0,0,0]$ to $\left[\frac{\pi}{2},\frac{\pi}{4},\frac{\pi}{4},\frac{\pi}{2},\frac{\pi}{2},\frac{\pi}{2}\right]$

This workspace definition is taken as convenience, and it is not optimal. Limiting the workspace to the first octant greatly simplifies the inverse kinematics and other functions used in the control software.

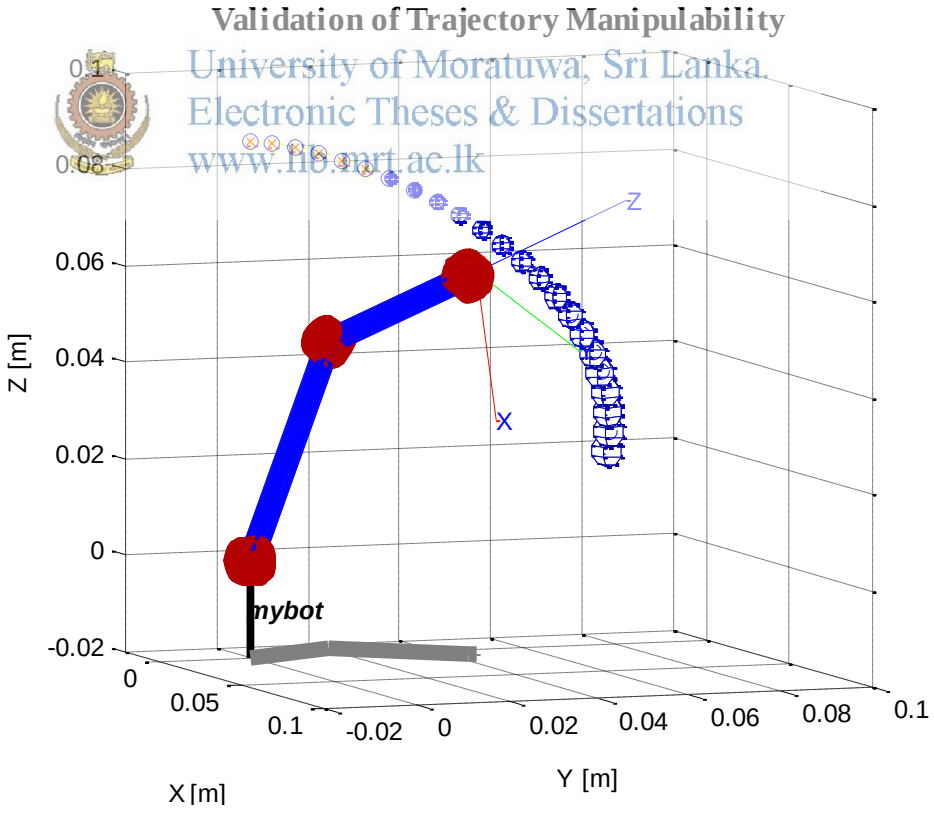

Figure 6.1:Trajectory manipulability validation

Ideally, the workspace boundary should have been numerically determined and plotted. Since there is a localization tool, after the localization has been done, the manipulator software would perform checks using forward kinematics, in order to validate, the points of target in work space (bounded by the localization points) are out of the reachable space, this is done by checking the manipulability of the manipulator. Manipulability of the robotic manipulator is determined within the bounded region for a given path using Yoshikawa's manipulability measure [16, 20]. For each given path, the manipulability can be plotted to verify reachability etc. The plot Figure 6.1 shows the manipulability for a selected trajectory. It can be seen in the plot, for the given path, the points beyond 0.08[m]: marked in yellow, have very low manipulability, this is due to the joint singularity at when all joint angles are at 0 [rad]. And as the points at the lower part of the path have a higher manipulability and they are plotted with a scaled sphere, which corresponds to the manipulability. Thus for a given path, the manipulability of the manipulator is verified before commencing movement.

6.2 Registration Process Verification

The registration algorithm was verified using the MATLAB environment. As shown in Figure 6.2, given two sets of point clouds, a transform T_{mt} is found to relate model point cloud $P_m \in \{3,2,1\},\{7,0,3\},\{9,5,2\}$ with $R_x(0)R_y(0)R_z(0)$, to its corresponding physical points are located at $P_t \in \{2,2,1\},\{6,2,3\},\{6,6,2\}$. Using the transform, model point can be transformed to physical points. In Figure 6.2, the point shown in yellow of the model coordinate system is transformed to the point shown in green of the physical objects coordinate system.

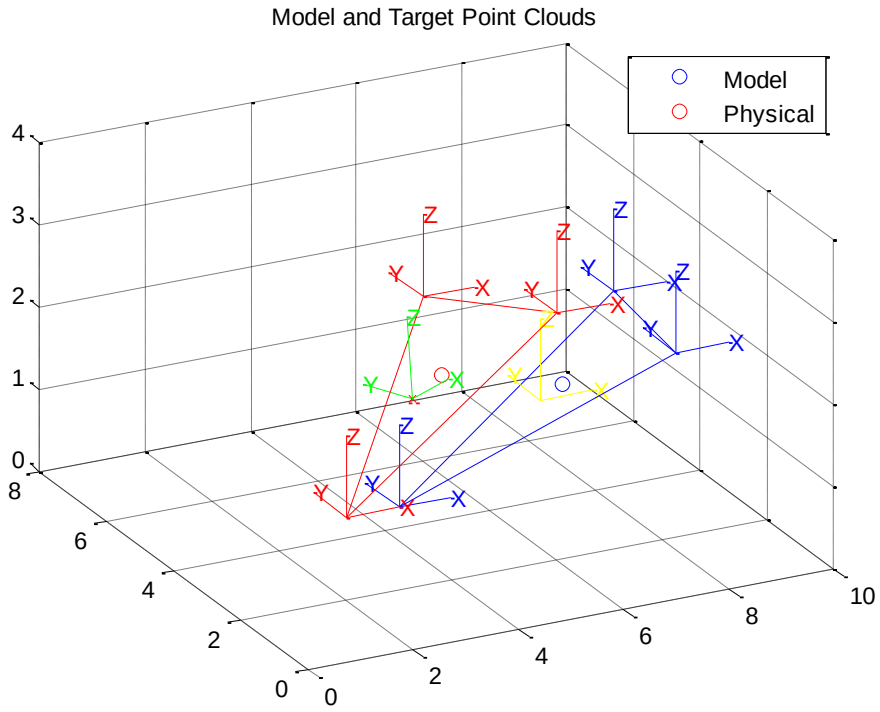

Figure 6.2:Plot of model to target transformation

6.3 Trajectory Following Validation

As noted in section 6.1.2 the manipulator end effector movement in the workspace, is determined by the manipulability. In order to test the designs kinematic configuration and numerical inverse kinematic generation was tested in simulation using MATLAB and Simulink software. Instead of using closed form inverse kinematics, for which the position and orientation both has to be specified, a trajectory was generated by inverting the manipulator Jacobian. The inverse Jacobian allows one to move the end effector to a position only target, while applying constrains on orientation.

A wave like trajectory with a circular radius of 10[mm], with displacement in the z-axis of 10[mm] from the x,y-axis plane was generated at coordinates [61.8, 20.1, 17.3][mm]. The path is to be followed with a fixed velocity of 0.6[$rads^{-1}$] the target path and the resultant actual path are shown in green and red in Figure 6.3.

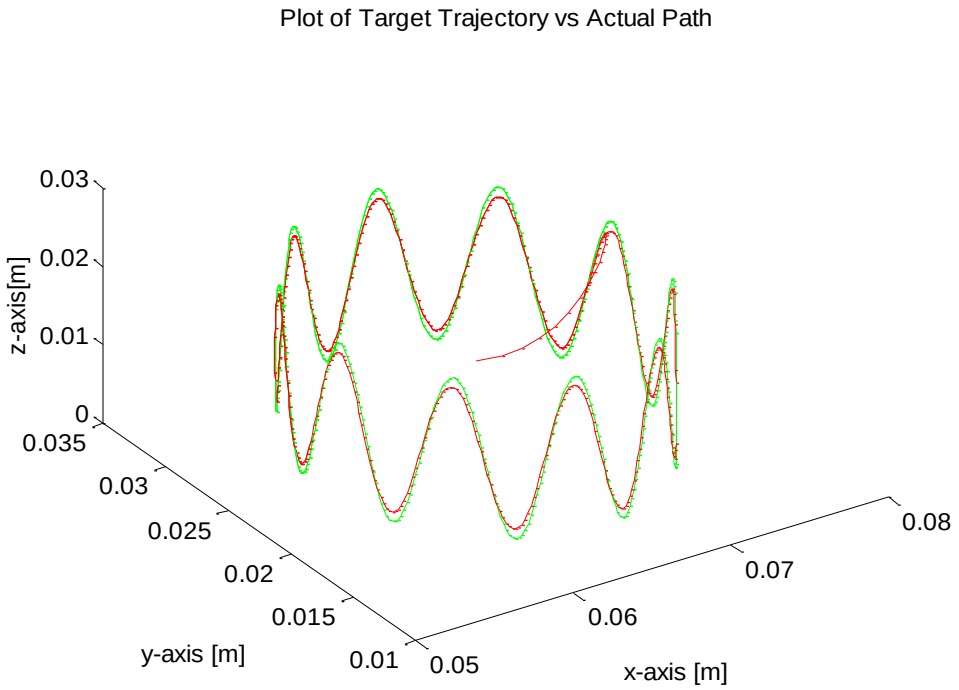

Figure 6.3: Plot of target trajectory and actual path

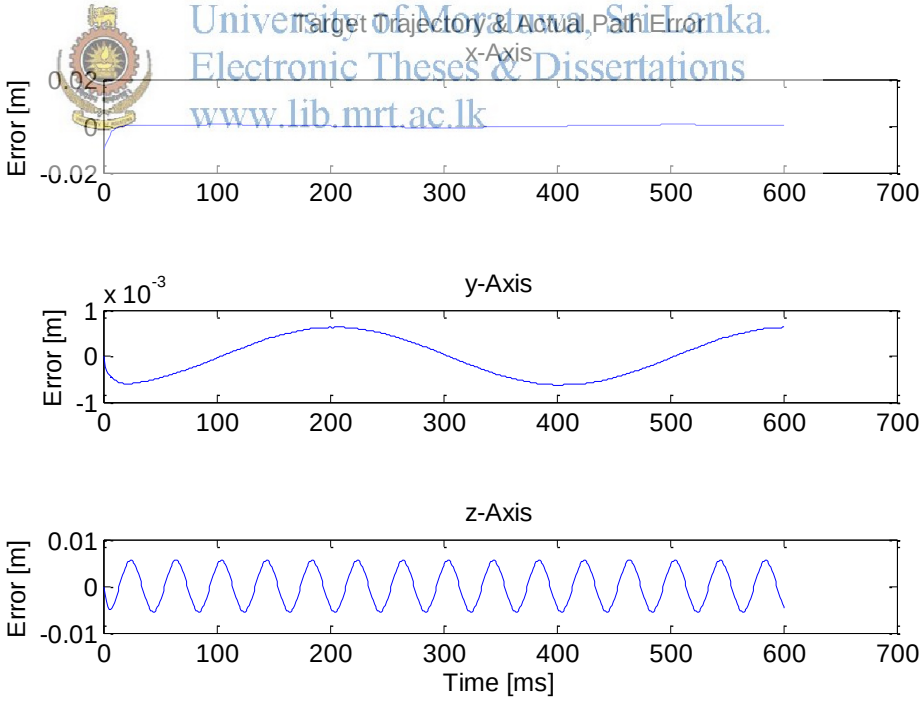

Figure 6.4: Cartesian error at the end effector

The resulting end effector movement and joint motion are looked at in Figure 6.4 and Figure 6.5. The Cartesian end effector movement for the given path has significant error in the z-axis, with tolerable error in the x and y-axis. The reason for this is that the joints 2 and 5 are in use for the major movement of the z-axis in the given configuration, as can be seen in Figure 6.5 Thus faster z-axis movement can be obtained by locking either of the joints. This is useful in generating of paths.

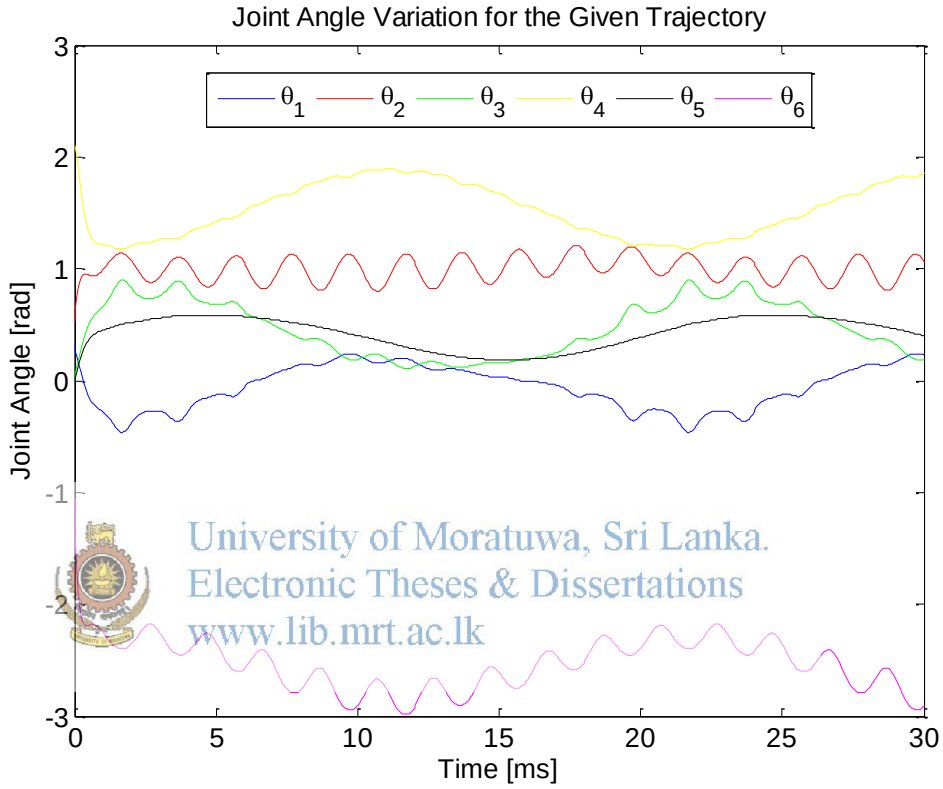

Figure 6.5: Joint angles for the given trajectory

6.4 Joint Controller Validation

Using the cascaded joint controller, designed in chapter 4.2, a situation was carried out in Simulink. The controller was made to follow a reference trajectory. A trajectory plot on based on a path from 0[rad] to 0.5[rad] (0[Deg] to 30[deg]) in 0.5[sec] based on a trajectory generated linear segments with blended splines as specified in section 2.9, the output of the path following capability is plotted in Figure 6.6 . In the plot, the set path is shown in blue while simulated path is shown in red. It can be seen in Figure 6.7 that the velocity and position are tracked well with

an error of less than 0.02[$rads^{-1}$] for velocity and less than 1.8×10^{-4} [rad] for position.

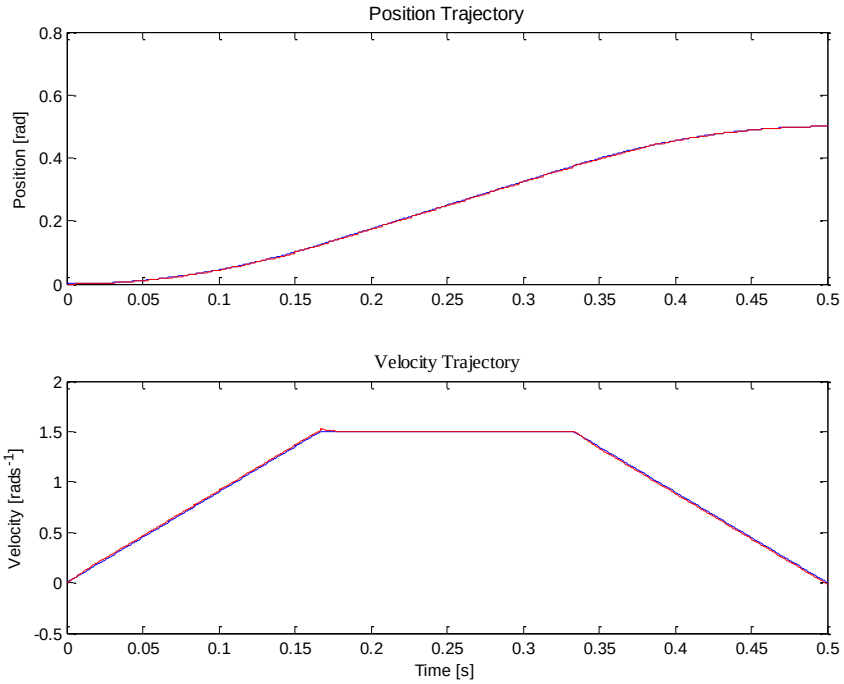

Figure 6.6: Controller trajectory following in simulation

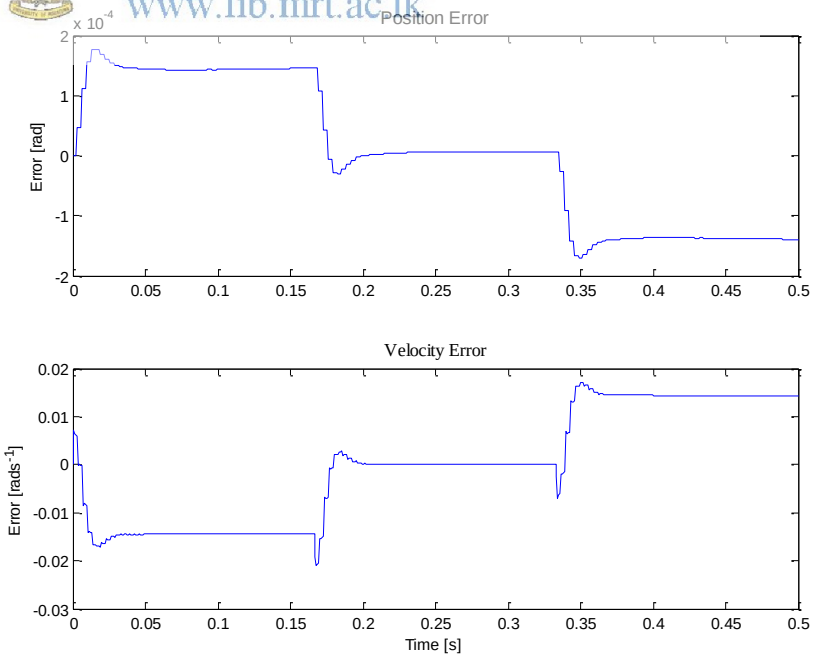

Figure 6.7: Error in trajectory following

It is notable that tracking capabilities depend are dependent on the velocity, and as velocity increases the tracking capability degrades.

Using the cascaded joint controller, designed in chapter 4.2, the physical joint actuator, was made to follow a reference trajectory. A trajectory plot on based on a path from 0.0175[rad] to 1[rad] (1[Deg] to 57[deg]) based on a trajectory generated linear segments with blended splines as specified in section 2.9., the output of the path following capability is plotted in Figure 6.8. In the plot, M[1…6] are recorded datasets of the position at the given time. The simulated position is plotted in red in the same subplot. The input velocity is shown in the subplot below. It can be noted that the target trajectory is tracked accordingly for the given input position and velocity variables. The positioning error is approximately 0.008 [rad] from the above measurement.

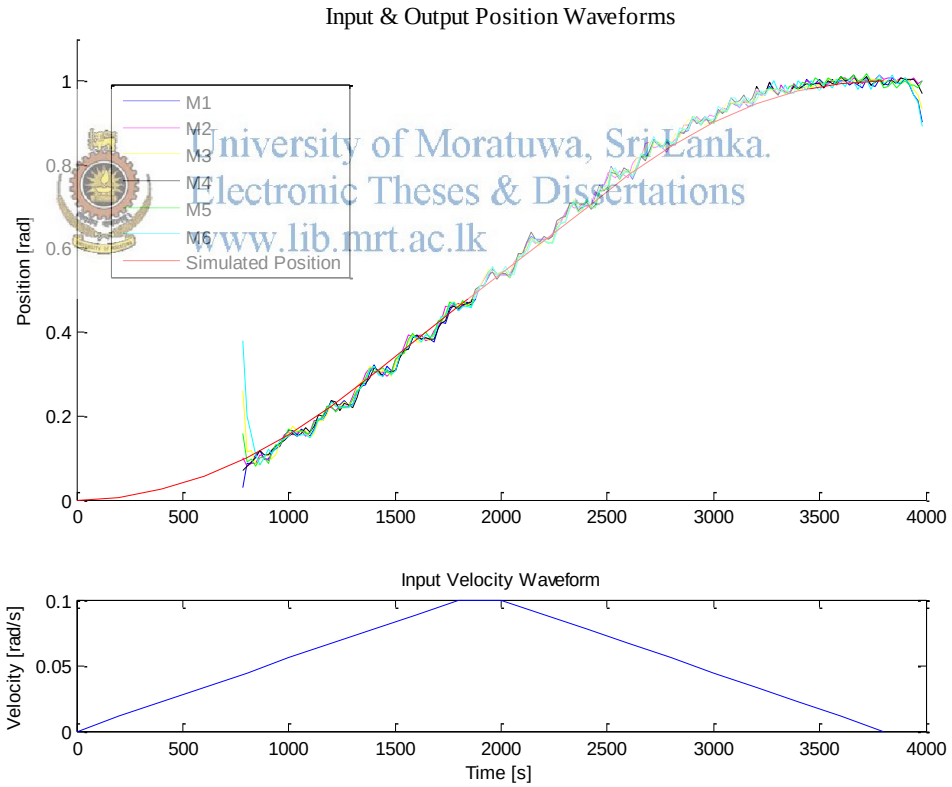

Figure 6.8: Drive trajectory following waveforms

The controller can be improved by adding current feedback. This could be done by adding in another loop, prior to the velocity loop. Thus the bandwidth requirements

for the torque controller would be higher than that of the velocity loop. An issue in having a current feedback is to monitor the current inside the ATMega microprocessor, at a high sampling frequency. The average latency for the ADC in an ATMega is greater than a 100 [μs]

6.5 Use Case

A simple use-case scenario (a non-clinical test) is conducted using the procedure stated in *Appendix B.*

6.5.1 Test Description

A table tennis ball resembling, the target robot workspace, sits in the vicinity of the robotic manipulator. The ball is marked with four points: resembling four land marks. The objective is to mark points on the ball, using a pencil attached to the end effector of the manipulator.

6.5.2 Test Procedure

The table tennis ball is modeled is software, resembling a reconstructed computer tomographic image. The four land marks corresponding to the physical workspace object are also marked on the virtual model. As described in Appendix B; the model is loaded into the software, and registration is carried out. Once the registration is carried out; the target path was loaded into the software and the points were marked. The deviations of the points etc. were recorded for examination.

6.5.3 Test Results

The given landmarks marked in Figure 6.9; in blue, the corresponding points were marked in the workspace by moving the manipulator. The landmarks and the corresponding workspace points are shown in Table 6.1. Using this correlation, the software computes a transformation matrix, from which it calculates the corresponding points for any set point within the workspace.

Using a set test point [4,6,18.5] , the corresponding target point is calculated by the software to be [79.3230,69.4128,32.3316]. The manipulators physical positioning of the said targets was measured. The target and measured points are shown in green

and red crosses in Figure 6.9. The results show some deviation from the expected target, as seen on Table 6.1.

Table 6.1: Results of Position Test

	Landmarks [mm]			
Model	[6,2,19]	[-6,-2,19]	[6,-2,19]	[-6,2,19]
Workspace	[71,60,30][	[62.1,70,28.4]	[69.1,63.5,25]	[64.3,70.5,33.6]
	Set Position [mm]			
Model	[4,6,18.5]			
Workspace	[79.3230,69.4128,32.3316]			
Measure	[3.5,7,18.5]	[6,5,18]	[3,4,18]	[2,5.5,18.5]

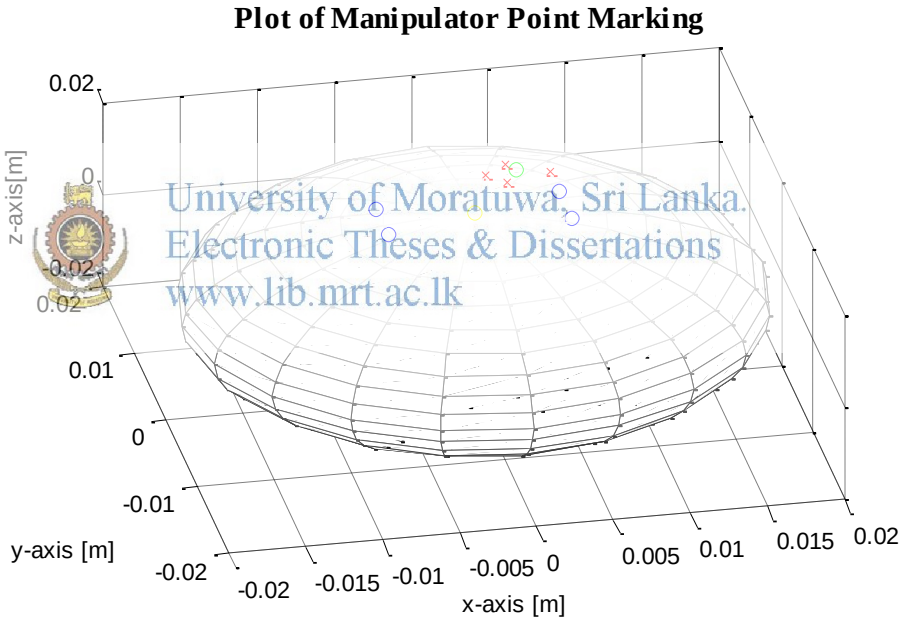

Figure 6.9: Plot of manipulator point markings

7 CONCLUSION

7.1 Prototype Limitations and Considerations

There are a number of factors that undermine the accuracy of the built prototype. In this work, it is assumed that gearing, shafts, bearings and the driven link are infinitely stiff. In reality all these elements have finite stiffness and flexibility. Thus deformations and resonances are not modeled, and are not accounted for. In terms of construction: The linkages are made out of 3D-printed ABS plastic, with the best facilities of 3D-printing available; they were not precise in lengths, and alignment. In assembly of the robotic manipulator, the printed parts do not mate perfectly, thus there are probable miss-alignments, the softness of the plastic causes minor deformities, under no load, and considerable deformity under load (especially when accelerating). Minimal attention was given to mechanical engineering design: there are no bearings used etc.

The design focus on using optimized servo control, this allows reduce joint position and velocity errors, thus in joint space movement maybe considerably accurate; but this does not guarantee accuracy, in operational space, due to the fact that there is no feedback mechanism from the environment, such as force, vision etc.

In the developed simple prototype, operation workspace is only limited to one octant. (Elbow and all angles positive), the current octant of operation was chosen because of ease, but this may not be the optimal octant to use in a single octant operation. For all octant use further work is required to analyses boundaries in all octants. If a single octant operation is required, the best octant can be determined using an optimization algorithm [21]

The choosing of the link sizes were done, arbitrary in order to achieve a speculated scale for the manipulator. For an optimized multi-link robotic manipulator, the common practice of finding optimal link sizes that maximize dexterity [13] would be to use a form of optimization to calculate the link lengths.

Generally, an optimization algorithm is used for path planning, for the best path selection, avoidance of singularities, collision objects on path etc. In this work, no such path planning is implemented other than simple checks for joint limits.

The “ready pose” and other “natural poses” of the manipulator are also determined using optimization algorithms. Use of optimization of all forms on path planning, natural poses are left for future work and are not addressed in this work.

The drives used in the current prototype are in majority parts of low cost toy-grade servos, they are bulky and inefficient in comparison to motor drives offered by domain specific vendors [22, 23].The physical sizes of the drives are a dominant factor in stating the joint angle limits, in the designed manipulator. Thus by using smaller drives, with similar or better performance, the joint limits can be extended, and performance of the entire system can be elevated.

For the construction of the links of the robotic manipulator, a stiff and lightweight metallic alloy should be used. At present the links for the prototype are constructed in 3D-printed ABS material which is rigid and lightweight. It is advantages in terms of cost effectiveness, in comparison to building the same with a lightweight metallic alloy.

The drill tool uses diamond dust covered burrs. A study on the amount of force exerted by the drill on the workspace can be identified and controlled by acquiring force feedback from the tool. The velocity of drill movement also is a factor worth studying. If the burr is moved too slowly, it would cause the flesh to burn. If moved too fast, it would cause physical damages. Father investigation of using feedback information (torque, rpm etc.) from the drill can be used for determining burr breakthrough. This can help prevent excess drilling, burning of the bone. [24, 25]

7.2 Conclusion

This thesis a study was conducted on improving the drilling and cleaning process of cochleostomy, in surgeries such as cochlear implant surgery. As a solution to the complications that occur during the problematic task of cleaning and drilling, a robotic surgical micromanipulator was introduced as a helper tool to ease the task. The thesis aims to provide a description of the design, analysis and prototype implementation of a surgical micromanipulator, created specifically for the drilling and cleaning task in the cochleostomy procedure.

A kinematic configuration for the robotic manipulator was synthesized using the conceptual design considerations; the achieved resultant design was a unique six

DOF manipulator, with the size of a human hand. The manipulator consists of one three DOF joint at the base, one DOF joint at the elbow with an offset and one, two DOF joint. Thus the resultant design exhibits the likeness of an anthropomorphic robotic manipulator, with only two lengthened links. The design enables the manipulator to obtain the majority of positioning from the lower portion of the manipulator, and precise orientation and positing from the upper portion. The elbow configuration enables the microscope be attached to the surgical microscope, providing the surgeon with visual and tool accessibility, similar to a conventional cochlear surgery.

Attention was paid to closed form solutions of inverse kinematics and singularity analysis, for the designed manipulator configuration. Due to closed form solutions it is possible to choose the operating workspace of the robot as such that, robot joint limits and singularities configurations are avoided. This is important to certify correct operation.

Analysis of system dynamics were shown, the analysis included inertial, payload and gravity loading. This provided the mean inertia values which are required for drive selection and controller design. The largest inertia variation of the links is 3.1676×10^5 [g.mm^2] and variation factor is 1.9860, the largest variation of gravity loading torque was 33.9[mN.m.] when link 2 is stretched horizontally. The robot was designed to operate with a maximum payload of 30[g].

Special attention is paid to the design of controllers. Decentralized joint space control of joint actuators was achieved by using a cascaded position and velocity controller with feed forward action was designed. For the design of controllers, root locus methods were used with a model based approach. Model parameters for the drive were taken from experimental measurement.

For the purpose of patient registration, which is required for the manipulator usage, the surgical workflow alteration is purposed, in which, before manipulator use it is needed to move the end effector to the identified landmarks, from which the model to workspace correlation is obtained using the iterative center point algorithm.

As proof of concept of the theoretical design, a demo prototype was made of the surgical manipulator. The prototype implementation included making the mechanical

linkages, the drive actuators, a robot controller, and control software. A primary goal for the prototype construction was to keep the cost minimal. The manipulator linkages were 3D printed using ABS plastic. The robot controller includes electronics for algorithm processing, communications and servo motion. Control software was designed to communicate with the robot controller, and process commands.

Results of the manipulator design were obtained by simulation and from experimentation using the built prototype.

Paths were created either using closed form inverse kinematics or iterative inverse kinematics using Jacobian inverse. The trajectory following capabilities for iterative inverse kinematics were tested using simulations

The designed cascaded position and velocity controller was validated by testing and analyzing joint motion for a given linear segment trajectory with parabolic blends, for which the controller in simulation tracked velocity with an error of 0.0018 and position with an error of 0.018[rads^{-1}]. In experimental measurements the positing error reached up to approximately 0.008[rad]. Further experiments were conducted on manipulability and trajectory following.

7.3 Final Remarks

In this thesis a description of the design, analysis and prototype implementation of a surgical micromanipulator, created specifically for the drilling and cleaning task in the cochleostomy procedure

The methodology used for the development and simple construction of the manipulator arm can be used for the construction of this or any similar manipulator. Theoretical concepts, design calculations are done with modular MATLAB scripts, and can be easily manipulated for any change in size, materials, drives etc. thus the current economical prototype model can be built at a different scale using higher quality materials, drives, and faster control electronics.

In conclusion, I believe the conceptual design and prototype may prove to be a useful and practical device that addresses a valid problem. Thus further investigation and conducting of clinical trials are a beneficial cause.

REFERENCES

[1] O. Majdani, T. S. Rau, S. Baron, H. Eilers, C. Baier, B. Heimann, *et al.*, "A robot-guided minimally invasive approach for cochlear implant surgery: preliminary results of a temporal bone study," *International journal of computer assisted radiology and surgery*, vol. 4, pp. 475-486, 2009.

[2] S. Baron, H. Eilers, B. Heimann, S. Bartling, R. Heermann, T. Lenarz, *et al.*, "Robotic-guided minimally-invasive cochleostomy: first results," *Deutsche Gesellschaft für Computer- und Roboterassistierte Chirurgie*, vol. 2, pp. 1-7, 2007.

[3] O. F. Adunka and C. A. Buchman, "Scala tympani cochleostomy I: results of a survey," *The Laryngoscope*, vol. 117, pp. 2187-2194, 2007.

[4] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Annals of Surgery*, vol. 239, pp. 14-21, 2004.

[5] J. E. Moore Jr and D. J. Maitland, *Biomedical Technology and Devices*: CRC press, 2013.

[6] C. Coulson, R. Taylor, A. Reid, M. Griffiths, D. Proops, and P. Brett, "An autonomous surgical robot for drilling a cochleostomy: preliminary porcine trial," *Clinical Otolaryngology*, vol. 33, pp. 343-347, 2008.

[7] L. B. Kratchman, G. S. Blachon, T. J. Withrow, R. Balachandran, R. F. Labadie, and R. J. Webster, "Design of a bone-attached parallel robot for percutaneous cochlear implantation," *IEEE Transactions on Biomedical Engineering*, vol. 58, pp. 2904-2910, 2011.

[8] T. Ota, A. Degani, D. Schwartzman, B. Zubiate, J. McGarvey, H. Choset, *et al.*, "A highly articulated robotic surgical system for minimally invasive surgery," *The Annals of thoracic surgery*, vol. 87, pp. 1253-1256, 2009.

[9] J.-J. E. Slotine and W. Li, "Applied nonlinear control," *NJ: Prantice-Hall, Englewood Cliffs*, 1991.

[10] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, *et al.*, "The DLR MIRO: a versatile lightweight robot for surgical applications," *Industrial Robot: An International Journal*, vol. 35, pp. 324-336, 2008.

[11]J. Huckaby and H. I. Christensen, "Dynamic Characterization of KUKA Light-Weight Robot Manipulators," Georgia Institute of Technology 2012.

[12]D. L. Peiper, "The kinematics of manipulators under computer control," Ph. D Thesis, Stanford University 1968.

[13]J. J. Craig, *Introduction to robotics: mechanics and control* vol. 3: Pearson Prentice Hall Upper Saddle River, 2005.

[14]R. P. Paul and H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *The International Journal of Robotics Research*, vol. 5, pp. 32-44, 1986.

[15]J. Lloyd and V. Hayward, "Kinematics of common industrial robots," *Robotics and Autonomous Systems*, vol. 4, pp. 169-191, 1988.

[16]P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* vol. 73: Springer Science & Business Media, 2011.

[17]B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*: Springer Science & Business Media, 2009.

[18]S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, 2001, pp. 145-152.

[19]B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America. A, Optics and image science*, vol. 4, pp. 629-642, 1987.

[20]T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, pp. 3-9, 1985.

[21]R. Konietschke, T. Ortmaier, H. Weiss, R. Engelke, and G. Hirzinger, "Optimal Design of a Medical Robot for Minimally Invasive Surgery, 2," *Jahrestagung der Deutschen Gesellschaft fuer Computer-und Roboterassistierte Chirurgie*, pp. 1-8, 2003.

[22]M. Mainz, *Micromotion - Micro Drive Engineering* [Online]. Available: http://micromotion-gmbh.eu/, [Accessed: 2015, May,4th].

[23]Citizen, *Citizen Micro Co. Ltd.* [Online]. Available: http://www.citizen-micro.com/pro/mortor.php, [Accessed: 2015, May, 4th].

[24]W.-Y. Lee, C.-L. Shih, and S.-T. Lee, "Force control and breakthrough detection of a bone-drilling system," *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 20-29, 2004.

[25]J. Lee, B. A. Gozen, and O. B. Ozdoganlar, "Modeling and experimentation of bone drilling forces," *Journal of biomechanics*, vol. 45, pp. 1076-1083, 2012.

APPENDIX A. DRIVE PARAMETER IDENTIFICATION

A.1 Armature Resistance

Armature resistance was found by

1) Using a multi-meter: resistance was measured across the ports of the DC motor, the value was found to be 10.9 [Ohm]
2) With the rotor shaft fixed, a constant voltage was applied to the ports of the motor, then the current and the voltage were noted down

Table A.1: Measurements for Armature Resistance

Voltage [v]	0.62	0.72	0.84	0.94	1.06	1.17	1.29	1.39	1.51	1.59	1.73
Current [mA]	35.49	60.6	69.4	76.6	87.9	96.9	106.9	116.1	120.4	129.4	136.5

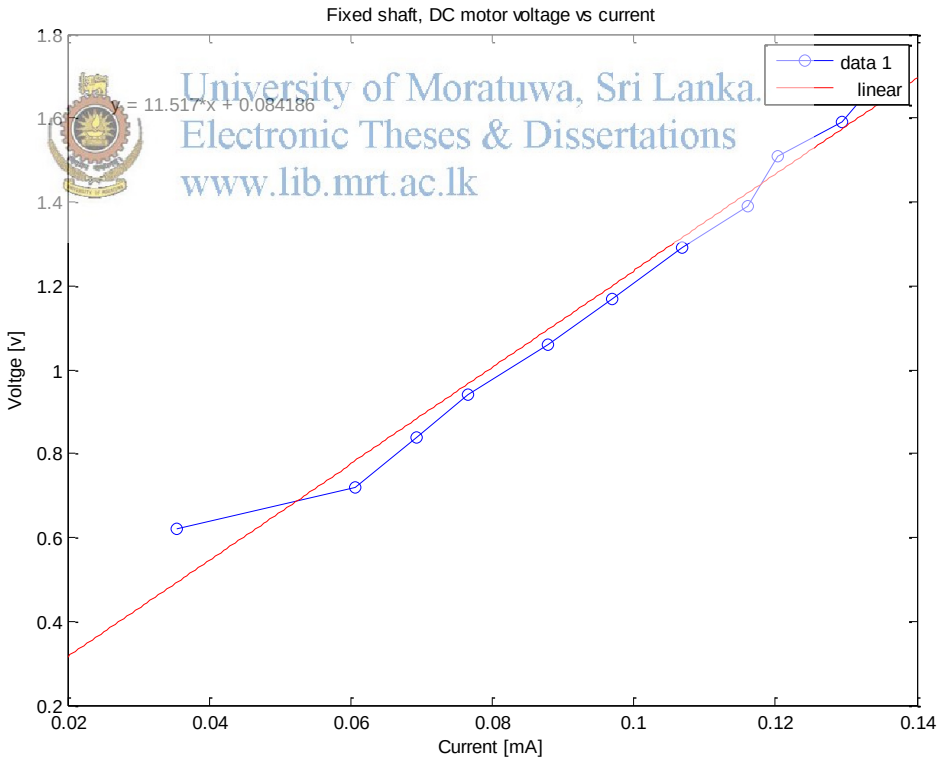

Figure A.7.1: Motor current vs. voltage

The slope gives the armature resistance, the value being 11.517 [Ohm]

A.2 **Motor Inductance**

Inductance was measured using an LCR meter. Mean value of 17 measurements were taken to be the inductance.

Table A.2: Armature Inductance Measurements

Value [uH]								
14.67	16.18	16.16	14.48	14.43	14.33	14.54	16.99	14.91
14.55	14.68	14.64	16.42	14.07	14.1	14.55	15.33	

Mean value: 15.002 [uH]

A.3 **Motor Constant**

The motor constant is equal to the induced voltage in the armature winding divided by the angular velocity

$$K_e = \frac{V_{input} - RI}{\omega(t)}$$

Where: K_e is the motor constant and $\omega(t)$ is the angular velocity.

Table A.3. Measurements for the Motor Constant

Voltage [v]	Current [mA]	Ang. Velocity [rad/s]
0.61	12.43	213.62
0.65	12.98	219.19
0.7	13.28	245.04
0.73	13.8	251.32
0.88	13.86	345.57
0.99	13.99	370.7
1.05	14.56	389.55
1.13	14.75	420.97
1.2	15.07	427.35
1.27	15.37	452.39
1.28	15.46	515.22

At different input voltages; voltage, current and angular velocity are noted down. The slope of the voltage vs. velocity gives the motor constant.

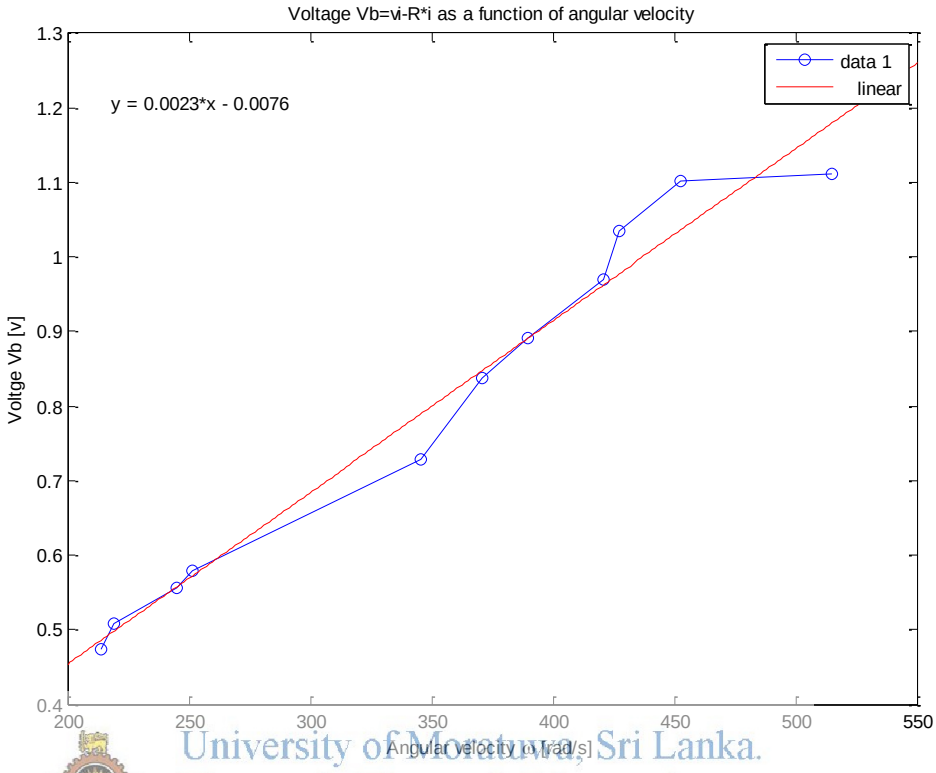

Figure A.7.2: Angular velocity vs. armature voltage

Mean motor constant: 0.0023056 [V.s]

A.4 **Motor Friction**

Motor friction is determined by measuring the armature current and angular velocity, measurements from the previous sections were used.

$$I \cdot K_t = B_m \cdot \omega_m(t) + \tau_{mdry}(t) + \tau_{mstic}(t)$$

Where: B_m is the viscous friction, τ_{mdry} is the dry friction and τ_{mstic} is the stiction of the motor. Neglecting stiction, the slope of the plot $K_t \cdot I$ vs. $\omega(t)$ gives the viscous friction of the motor. The intersection point of the y-axis gives the dry friction

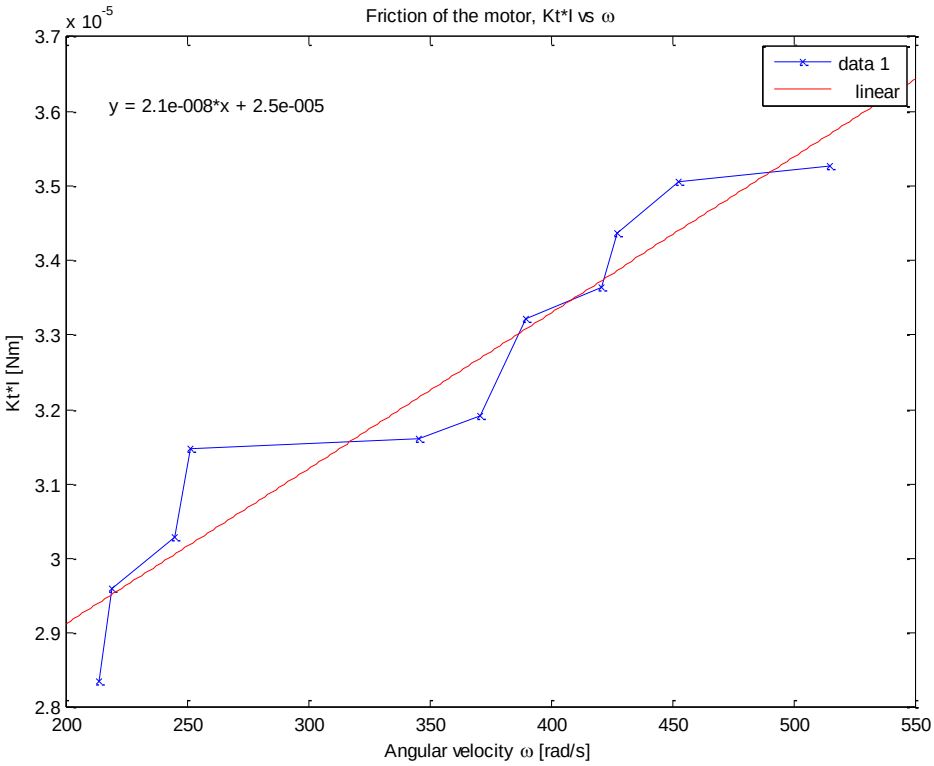

Figure A.7.3: Motor angular velocity vs. torque

The resultant viscous friction and dry friction components for the motor are:

Viscous friction: 2.0895e-008 [Nm.s]
Dry friction : 2.4939e-005 [Nm]

A.5 **Drive Friction**

Similar to motor friction, drive viscous and dry friction can be calculated via current and angular velocity measurements.

As before : $I \cdot K_t = B_m \cdot \omega_m(t) + \tau_{mdry}(t) + \tau_{mstic}(t)$

Velocity of the shaft with the gear train is estimated to be :

$$\omega_{mg}(t) = \frac{v_{input} - R \cdot I}{K_t}$$

Neglecting stiction, the slope of the plot $K_t \cdot I$ vs. $\omega(t)$ gives the viscous friction of the drive. The intersection point of the y-axis gives the dry friction

Table A.4. Measurements for the Drive Friction

Voltage [v]	Current [mA]	Ang. Velocity Gear Shaft [rad/s]	Ang. Velocity Motor shaft [rad/s]
2.44	40	4.5029	878.60
2.9	42	5.4454	1070.72
3.25	43	6.0737	1219.39
3.45	44	6.4926	1302.29
3.6	46	6.8068	1358.50
3.7	47	7.1209	1397.56

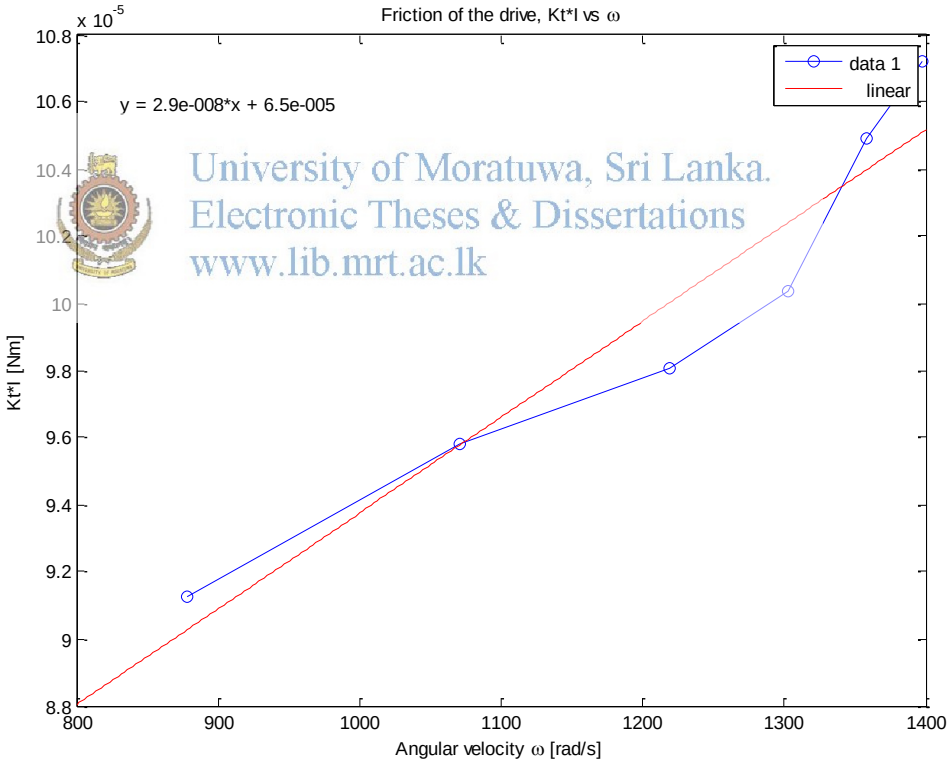

Figure A.7.4: Drive angular velocity vs. torque

The resultant viscous friction and dry friction components for the drive are:

Viscous friction: 2.8599e-008[Nm.s]
Dry friction : 6.5152e-005 [Nm]

A.6 **Motor Stiction**

Stiction torque is the starting friction imposed by the motor. By slowly increasing the torque applied to the motor shaft, it is possible to determine at what torque the motor begins to rotate. Stiction is calculated by measuring the current, at which the motor begins to rotate, and multiply by motor constant to get the torque.

The current at which the motor begins to turn : 70.5[mA];

The resultant stiction torque: 1.3586e-004 [Nm.]

A.7 **Drive Stiction**

Similar to motor stiction, starting friction of the drive, is calculated by ramping the input voltage as to increase input torque via the increasing current.

The current at which the motor begins to turn: 95.0[mA];

The resultant stiction torque: 1.9175e-004 [Nm.]

APPENDIX B. USE CASE

This section describes a simple use-case is described step-by-step. It is assumed that the Robotic manipulator and its software is setup, connected and the robotic manipulator is in the vicinity of the workspace.

1. Set to “ready pose” using the software.
2. In order to register the workspace; in the “Reg.” tab of the “Control Modes and Function; load the previously acquired model of the workspace.
3. Using “Joint/Cart.” Tabs in the Control Modes, move the manipulator to each registration point, select the corresponding registration point and its target in the “Reg.” tab. Click “Register” once the registration points are selected. Once the registration is complete. The device is ready for use.
4. In the “Path” Tab Paths can be loaded. The manipulator can traverse through a path using “Start/Stop” and “Pause” buttons

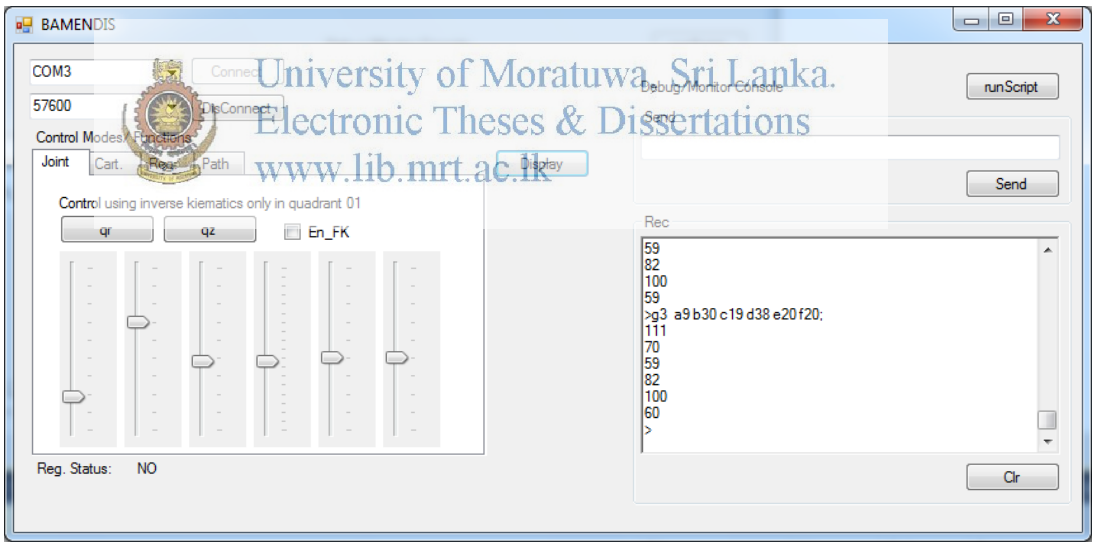

Figure A.7.5: Client software interface

APPENDIX C. LIST OF COMMANDS

C.1 **Command line Syntax**

G{CODE} a{value}1 **b{value}…h{value} ;**

C.2 **Commands**

1. Help

G0;

Command displays help message

2. Command base joints

G1 a{theta1} b{ theta2} c{theta2};

Commands the first three joints in degrees

3. Command upper joints

G2 a{theta1} b{ theta2} c{theta2};

Commands the last three joints in degrees

4. Command all joints

G3 a{theta1} b{ theta2} c{theta2} d{theta4} e{ theta5} f{theta6};

Commands the all joints in degrees

5. Command specific joint

G4 a{joint} b{ theta};

Command specific joint angle in degrees

6. Display calibration offsets

G5;

Command displays calibration offsets

7. Calibrate Write

G6 a{joint} b{value};

Writes the specific calibration value to the joint

APPENDIX D. LIST OF SIMULATION SCRIPTS

This is a list of simulation scripts, provided in the CD. The MATLAB files requires Robotic Toolbox v9.8

Model files

mdl_mbot.m	Model including tool
mdl_mbot_nt.m	Model without tool
fGenIK_c1.m	Inverse Kinematics Generator
fGenFK.m	Forward Kinematics Generator

Control Generation

My_pos_final_orig.m	Cascaded Controller, continuous, digital
disMeorig .mdl	Cascaded Controller in Simulink
disMeorigD.mdl	Cascaded Controller digital in Simulink
disMe.mdl	Position only with velocity feed forward
disMeDD.mdl	Position with velocity FF, in digital

Validators

S_hw_validation_1.m	Hardware validation of link movement
verify_IK_random	
myBot_dhMod_verif_toolbox	Verify Configuration using toolbox
myBot_dhMod_verif	Verify Manually
Dynamics.m	Calculate and verify Dynamic parameters
trajectory_algo_via_points.m	Joint space trajectory generator
ICP_validation	ICP_validation
AOQ_validation	AOQ_validation

Library generation

DoFK.m

doIK.m

GoMx.m

GoMy.m

GoMz.m

GoX.m

GoY.m

GoZ.m

probot.m mybot model, print

VB files

Vb project folder