

Fall 12-18-2015

Design, Manufacturing and Control of an Advanced High-Precision Robotic System for Microsurgery

Arezoo Ebrahimi

University of Maine, arezoo_ebrahimi@umit.maine.edu

Follow this and additional works at: <http://digitalcommons.library.umaine.edu/etd>



Part of the [Other Mechanical Engineering Commons](#)

Recommended Citation

Ebrahimi, Arezoo, "Design, Manufacturing and Control of an Advanced High-Precision Robotic System for Microsurgery" (2015).
Electronic Theses and Dissertations. 2378.
<http://digitalcommons.library.umaine.edu/etd/2378>

This Open-Access Thesis is brought to you for free and open access by DigitalCommons@UMaine. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of DigitalCommons@UMaine.

**DESIGN, MANUFACTURING AND CONTROL OF AN ADVANCED
HIGH-PRECISION ROBOTIC SYSTEM
FOR MICROSURGERY**

By

Arezoo Ebrahimi

BSIE Sharif University of Technology, 2006

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Mechanical Engineering)

The Graduate School

The University of Maine

December 2015

Advisory Committee:

Mohsen Shahinpoor, Professor of Mechanical Engineering, Advisor

Vincent Caccese, Professor of Mechanical Engineering

Xudong Zheng, Assistant Professor of Mechanical Engineering

THESIS ACCEPTANCE STATEMENT

On behalf of the Graduate Committee for Arezoo Ebrahimi I affirm that this manuscript is the final and accepted thesis. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

Dr. Mohsen Shahinpoor, Professor of Mechanical Engineering December 4, 2015

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of Maine, I agree that the library shall make it freely available for inspection. I further agree that permission for "fair use" copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature:

Date:

**DESIGN, MANUFACTURING AND CONTROL OF AN ADVANCED
HIGH-PRECISION ROBOTIC SYSTEM
FOR MICROSURGERY**

By Arezoo Ebrahimi

Thesis Advisor: Dr. Mohsen Shahinpoor

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Mechanical Engineering)
December 2015

Microsurgeries like ophthalmic surgery confront many challenges like limited workspace and hand motion, steady hand movements, manipulating delicate thin tissues, and holding the instrument in place for a long time. New developments in robotically-assisted surgery can highly benefits this field and facilitate those complicated surgeries. Robotic eye surgery can save time, reduce surgical complications and inspire more delicate surgical procedures that cannot be done currently by surgeon's hands.

In this thesis work, the requirements for ophthalmic surgeries were studied and based on that a robotic system with 6 DOF is proposed and designed. This robotic

system is capable of handling the position and orientation of the surgical instrument with theoretical accuracy of 10 μm . The design features a remote center of motion that defines the point of entry into the eye or patient's body. The forward and inverse kinematics equations and workspace analysis of the robot is also discussed and presented.

Six miniature DC motors with their PID controllers were installed on robot arms in order to run 6 DOF systems. Therefore, the dynamic behavior of a DC motor was studied and modeled and then the position and velocity transfer functions were derived and used to study the behavior of the system and also to manually tune the PID controller.

The function of different elements of the control system including encoder, controller modules, Controller Area Network (CAN) and the controller software were discussed as well. The graphical user interface called EPOS Studio and performs as the motion controller is introduced and the way it organizes communications among the elements of the control system was described.

ACKNOWLEDGEMENTS

The completion of this undertaking could not have been possible without the participation and assistance of so many people.

First of all, I would like to express my gratitude to my advisor *Prof. Mohsen Shahinpoor*, for his continuous support, patience, motivation and encouragement throughout my graduate studies.

Besides my advisor, I would like to thank the rest of my thesis committee: *Prof. Vincent Caccese* and *Dr. Xudong Zheng*, for their encouragement and useful comments.

Most of all, I would like to thank my family for their countless love and endless support and inspiration.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1 : INTRODUCTION	1
1.1 Robot Assisted Surgery	2
1.2 Eye Surgery	4
1.2.1 Cataract Surgery	6
1.2.2 Glaucoma Surgery	7
1.2.3 Refractive Surgery	8
1.2.4 Vitreo-Retinal Surgery	8
1.2.5 Eye Muscle Surgery	9
1.2.6 Cataract Surgery Procedure	9
1.3 General Design Requirements for Eye Surgical Robot.....	14
1.3.1 Safety and Accuracy Considerations.....	14
1.3.2 Required Performance and Degrees of Freedom	16
1.3.3 Required Velocities and Forces at the Tip.....	17
1.4 Current Robotic Ophthalmic Surgery Systems	18
1.4.1 Eye Surgery with the da Vinci®	18

1.4.2 EYE-RHAS (Eye Robot for Haptically Assisted Surgery).....	20
1.4.3 University of Tokyo	21
1.4.4 JPL RAMS (Robot-Assisted Micro-Surgery)	23
1.4.5 UCLA – IRISS.....	24
CHAPTER 2 : DESCRIPTION OF DESIGNED ROBOTIC SYSTEM.....	26
2.1 Joints Design and Configuration	26
2.1.1 Ring Mechanism	27
2.1.2 Double Rotor Mechanism.....	28
2.2 Axes of Motion	30
2.3 Motor Selection: Forces, Velocities and Precision	32
2.4 Forward Kinematics	36
2.5 Inverse Kinematics	41
2.6 Workspace.....	43
2.7 Accuracy.....	45
CHAPTER 3 : CONTROL THEORY OF THE SYSTEM.....	47
3.1 Introduction to Theory of Control	47
3.2 Closed Loop Transfer Function.....	48
3.3 Dynamic Model of a DC Motor	50
3.4 Motor (V) Transfer Function Analysis.....	55
3.4.1 Step and Impulse Responses	56

3.4.2 Frequency Domain Analysis	58
3.5 Proportional Controller	59
3.6 PID Controller	63
3.6.1 Transfer Function of PID Controller	65
3.6.2 Proportional Gain Effect.....	66
3.6.3 Integral Gain Effect.....	67
3.6.4 Derivative Gain Effect.....	68
3.6.5 PID Controller Gain Values	69
CHAPTER 4 : CONTROLLER ARCHITECTURE	70
4.1 Encoder.....	70
4.2 Controller Module	73
4.2.1 Maxon Controller for EC Brushless Motors	75
4.3 Controller Area Network (CAN).....	79
4.4 Easy Positioning System (EPOS) Controller Software	81
4.4.1 The Motion Controller.....	83
4.4.2 Position Accuracy.....	87
CHAPTER 5 : CONCLUSION.....	89
5.1 Conclusions.....	89
5.2 Recommendations for Future Work	91
BIBLIOGRAPHY	93

APPENDIX A: MATLAB CODE FOR FORWARD KINEMATICS	99
APPENDIX B: MATLAB CODE FOR INVERSE KINEMATICS.....	101
APPENDIX C: MATLAB CODE TO PLOT RESPONSES OF THE DC MOTOR SYSTEM	103
BIOGRAPHY OF THE AUTHOR.....	105

LIST OF TABLES

Table 2-1: Degrees of freedom.	27
Table 2-2: Actuators requirements.....	33
Table 2-3: Selected motor specifications.....	34
Table 2-4: Denavit-Hartenberg link parameters.....	39
Table 2-5: Misplacement variables.....	46
Table 4-1: DIP switch setting for robotic system with 6 nodes.....	77
Table 4-2: Data sheet to convert the motion of six axes to quad count (qc) unit	86

LIST OF FIGURES

Figure 1-1: Block diagram for standard master-slave robotic surgical system (Mahpeykar 2013).	3
Figure 1-2: Anatomy of the human eye (Biographicsmedia 2006).....	5
Figure 1-3: Anatomy of the eye - anterior and posterior parts.	6
Figure 1-4: Cataract eye (left), healthy eye (right).....	6
Figure 1-5: Effect of cataract on vision.....	7
Figure 1-6: Using speculum to hold eye open.....	10
Figure 1-7: Making incision (ORBIS Cyber-Sight 2006).	11
Figure 1-8: Viscoelastic injection (ORBIS Cyber-Sight 2006).....	11
Figure 1-9: Creating the main incision (ORBIS Cyber-Sight 2006)	12
Figure 1-10: Capsulorhexis.....	12
Figure 1-11: Hydro-dissection (ORBIS Cyber-Sight 2006) and (Robinson 2012)	13
Figure 1-12: Phacoemulsification.....	13
Figure 1-13: IOL insertion.....	14
Figure 1-14: Anterior and Posterior manipulation angles (β).....	16
Figure 1-15: Range of angles that an instrument enters the eye (α).....	17
Figure 1-16: Da Vinci® system for eye surgery.....	19
Figure 1-17: Eye-RHAS, eye robot for haptically assisted surgery (Meenink 2011)	21
Figure 1-18: EYE-Rhas, master console (Hendrix 2011)	21
Figure 1-19: Robotic slave system for vitreo-retinal surgery (Ueta, et al. 2009).....	22

Figure 3-8: The step and impulse responses of the velocity transfer function.	57
Figure 3-9: The position and velocity responses of the system to the voltage frequency.	58
Figure 3-10: A simple closed loop system with proportional controller.....	59
Figure 3-11: The proportional controller step response for $K_p=1$	61
Figure 3-12: The proportional controller step response for $K_p=5$	62
Figure 3-13: The proportional controller step response for $K_p=10$	62
Figure 3-14: The proportional controller step response for $K_p=50$	63
Figure 3-15: The block diagram of closed loop PID controller.	64
Figure 3-16: The PID controller step response for $K_p=1$ (left) and $K_p=50$ (right).	66
Figure 3-17: The PID controller step response for $K_p=500$	66
Figure 3-18: The PID controller step response for $K_i=1$	67
Figure 3-19: The PID controller step response for $K_i=100$	67
Figure 3-20: The PID controller step response for $K_d=0$ (left) and $K_d=1$ (right).	68
Figure 3-21: The PID controller step response for $K_d=2$	68
Figure 3-22: The best PID response.	69
Figure 4-1: The microsurgery robotic system with its motors and controllers connected through the USB port to the controller software.	70
Figure 4-2: Motor (I) encoder and its cable.....	71
Figure 4-3: The signals of an incremental encoder.	72
Figure 4-4: Physical layout of the six motor controllers CAN bus.	74
Figure 4-5: The EC brushless controller, interfaces and designations	75
Figure 4-6: JP1 – Numbering scheme.....	76

Figure 4-7: Controller (III) in operating status with the Green LED “ON” .	78
Figure 4-8: DC brushed controller, interfaces and designations	78
Figure 4-9: Motor/encoder connector (left) and DC brushed controllers (V) & (VI) wired to the system (right)	79
Figure 4-10: Schematic overview of the controller with external connections and communication.	81
Figure 4-11: The EPOS Studio screen with the Workspace tab open	82
Figure 4-12: Communication set up with six nodes	83
Figure 4-13: Different operating modes under the “Tools tab”	84
Figure 4-14: Data entry for 45 degree motion of node 1 (around the ring) in profile position mode	85

CHAPTER 1 : INTRODUCTION

Robotics is a rapidly growing field that has applications in diverse industries including manufacturing, automated production lines, welding, painting, spraying, assembly, packaging, product inspection, testing and many more. The advantages of employing robotic systems are due to their ability to produce more accurate and high quality work in a short amount of time which significantly reduces the cost of production and makes them the best fit to perform many industrial tasks.

A robot is a mechanical or artificial executor, usually an electro-mechanical machine that is guided by a computer program or electronic circuit which can be autonomous or semi-autonomous. Also there are some tasks that require such high level of accuracy and precision that cannot be achieved using human hands. This is one of the most promising attributes of robots and based on this feature medical application of robots is recently in demand.

Designing a robot for medical applications highly benefits hospitals, medicine, patients and doctors. It requires versatility and a specific design for certain applications. Medical robots can assist in two different areas of health care and surgeries.

Health care robots are designed to perform a variety of tasks to help patients, elderly or individuals while surgical robots are designed to assist surgeons perform very specific tasks with high accuracy and less trauma. The objective of these robots is to enhance surgical skills and to help surgeons to perform complicated

procedures more ergonomically and non-invasively with more precision and fewer efforts.

1.1 Robot Assisted Surgery

One of the fields that highly benefits from robotic advances is robotic surgery. Surgery performed by surgeons is quickly moving towards less invasive and more precise surgery to reduce hospitalization time and expedite recovery and healing.

Although improvement on manual procedures has yielded successively better results, these are merely marginal and do not reflect the real needs for effective techniques (Gheshmi 2012). Therefore, the need for utilizing new techniques in performing surgeries can be seen through robotic surgery (Patel 2008). There are some surgical procedures that cannot be done by hand or they are extremely difficult to perform, such as delicate eye surgeries or brain surgery in which tiny blood vessels are blocked or aneurysm or hemorrhage have occurred. Currently, surgeons practically use other indirect methods for those kinds of surgeries that cause more trauma and increase healing and hospitalization time and cost in some cases.

Robotic surgery is already a possibility for robotic microsurgery (Shahinpoor and Gheshmi 2014). There have been recent successes in endoscopic and laparoscopic techniques, visualization and surgical instruments which have enabled performing a number of traditionally manual surgeries through robotic surgery in less time and with less pain. For example, in laparoscopy, large incisions have been substituted by a few small incisions (up to five, 5-12mm in diameter) and therefore

there's no need to cut the body open anymore to reach inside the abdominal area. The robotic arms are equipped with end-effectors having grippers, haptic, optical and force sensors and other surgical tools which perform their functions and continuously transfer information to the surgeon.

The term “*robotically-assisted surgery*” is often used in literature (Ghani, et al. 2012) (Lambaudie, et al. 2010) to emphasize that the robot is fully under the control of the surgeon. A master-slave system is the common layout used in robotic assisted surgery which consists of master console, controller and slave robot. The slave system manipulates the surgical instruments and tools, while the surgeon does the surgery by controlling the master console. Figure 1-1 shows a block diagram of this type of master-slave surgical systems layout.

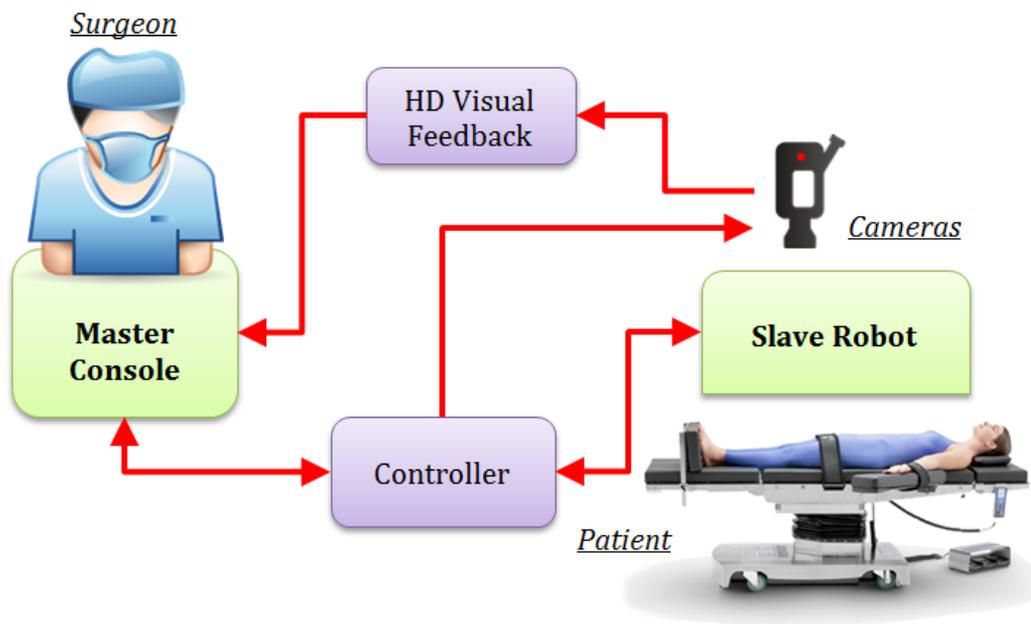


Figure 1-1: Block diagram for standard master-slave robotic surgical system (Mahpeykar 2013).

Tele-operation is one of the spectacular advantages of this system, enabling the surgeon to perform the surgery from a remote location. One of the commercially available surgical master-slave systems is the Da Vinci® System by (Intuitive Surgical 2013). Performing surgery using this system significantly reduces the trauma because of using long slender instruments (300mm long, 8.5 mm in diameter) that enter the human body via small incisions, often fitted with a trocar.

These instruments are manipulated outside the body and their movements inside the body are inverted and scaled by the instrument pivoting point (at the insertion site).

One of the other benefits of using a master-slave layout is scaling the surgeon's hand motion and filtering the tremor which will reduce the probability of unwanted collision between surgical tools with adjacent tissues in high precision surgeries and therefore reduces human error possibilities.

1.2 Eye Surgery

Eye surgery, also known as ocular surgery, is surgery performed on the eye or its adnexa, typically by an ophthalmologist (Encyclopedia of Surgery 2015). The eye is a fragile organ, and requires extreme care before, during, and after a surgical procedure. The anatomy of human eye is shown in Figure 1-2. An expert eye surgeon is responsible for selecting the appropriate surgical procedure for the patient, and for taking the necessary safety precautions.

Robot-assisted methods offer many advantages over conventional surgical techniques such as improved precision, reduced tremor, and amplified scale of

motion. One of the eye surgeries performed on the inner side of the eyeball is refractive cataract surgery. The other type is external eye surgery which is performed on the rectus muscles controlling the eye movement. Refractive surgery is often external, only affecting the cornea (LASIK¹ for example) or can be intraocular. Figure 1-3 shows the anterior and posterior sections of the eye. The majority of eye surgeries, almost 80% of cases, are performed on the anterior part of the eye.

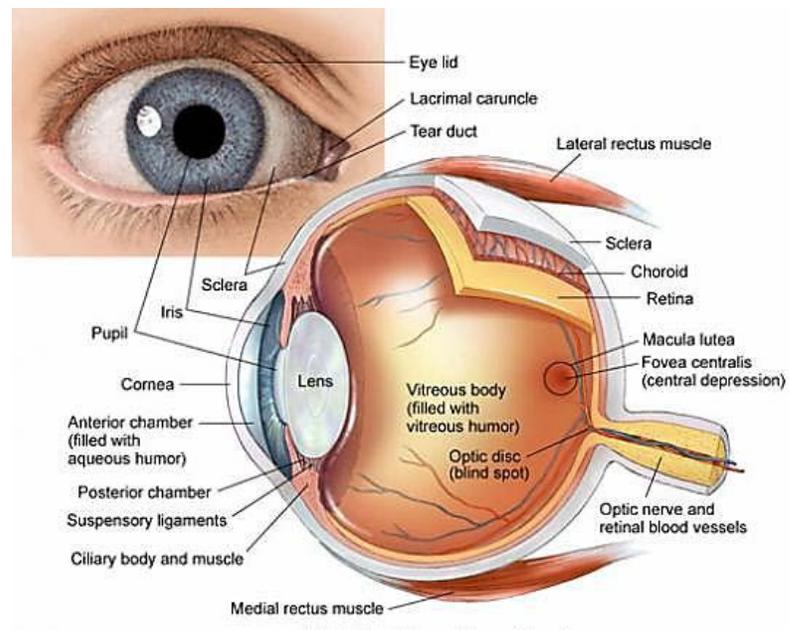


Figure 1-2: Anatomy of the human eye (Biographicsmedia 2006)

¹ Laser-assisted in situ keratomileusis

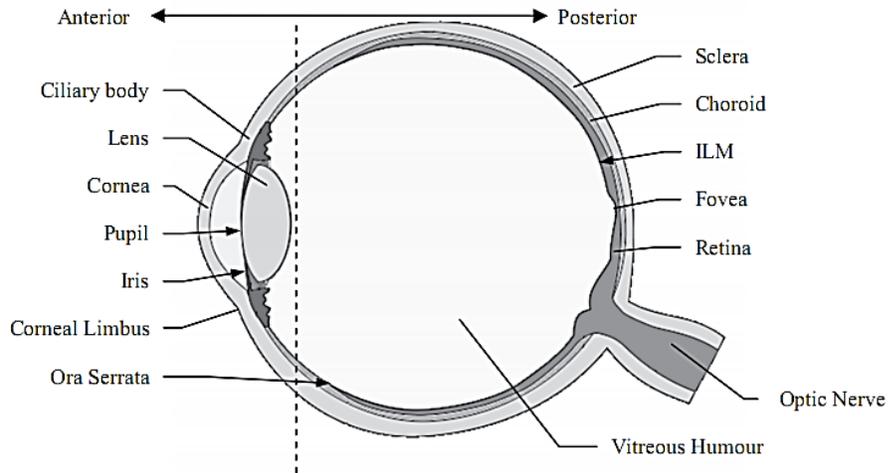


Figure 1-3: Anatomy of the eye - anterior and posterior parts.

1.2.1 Cataract Surgery

When the eye's naturally clear lens becomes opaque or clouded, it's called a cataract. Most cataracts are the result of the natural process of aging. Others may be present at birth or develop as a result of physical, drug, or chemical injury. Cataract surgery, one of the most common operations performed in the U.S., clears up the cloudiness. Figure 1-4 shows a cataractous eye lens compared to normal clear lens.

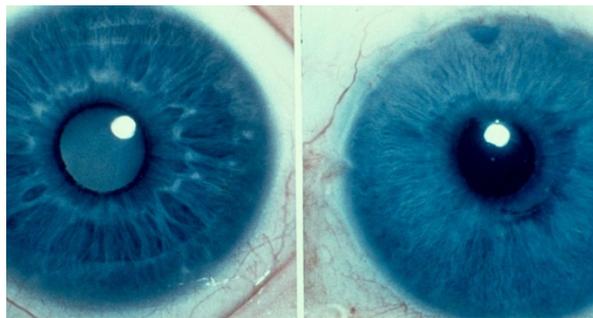


Figure 1-4: Cataract eye (left), healthy eye (right)



Figure 1-5: Effect of cataract on vision

Cataracts could be treated using eye drops in the early stages of development. However, surgical treatment is required if the cataract is fully developed. During cataract surgery, the cloudy lens is removed or cleaned out and replaced by a clear manmade lens called artificial intraocular lens (IOL). With over 1.8 million surgeries a year in the US, cataract surgery is the most common eye surgery in the United States (Goroll and Mulley 2012).

1.2.2 Glaucoma Surgery

The glaucoma disorder happens due to damage of optic nerve which is usually resulted from high pressure in the eye due to the aqueous humor. The reason of high pressure is unbalanced production and drainage of aqueous fluid inside the eye globe. Unfortunately the damage is permanent however there are some surgical

treatments to control the situation and prevent the eye from further damages. These surgical treatments involve with opening the blocked schlemm's canal and trabecular mesh tissues to let the aqueous humor fluid drain properly and reduce the intraocular pressure (IOP).

1.2.3 Refractive Surgery

A refractive problem is an issue of eye lens to focus parallel rays of light properly to produce a clear image on the retina. Any defect in the shape or curvature of the cornea, length of the eyeball or age of the lens can cause refractive errors. This problem can be temporarily corrected using eye glasses or contact lenses. To treat this disorder permanently refractive surgery known as LASIK is performed. In this surgical treatment the cornea is cut open to access the eye lens and then an excimer laser is used to treat the surface to the desired shape.

1.2.4 Vitreo-Retinal Surgery

Vitreo-retinal surgery refers to any ophthalmic surgery performed to treat problems related to the retina such as retinal detachment, vitrectomy and diabetic retinopathy.

The retina is responsible to generate electrical impulse of image and sends this information via the optic nerve to the brain. This image is created from focused light rays entering the eye through the cornea and the pupil. Even though light is needed to generate an image, it also can damage retinal structures when too much of it is focused onto the retina.

To fix the damaged retina, the vitrectomy surgery is performed by making three small incisions through sclera. The surgery is performed through an operating microscope. For more extensive work, other instruments such as microscissors or microforceps may also be introduced through these incisions to perform microsurgery near the retinal surface.

1.2.5 Eye Muscle Surgery

Extraocular muscle surgery is the third most common eye surgery in the United States (American Society of PeriAnesthesia Nurses 2012). Eye muscle surgery treats diseases related to the alignment of eyes such as strabismus. Eye muscles are strengthened, weakened or repositioned during this operation. To perform this, a small incision is made in the thin white tissue (conjunctiva) overlying the muscle. The muscle is then separated from the eye and reattached in a new position.

Ophthalmic surgery is not only limited to mentioned categories and includes more types such as oculoplastic surgery, corneal surgery, eye removal, and so on. In the next section, cataract surgery is discussed in detail as it is one of the target surgeries of the intended robotic system design. Design requirements are mostly derived from the procedures in cataract surgery as they are applicable to other ophthalmic surgeries as well.

1.2.6 Cataract Surgery Procedure

In general cataract surgery procedure involves with replacing the cataractous natural lens with an artificial intraocular lens (IOL). The most popular technique to

perform this surgery is Phacoemulsification. In this method an ultrasonic needle is used to emulsify the lens and extract out the particles through small incisions. Also there are other methods that remove the entire natural lens in one or few pieces which require bigger incisions that takes longer to heal (Mahpeykar 2013).

The phacoemulsification surgical procedure steps are as follows:

1) Anesthesia and Dilation:

To numb eye nerves and prevent blinking or eye movement during surgery a topical anesthetic eye drop is used followed by an injection. This injection dilates the pupil to provide a better access to eye lens.

2) Exposure of the eye ball:

The eye lids are held open using a speculum to prevent blinking during the surgery, Figure 1-6.

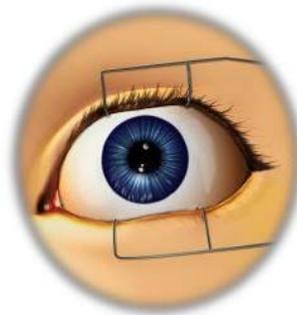


Figure 1-6: Using speculum to hold eye open.

3) Making incision:

A small incision (almost 1mm) is made to provide an entry to the anterior chamber of the eye for anesthetics and viscoelastic gel injection to the anterior chamber. It is also used to introduce a second instrument in the eye later.

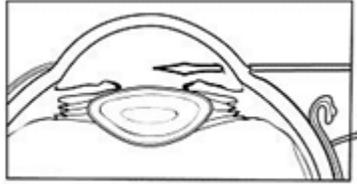


Figure 1-7: Making incision (ORBIS Cyber-Sight 2006).

4) Injection of viscoelastic gel

To stabilize and maintain the shape of the cornea and also pressurize the anterior chamber the viscoelastic gel is injected to the anterior chamber through the small incision made in previous step.

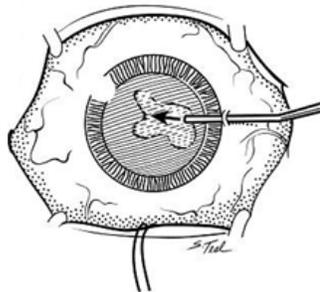


Figure 1-8: Viscoelastic injection (ORBIS Cyber-Sight 2006).

5) Main incision:

To introduce the main instrument, a 2 mm incision is made 90 degree from the previous smaller incision using diamond knife, shown in Figure 1-9.

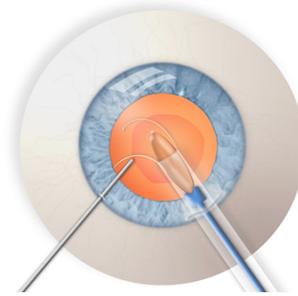


Figure 1-9: Creating the main incision (ORBIS Cyber-Sight 2006)

6) Capsulorhexis

In this step a portion of the lens capsule is cut open to access the lens. To create a round and smooth opening so that the lens can be emulsified the anterior capsule wall is scratched using a curved needle and then performing a curvilinear tearing using forceps shown in Figure 1-10.

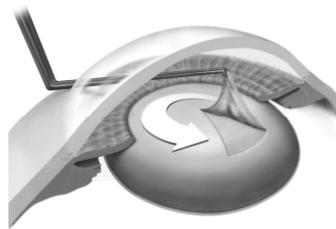


Figure 1-10: Capsulorhexis

7) Hydrodissection

To weaken the capsular-cortical connection so that the lens can be mobile, a cannula is filled with balanced salt solution (BSS) and is inserted through the small incision and injects a jet of BSS behind the lens, Figure 1-11. This is verified by rotating the lens in place to make sure it has been disconnected from the posterior capsule wall.



Figure 1-11: Hydro-dissection (ORBIS Cyber-Sight 2006) and (Robinson 2012)

8) Ultrasonic emulsification and irrigation aspiration

In this step a vibrating ultrasonic instrument with titanium or steel tip is inserted into the anterior chamber of the eye through the main incision. The cataractous lens is sculpted and emulsified by the tip of instrument while its particles aspirates through the tip, Figure 1-12.

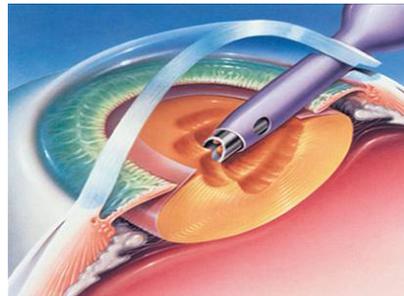


Figure 1-12: Phacoemulsification.

After clearing all the remnants additional viscoelastic gel is injected to fill and inflate the empty capsule.

9) Insertion of new lens

Now that the capsule is clean, a folded IOL, as shown in Figure 1-13, is inserted into the lens capsule via the main incision and then the IOL is unwrapped. This is also shown in Figure 1-13.

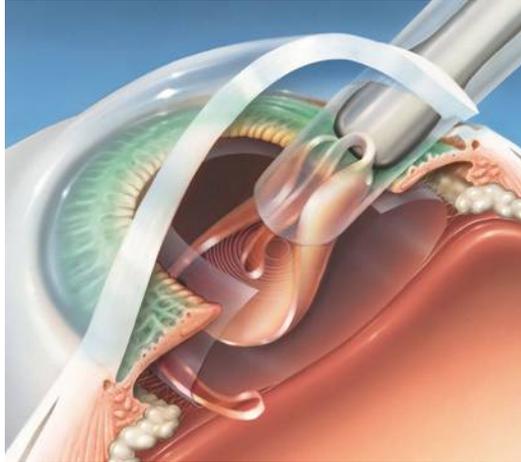


Figure 1-13: IOL insertion.

10) Viscoelastic removal

After the implantation of IOL, the viscoelastic gel is removed from the anterior chamber using an irrigation-aspiration device.

11) Wound sealing and pressure check

As the final step the surgeon seals the incisions by injecting some BSS to hydrate the incisions so that they do not leak and the pressure of the eye is also checked to make sure everything is okay.

1.3 General Design Requirements for Eye Surgical Robot

1.3.1 Safety and Accuracy Considerations

The designed robotic system must be able to deliver the required tasks while following the applicable safety requirements. Safety criteria main considerations mentioned are as follows (Taylor, et al. 1991). The robot must be fully under control and the surgeon must be in charge at all times. The instrument tip should stay

within a pre-specified positional envelop and never exert excessive force on the patient.

Also, the robot must be either allowed to fail to a safe state or it must continue to operate even in the presence of failure (Kazanzides 2009).

In order to provide these criteria the followings safety design requirements considered:

- 1) The instruments must be back-drivable.
- 2) To prevent unnecessary force to be applied on the patient in case of system malfunction, the active instrument must be mechanically leashed, meaning that the degrees of freedom manipulating the orientation and position of the instrument tip need to be rigid and consistent. This also leads to a minimally invasive surgery.
- 3) Motion of the surgical instrument must be very steady and accurate with virtually no shake or vibrations.
- 4) All joints need to hold their positions in case of power loss.

Eye surgery involves very thin organs. The thickness of the retinal membrane is only a few microns. The vessels attached to the retina are 50-150 μm in diameter. Therefore the accuracy of the robot must be less than 50 μm . Meaning, if the calculated accuracy of the system is above 50 μm , the robot is not suitable to perform such delicate procedures.

1.3.2 Required Performance and Degrees of Freedom

This surgical system is considered to be suitable to perform both anterior and posterior surgical procedures. For anterior surgical procedures the position of the instrument must be tangent to the cornea chamber, while in posterior surgical tasks it is more towards the inside of the eye as shown in Figure 1-14. Here the angle β is introduced as the angle between the tool axis and the optic axis of the eye and will be used to model the design.

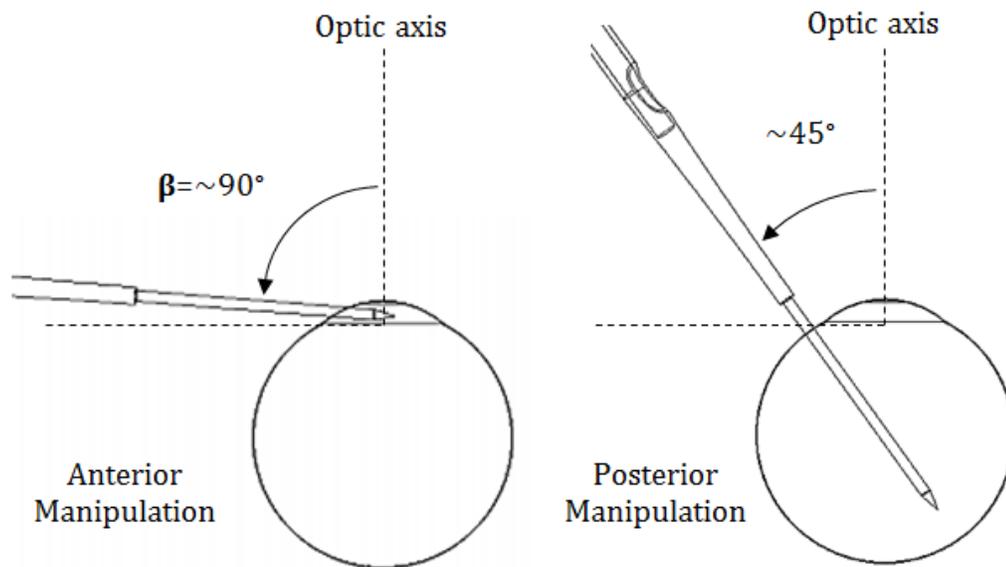


Figure 1-14: Anterior and Posterior manipulation angles (β)

The other performance criteria is the surgical instrument range of angles required in different eye surgery tasks like injections, capsulorhexis, vitrectomy, retinal manipulation and so on. This angle which is here called α is shown in Figure 1-15.

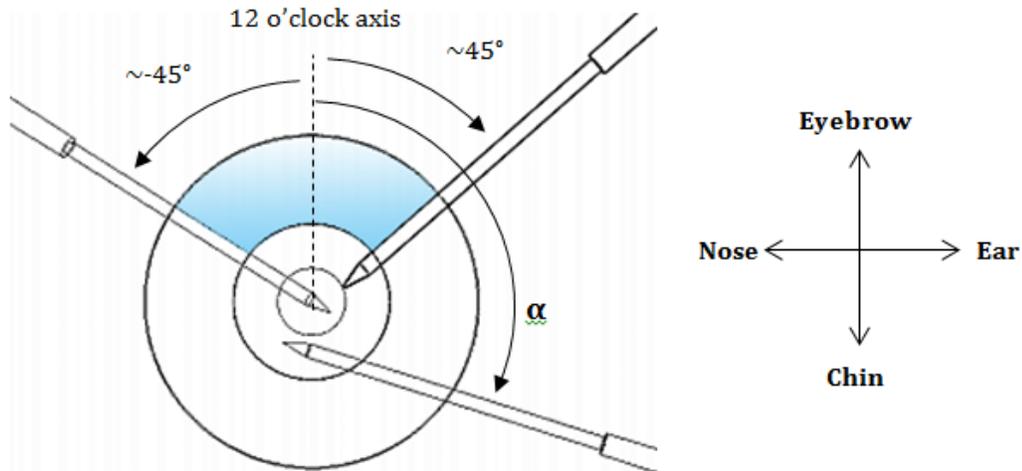


Figure 1-15: Range of angles that an instrument enters the eye (α)

Based on the procedure requirement, the range for α angle is -45° - 315° and for β angle is 0° - 90° .

To define the degrees of freedom for this robotic system different ophthalmic surgeries considered. For example, vitreo-retinal surgery demands 4 degrees of freedom (Meenink 2011).

For this device to be capable to perform different eye surgeries the robot must have 6 active degrees of freedom to control the position and orientation of the surgical instrument. Two of these are to relocate the entry point to the eye; two are required to orient the tool about this entry point; one is required to slide the instrument in and out of the eye; and one is required to rotate the tool about its axis.

1.3.3 Required Velocities and Forces at the Tip

The required velocity is usually low for microsurgeries to keep the accuracy high. Among different ophthalmic surgeries the maximum movement occurs in

cataract surgery during capsulorhexis step when the tip of the instruments move 10 mm in 0.4 second. Therefore, a maximum velocity of 25 mm/s is desired at the tip of the instrument.

To measure the forces required for eye surgery, vitreo-retinal surgery is considered since required higher forces compared to other types. This surgery involves with membrane peeling, vessel puncture and vessel dissection. An advanced instrument equipped with a tri-axial force sensor (X, Y, Z) was used (Jagtap and Riviere 2004) to measure required force for three mentioned procedures. On average their respective force in X/Y direction was: 55 mN, 25 mN and 20mN. In Z direction this was: 375 mN, 75 mN and 575 mN (RMS). Furthermore, in Z direction, the force range measured: 3140 mN for membrane peeling, 490 mN for vessel puncture and 5870 mN for vessel dissection (Meenink 2011).

1.4 Current Robotic Ophthalmic Surgery Systems

Robotic eye surgery is particularly attractive due to its advantages such as precision, accuracy, stability, amplified scale of motion, multitasking, automation, association of imaging systems, reduced tremor and tele-operation. More recent robotic eye surgery studies are briefly discussed below.

1.4.1 Eye Surgery with the da Vinci®

Da Vinci Surgical System (Intuitive Surgical, Sunnyvale, CA) is the most well-known platform in robotic surgery. A feasibility study of eye surgery with the da Vinci® system was done by (Bourla, et al. 2008). In this study, modified robotic instruments used to perform different intraocular surgeries such as vitrectomy, removing intraocular foreign body, and anterior capsulorhexis with the da Vinci system on porcine eyes. In this study the dexterity, maneuverability, and visualization necessary for intraocular surgery were examined.

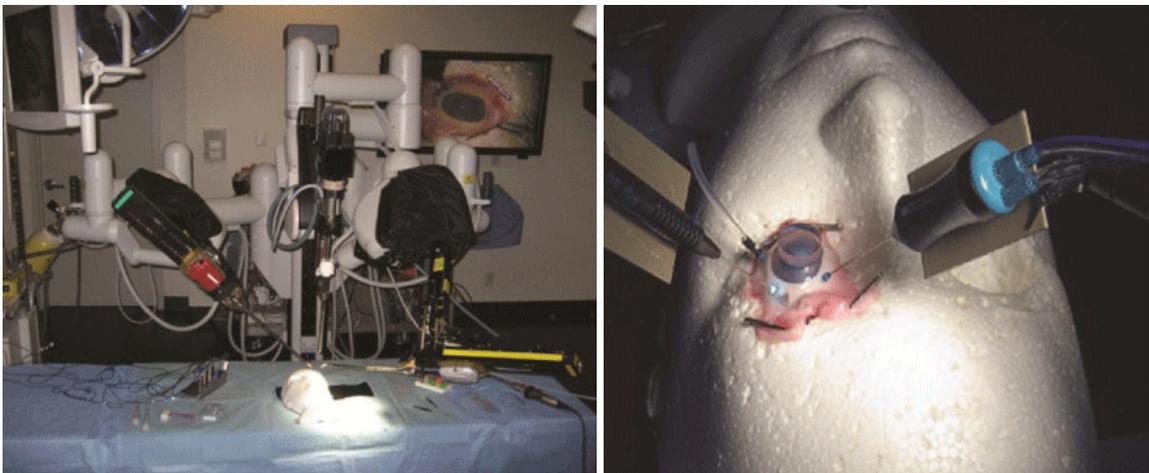


Figure 1-16: Da Vinci® system for eye surgery.

Instruments are adapted for the da Vinci® by getting them stuck onto a metal plate, Figure 1-16 (Pitcher, et al. 2012). The plates are grasped by standard robotic forceps. 3D endoscope of the da Vinci® provides visual feedback.

This study shows robotic instruments gives a full range of movement. The dexterity of the robotic arms is excellent, with steady instrument motion. However, controlling the robotic arms was not as intuitive as moving the wrist. A high stable point of rotation results in motion-related stress at the site of instrument insertion. Visualization of the external operative field during intraocular procedures required

camera realignment, which here inferior visual feedback compared with an ophthalmic microscope made anterior segment surgery hard to perform.

1.4.2 EYE-RHAS (Eye Robot for Haptically Assisted Surgery)

A master-slave system for vitreo-retinal eye surgery was developed at Eindhoven University of Technology (TU/e) by (Meenink 2011) and (Hendrix 2011). This is a two arm robotic system with 4 degrees of freedom each arm and a fixed remote center of motion and an additional degree of freedom to actuate the gripper motion of the instrument. The arms can be mounted on the patient's table. The master console as shown in Figure 1-18 is two joysticks controls by the surgeon. The surgeon's movements scales down and tremor filters out the by the controller. This system is not capable to relocate the entry point to the eye during the surgery and therefore is not suitable for cataract surgery.

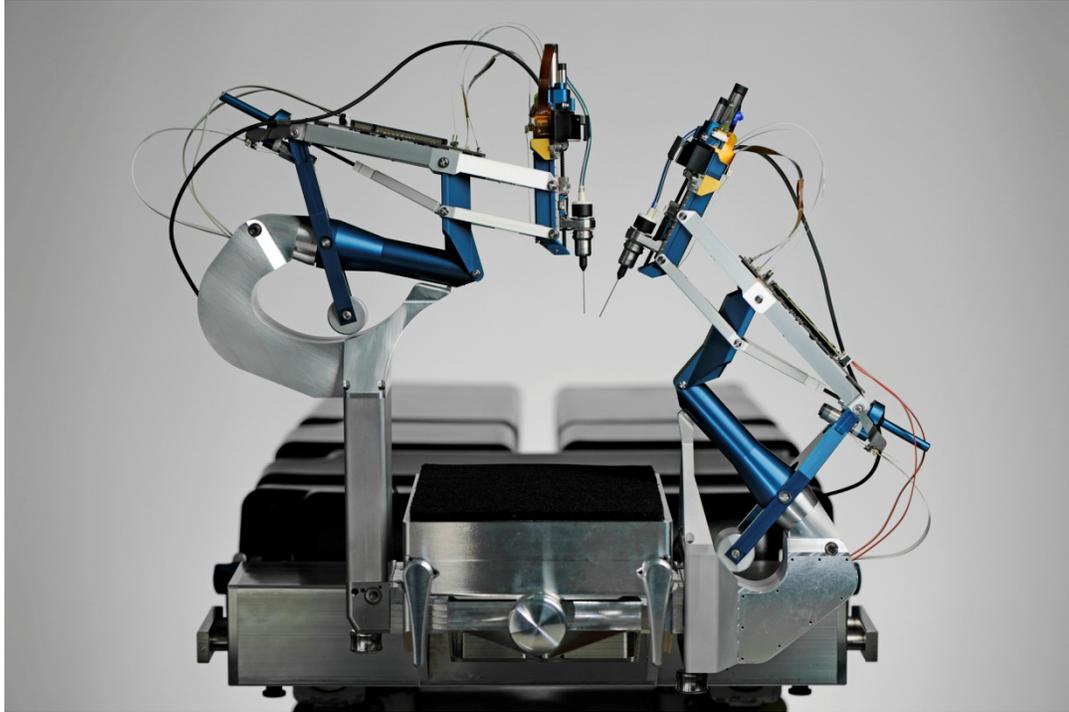


Figure 1-17: Eye-RHAS, eye robot for haptically assisted surgery (Meenink 2011)

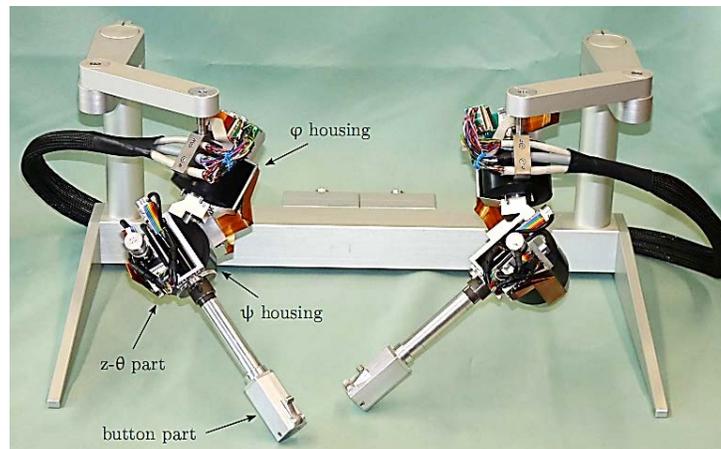


Figure 1-18: EYE-Rhas, master console (Hendrix 2011)

1.4.3 University of Tokyo

Another slave robot is developed at University of Tokyo for vitreo-retinal surgery (Ueta, et al. 2009). This robot has five degree of freedom, shown in Figure 1-

19. The system uses master console with a generic seven degree of freedom featuring 3D vision system, shown in Figure 1-20. Instrument axis has a motion range of $\pm 50^\circ$ rotation and -29 to 32 mm axial penetration, and 360° for rotation about tool's axis is realized with this slave robot.

The system features a fixed remote center of motion. The accuracy for this slave system is 5-10 times more than manually performed surgery.

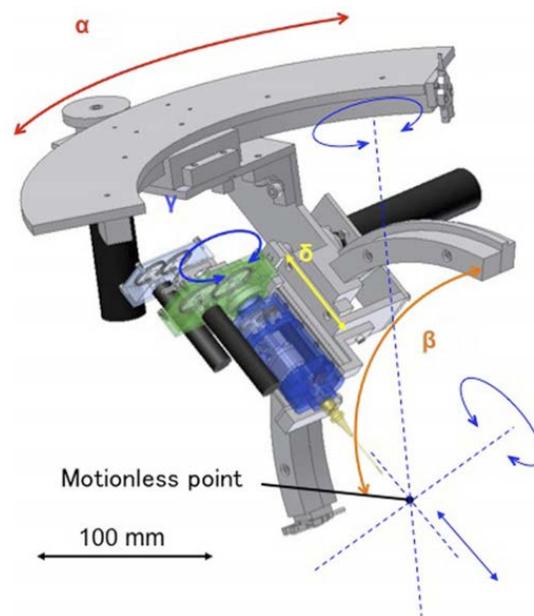


Figure 1-19: Robotic slave system for vitreo-retinal surgery (Ueta, et al. 2009)

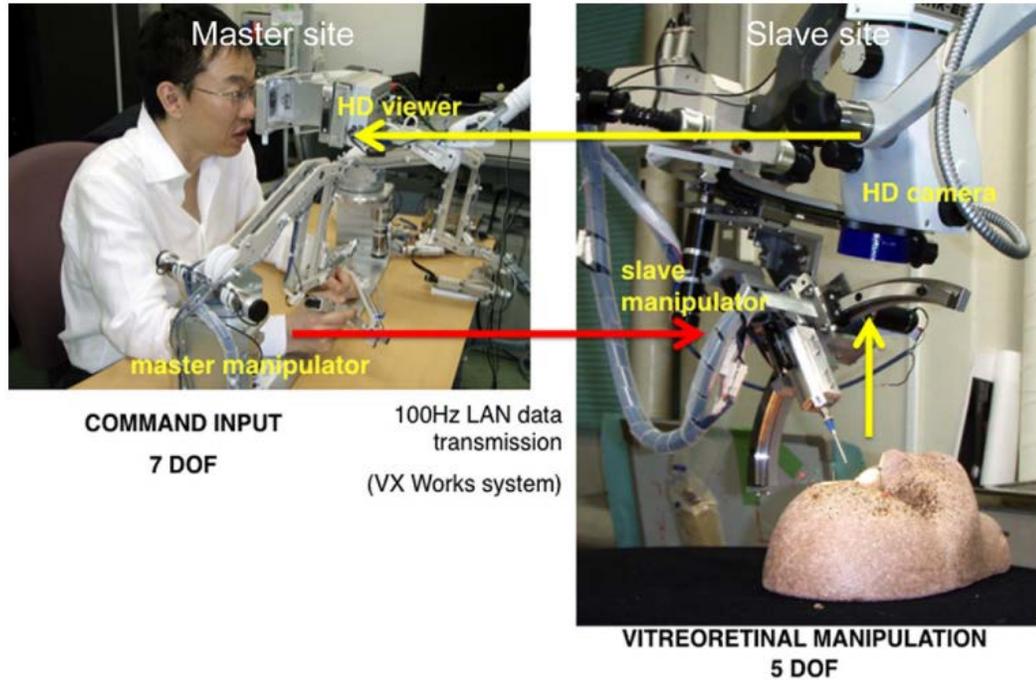


Figure 1-20: Robotic master-slave system (Ueta, et al. 2009)

1.4.4 JPL RAMS (Robot-Assisted Micro-Surgery)

A cable-driven telerobotic (master-slave) platform for eye surgery was introduced at NASA's Jet Propulsion Lab, California (Charles, et al. 1997). To improve precision and amplify force feedback the system provides scale-down human-input motions and tremor filtering.

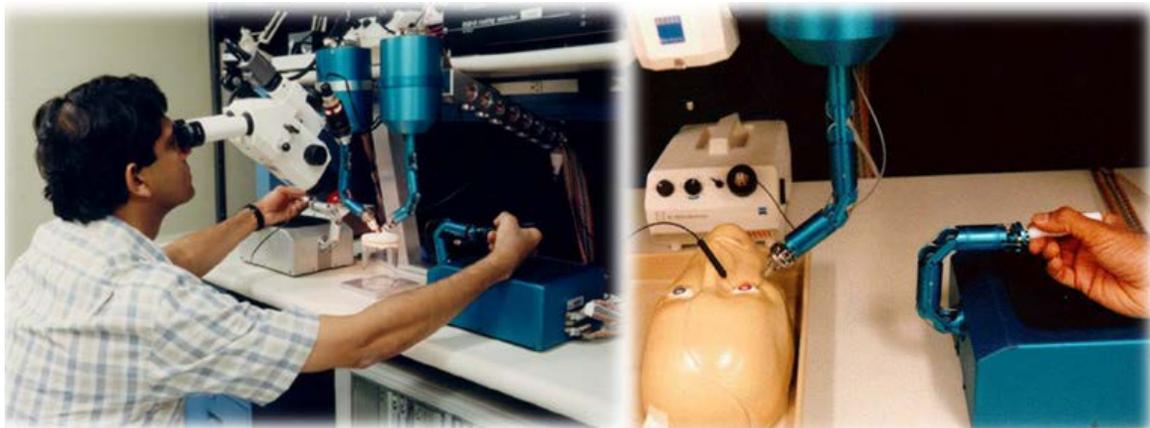


Figure 1-21: JPL Robot-Assisted Micro-Surgery (RAMS) system.

The slave robot has two compact arms with 2.5 cm diameter and 6 DOF, shown Figure 1-21. All joints are cable-driven and are connected to a cylindrical $\varnothing 120$ mm base at the top that contains the motors. The accuracy of surgical instrument is 12 μm (Das, et al. 1999).

The master arms are very similar to the slave arms and have the same dimensions with lower gear ratios to provide back-drivability. According to (Pitcher, et al. 2012) the complexity of the control software and the lack of mechanical remote center of motion were the main limitations of this system.

1.4.5 UCLA - IRISS

The Intraocular Robotic Interventional Surgical System (IRISS) is the result of research collaboration between Jules Stein Eye Institute and the UCLA Department of Mechanical and Aerospace Engineering. This system is designed as a microsurgical platform capable of performing complete ophthalmic procedures (Rahimy, et al. 2013). The IRISS system features a head-mounted "True Vision" (True Vision Displays, Inc., Cerritos, CA) stereoscopic visualization system, two joystick controls with tremor filtration and scaled motion, appropriately sized custom designed arms to accommodate different surgical instruments. The IRISS has the unique capability to manipulate two surgical instruments simultaneously through ocular incisions spaced millimeters apart.

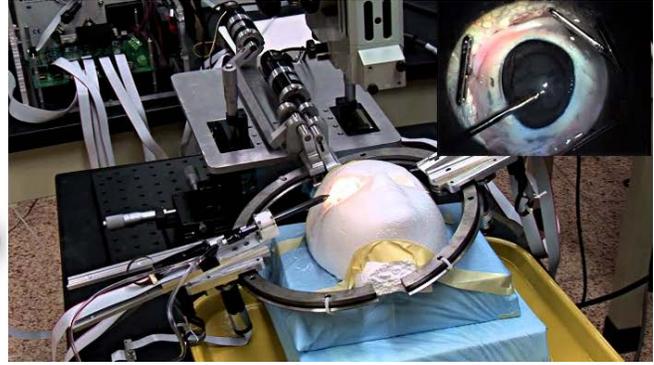
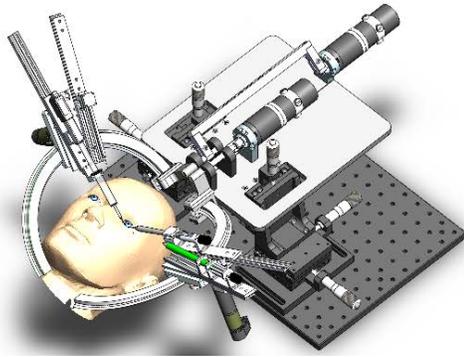


Figure 1-22: UCLA's IRISS robotic eye surgery system.

The slave manipulator, shown in Figure 1-22, can hold two surgical instruments simultaneously. Two motors drive the curved guides on a XY bed and have perpendicular axes with the curved guides. The instrument is then mounted on a carrier that slides over the curved guide. The carrier provides the axial motion of the instrument to the eye by means of a linear actuator.

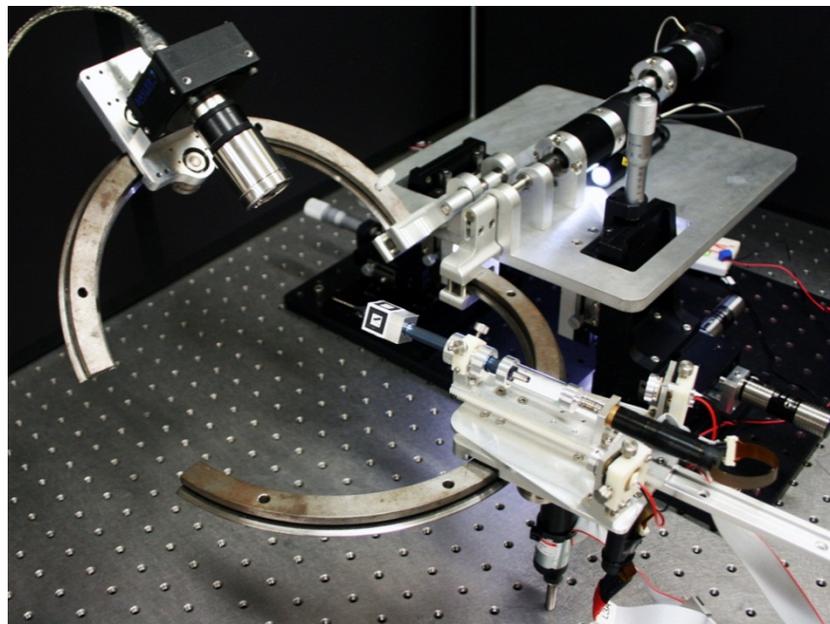


Figure 1-23: IRISS slave robot design.

CHAPTER 2 : DESCRIPTION OF DESIGNED ROBOTIC SYSTEM

To design the surgical instrument for eye surgery, the requirements discussed in Chapter 1 are considered. Based on those requirements, the instrument manipulator must have 6 degrees of freedom which two of them are reserved to relocate the eye entry point. The first one must provide 45° to 315° peripheral manipulation range and the second one is to provide 0° to 90° vertical manipulation range. The entry point to the eye must be fixed, while the surgical instrument must have at least 4 degrees of freedom about the entry point. Also the instrument must be capable of 30 mm penetration inside the eye and exerting push and pull forces up to 5.8 N. The design must be compact and easy to install and use with accuracy of less than 50 μm (Mahpeykar 2013).

2.1 Joints Design and Configuration

In this design the entry point is mechanically fixed. The advantages of this method over any other method which fixes the entry point by software is its higher accuracy and safety since it requires less actuation and is less influenced by calculation errors (Ida, et al. 2012).

This approach uses the *Remote Center of Motion* (RCM) concept. This RCM point is the intersection point for all degree of freedom axes as well as the entry point to the eye. These degrees of freedom are listed in Table 2-1:

Table 2-1: Degrees of freedom.

Axis	DOF	Description
1	φ	Ring motion
2	R	Overhead linear joint
3	θ_1	Double-rotor primary axis
4	θ_2	Double-rotor secondary axis
5	Z	Instrument insertion from entry point to the eye
6	ψ	Rotation of the tool about its own axis

To relocate the entry point a 2 DOF planar mechanism is required. Since the instruments enter the eye from the periphery, a polar ring system seems to be the best choice. This mechanism can control an angle and a radius to reach to a point.

The mechanism chosen to provide 4 degrees of freedom from the entry point is the double rotor mechanism due to its simplicity, slender design and low number of moving parts. The configuration and design of ring and double rotor mechanisms is described in next sub-sections.

2.1.1 Ring Mechanism

Ring mechanism is designed to generate a rotary motion about the eye and its 350 mm diameter provides required workspace for the optical devices above the patient's head. To manipulate the radial position, a linear joint is used. The configuration is shown in Figure 2-1.

The ring itself serves as the surgical head-master and holds the system together. The linear joint acts like an overhead crane positioning the double-rotor RCM in the radial direction. These two in combination provide a rigid platform for controlling the RCM.

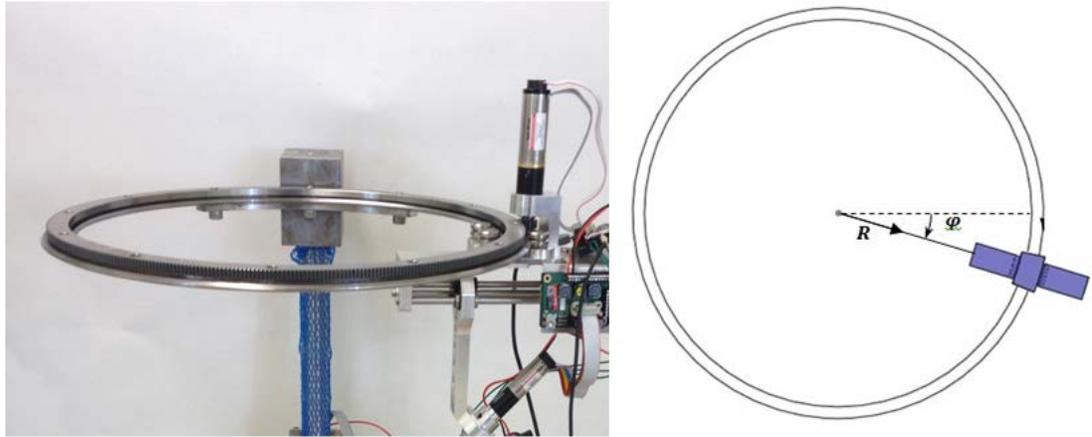


Figure 2-1: Ring and overhead crane mechanism

A carriage assembly featuring V-groove bearings is then mounted on the ring slide. The carriage has a motor driven pinion that is meshed with the external gear teeth of the ring slide.

2.1.2 Double Rotor Mechanism

The double-rotor mechanism consists of two rotors and three axes holding constant angle of 22.5° between them by the designed linkages as seen in Figure 2-2. This is a right-angle conic mechanism which cone's apex is the RCM point.

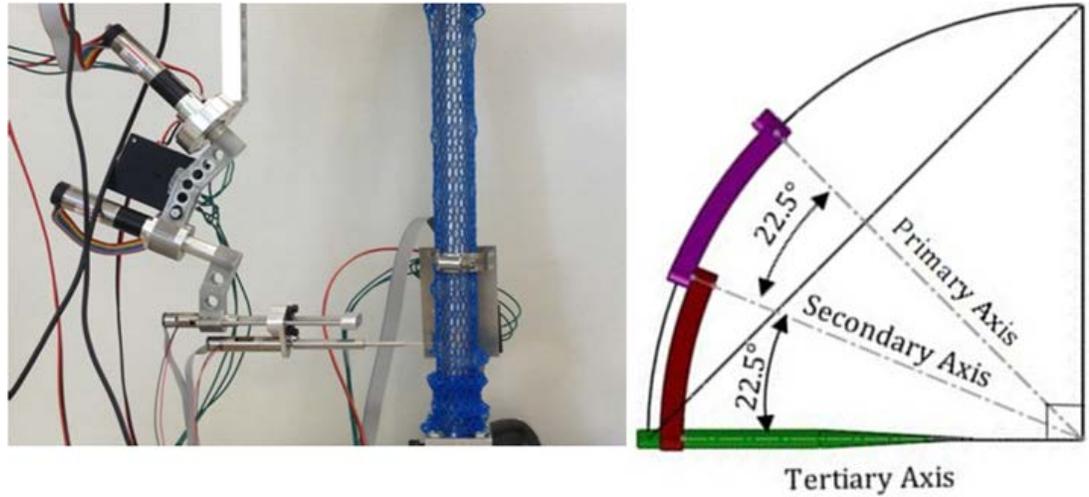


Figure 2-2: Double-rotor mechanism

Here, the rotations about the primary and secondary axes of the double-rotor mechanism are called θ_1 and θ_2 respectively. A linear joint with a 45° angle with the ring axis connects the double-rotor mechanism to the ring.

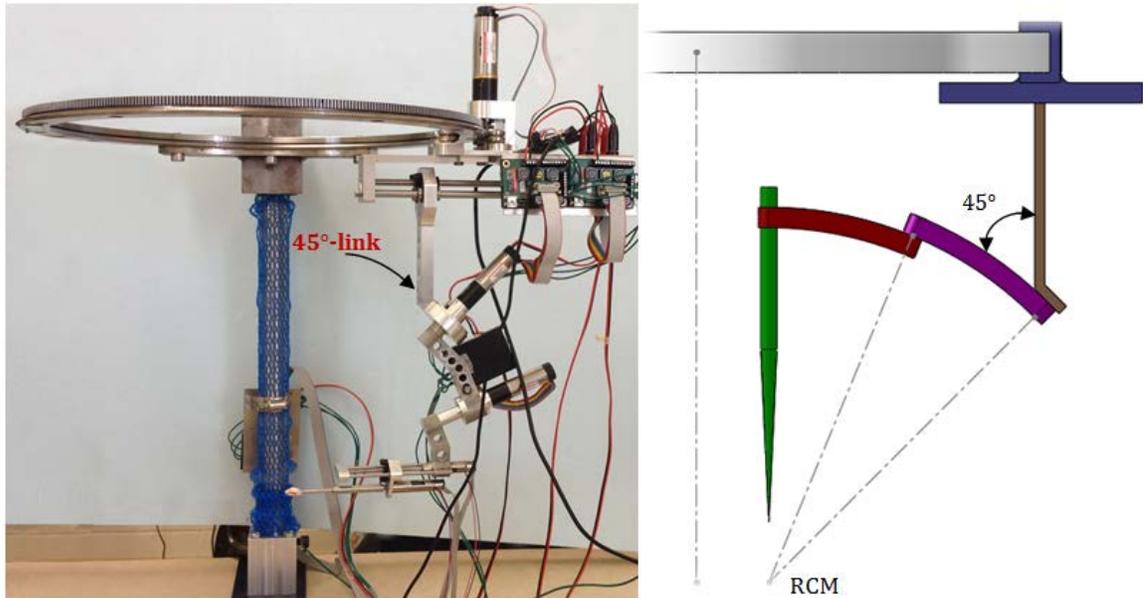


Figure 2-3: The linear 45° joint.

2.2 Axes of Motion

The ring motion is defined as **Axis-I** of this robotic system with range of 315° since the other 45° of the ring is used to mount it on place. The lead screw linear joint bolted to the bottom of the ring forms the **Axis-II** of motion. Precision acme lead screw with anti-backlash nut is used to provide a rigid positioning mechanism. See Figure 2-4. This lead screw mechanism accommodates a range of linear motion up to 90 mm.

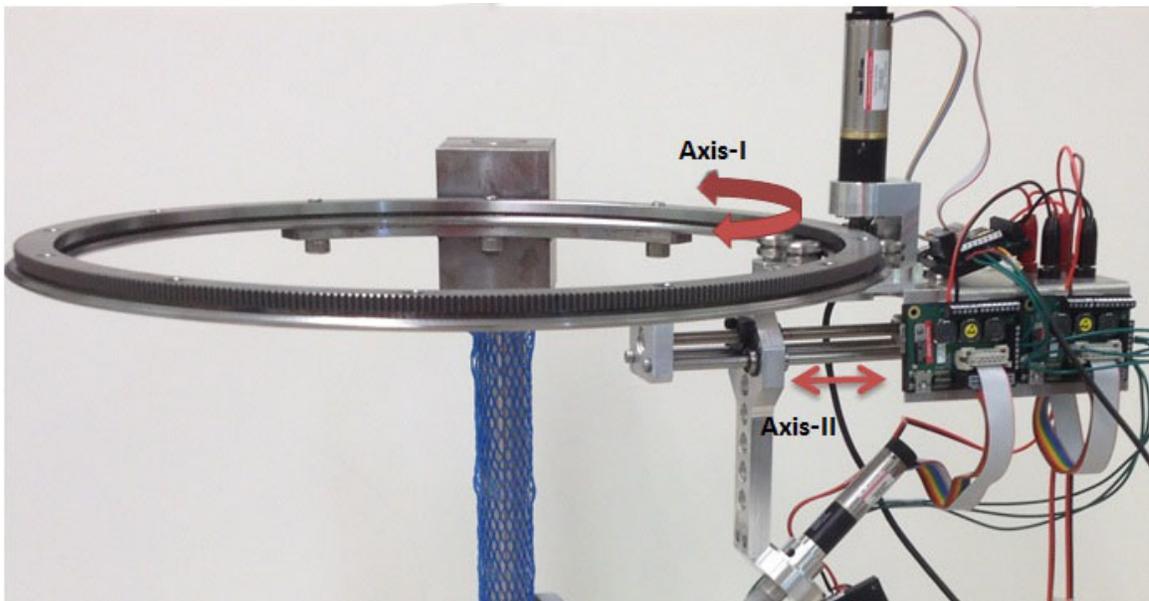


Figure 2-4: Ring and linear motion, Axis-I & II.

The 45° -link which is installed to the anti-backlash nut of the linear joint from one side and is connected to the double-rotor mechanism to the other side forms the **Axis-III** of the system which is also the primary axis of the double rotor.

The secondary axis of the double rotor is **Axis-IV** of the system and has the same design as Axis-III. The first and secondary axes of the double rotor both accommodate 360° rotation, see Figure 2-5.

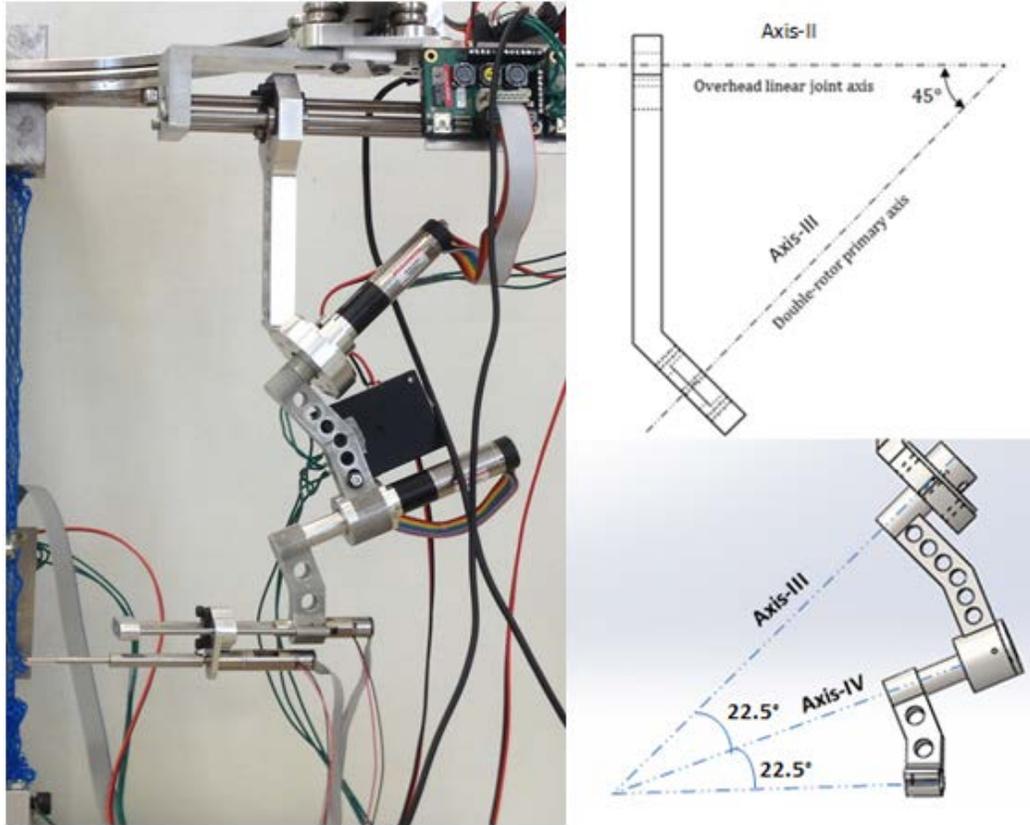


Figure 2-5: Double rotor joints, Axis III & IV.

A linear joint connected to the end of double rotor mechanism forms **Axis-V**. This axis is parallel to surgical tool's axis and has basically the same design as Axis-II. To position the surgical instrument a custom made nut with two linear bearings on the sides and an anti-backlash nut in the middle is used. This linear joint accommodates 60 mm penetration for surgical instrument. **Axis-6** is the axis of instrument rotation about itself which provides a range of 360° rotation of surgical device. This is shown in Figure 2-6.

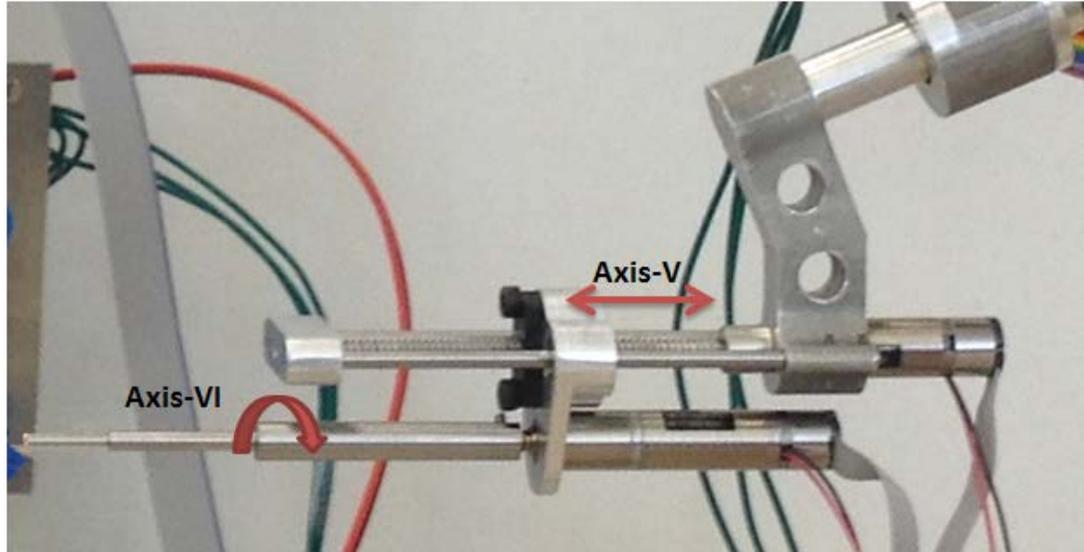


Figure 2-6: Axis-V and VI, instrument carrier

In this design motors are direct drive and placed on the joints to eliminate unnecessary backlash in the system and delivers more efficiency and accuracy. However, this increases the moving mass of the system and causes some wiring complications but since light miniature motors are readily available and can be used here, the benefits overweight the drawbacks.

2.3 Motor Selection: Forces, Velocities and Precision

To properly select motors for this design required force and velocity for each axis of the robot listed in Table 2-2 base on simple calculation and estimations considering the weight of links and surgical forces (Mahpeykar 2013).

The selected motor specifications are listed in Table 2-3.

Table 2-2: Actuators requirements.

Axis	Joint Type	Mechanism	Max. Continuous and intermittent Force/Torque	Max. Cont. Velocity Required	Cont. Power 100% drive efficiency	Desired Position Accuracy
Axis I	Ring Motion	Ring Gear/Pinion Drive	230 N.mm 400 N.mm	540 rpm	12.65 W	3°
Axis II	Linear	Precision Lead Screw	7 N 20 N	25 mm/s	0.5 W	50 μm
Axis III	Rotary	Direct Geared motor	200 N.mm 320 N.mm	100 rpm	2 W	0.5°
Axis IV	Rotary	Direct Geared motor	80 N.mm 130 N.mm	100 rpm	0.8 W	0.5°
Axis V	Linear	Precision Lead Screw	5 N 8.5 N	25 mm/s	0.2 W	50 μm
Axis VI	Rotary	Direct Geared motor	20 N.mm 35 N.mm	200 rpm	0.5 W	1.5°

Table 2-3: Selected motor specifications.

Motor	Shaft Diameter	Power	Voltage	Encoder Pulse	Gear box
Axis I	22 mm	25 W	24 V	512	19
Axis II	22 mm	25 W	24 V	512	1
Axis III	16 mm	8 W	12 V	512	84
Axis IV	16 mm	8 W	12 V	512	19
Axis V	10 mm	2.7 W	12 V	512	1
Axis VI	10 mm	2.7 W	12 V	128	16

This system has two linear joints with lead screw and anti-backlash nut design to provide required positioning accuracy. For the Axis II, a ¼" screw with 1.27 mm lead (0.05") is used. Therefore, the linear precision of *Axis II* can be calculated as follows:

$$\text{Axis II Precision} = \frac{1.27 \text{ mm/turn}}{512 \text{ CPT}} \times 1000 \frac{\mu}{\text{mm}} = \mathbf{2.48 \mu\text{m}}$$

For Axis V, a 9/64" screw with lead of 1.22 mm (0.048") is chosen. Then the precision would be:

$$\text{Axis V Precision} = \frac{1.22 \text{ mm/turn}}{512 \text{ CPT}} \times 1000 \frac{\mu}{\text{mm}} = \mathbf{2.38 \mu\text{m}}$$

For *Axis I*, a pinion used to mesh with the ring gear has 19 teeth and the ring gear has 456 teeth and a ratio of 1:24 is between actuator shaft and the ring motion. The angular precision, assuming zero-backlash, would be:

$$\text{Axis I Precision} = \frac{1}{512} \left(\frac{\text{encoder}}{\text{motor}} \right) \times \frac{1}{19} \left(\frac{\text{motor}}{\text{gearhead}} \right) \times \frac{1}{24} \left(\frac{\text{Pinion}}{\text{Ring}} \right) \times \frac{360^\circ}{1} = \mathbf{0.001^\circ}$$

For *Axis III*, the angular precision would be:

$$\text{Axis III Precision} = \frac{1}{512} \times \frac{1}{84} \times \frac{360^\circ}{1} = \mathbf{0.008^\circ}$$

The same configuration with a 19:1 gearhead instead, is used for *Axis IV*. Therefore the precision would be:

$$\text{Axis IV Precision} = \frac{1}{512} \times \frac{1}{19} \times \frac{360^\circ}{1} = \mathbf{0.037^\circ}$$

For *Axis VI*, the angular position precision would be:

$$\text{Axis VI Precision} = \frac{1}{128} \times \frac{1}{16} \times \frac{360^\circ}{1} = \mathbf{0.17^\circ}$$

Brushless motors deliver smooth operation, back drivability, compact design and high power-to-size ratio and torque.

2.4 Forward Kinematics

Now that all the link frames at each joint are defined, kinematic relations between joints can be describe.

To study forward kinematics of this robotic system, different general methods can be adopted. One common method is called *Denavit–Hartenberg method (D-H)*. In this method the geometry of each connection (link) in the mechanism is defined by 4 parameters namely: length, twist, offset and angle. Link parameters for a general connection are illustrated in Figure 2-7. After assigning coordinate frames to each link these parameters can be defined as (Mahpeykar 2013):

a_{i-1} (length): the distance from Z_{i-1} axis to Z_i axis

α_{i-1} (twist): the angle from Z_{i-1} axis to Z_i axis measured about X_i axis

d_i (offset): the distance between X_i and X_{i-1} measured along Z_i axis

θ_i (angle): the angle from X_{i-1} to X_i measured about Z_i axis.

The transformation matrices can be found using these parameters.

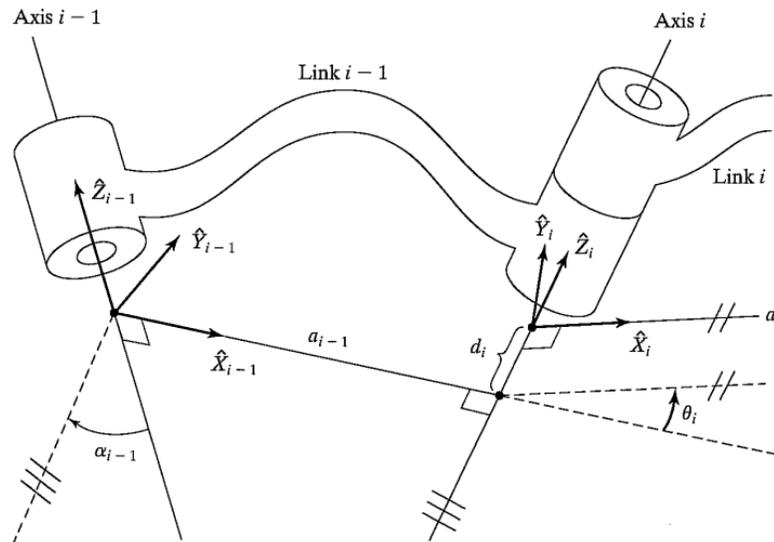


Figure 2-7: Link parameters: length, twist, offset and angle (Craig 2004)

A 4x4 matrix will be resulted from forward kinematics analysis. This matrix describes the position and orientation of the $\{n^{th}\}$ link frame with respect to the base frame $\{0\}$ and is a function of joint variables (θ_i) , as presented in Eq. 2-1. Coordinates of the end-effector's X-axis, Y-axis and Z-axis are described in first three columns which together describe the orientation of tool frame relative to base frame. The position of the end-effector's frame is described in fourth column.

$${}^0T_{n4 \times 4}(\theta_1, \theta_2, \theta_3, \dots, \theta_n) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-1}$$

In next step joint's axes need to be defined. \mathbf{Z}_1 represents the axis of the ring motion corresponding to $\boldsymbol{\varphi}$ angle. \mathbf{Z}_2 is the axis of the first linear joint, \mathbf{R} position, corresponding to the radial movement of the RCM. \mathbf{Z}_3 is the axis of the first rotary joint of the double-rotor mechanism, $\boldsymbol{\theta}_1$ angle. \mathbf{Z}_4 is the axis of the second rotary joint of the double-rotor mechanism, $\boldsymbol{\theta}_2$ angle. \mathbf{Z}_5 is the axis of the second linear joint, corresponding to the linear motion of the surgical instrument, \mathbf{Z} position. \mathbf{Z}_6 is the tool axis and rotation about itself, $\boldsymbol{\psi}$ angle. These joint's axes are shown in Figure 2-8.

Then coordinate frames must be assigned to each link, same way shown in Figure 2-7. These coordinate frames are attached to the robot links as illustrated in Figure 2-9, with the frames of the double-rotor mechanism shown in detail on the right side of the figure.

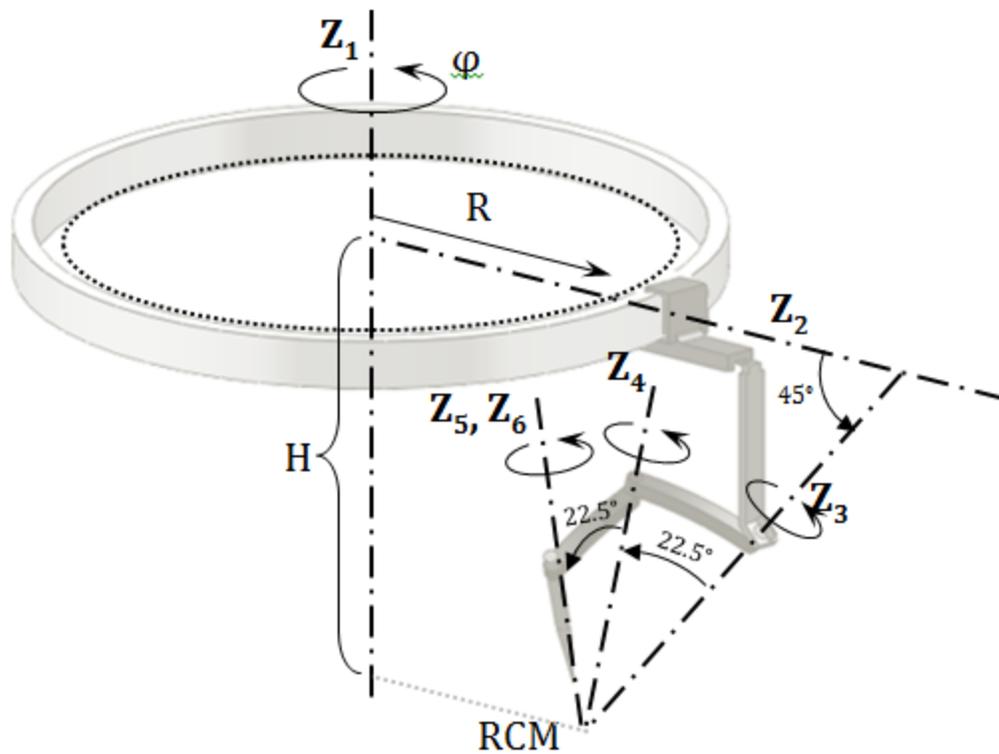


Figure 2-8: Defining joints axes

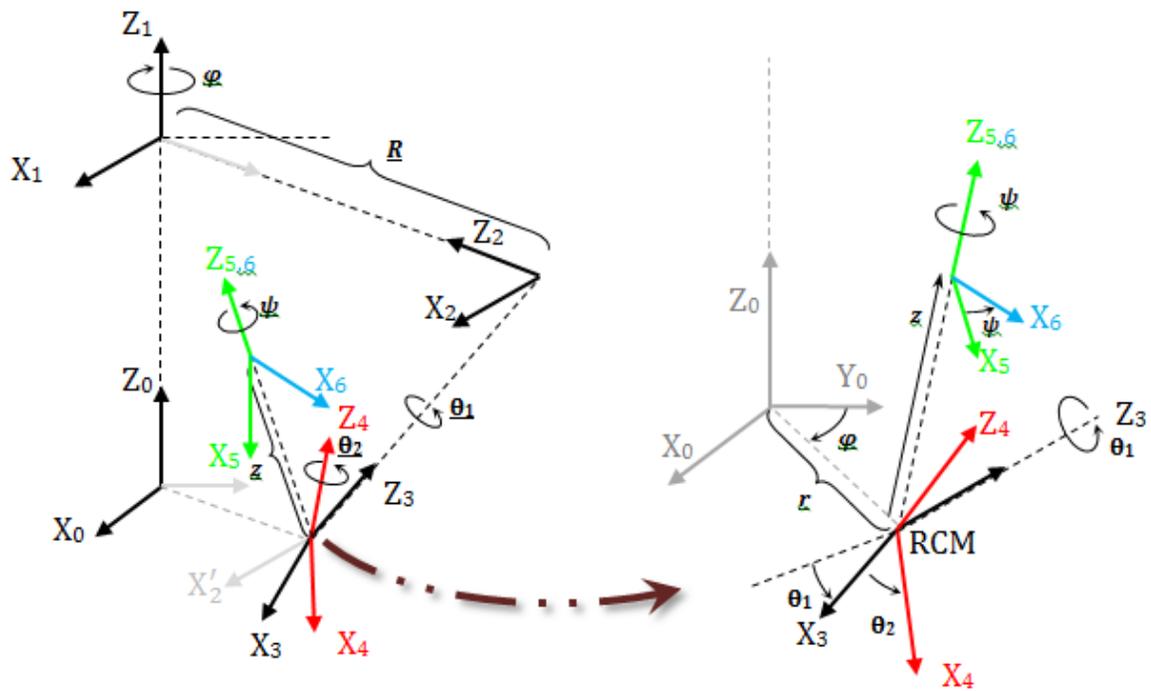


Figure 2-9: Link coordinate frames assignment.

Link parameters are identified and listed in Table 2-4. The ring motion angle is shown as $-\varphi$ since the positive direction for φ is chosen to be clockwise looking down from the top of the ring.

Table 2-4: Denavit-Hartenberg link parameters

$\{i\}$	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	H	$-\varphi$
2	0	90°	$-R$	0
3	0	-135°	$-H\sqrt{2}$	θ_1
4	0	22.5°	0	θ_2
5	0	22.5°	Z	0
6	0	0°	0	ψ

All distances between axes a_{i-1} are zero in Table 2-4, which means that each axis intersects with its neighboring axis. This in fact simplifies the kinematic equations. H is defined as the vertical distance between the base frame and the ring frame with the following relation applies:

$$R - H\sqrt{2} = r \quad \text{Eq. 2-2}$$

Where, r is the distance between the RCM and the base frame and is solely controlled by the first linear actuator.

The general transformation matrix from link i to $i - 1$ with link parameters a_{i-1} , α_{i-1} , d_i and θ_i is presented in Eq. 2-3 below:

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i \cdot s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-3}$$

Using the data from Table 2-4 with Eq. 2-3 the following transformation matrices can be found:

$${}^0_1T = \begin{bmatrix} c\varphi & s\varphi & 0 & 0 \\ -s\varphi & c\varphi & 0 & 0 \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-4}$$

$${}^1_2T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & \mathbf{R} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-5}$$

$${}^2_3T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ -s\theta_1/\sqrt{2} & -c\theta_1/\sqrt{2} & 1/\sqrt{2} & -H \\ -s\theta_1/\sqrt{2} & -c\theta_1/\sqrt{2} & -1/\sqrt{2} & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-6}$$

$${}^3_4T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ s\theta_2 c22.5 & c\theta_2 c22.5 & -s22.5 & 0 \\ s\theta_2 s22.5 & c\theta_2 s22.5 & c22.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-7}$$

$${}^4_5T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c22.5 & -s22.5 & -\mathbf{Z}s22.5 \\ 0 & s22.5 & c22.5 & \mathbf{Z}c22.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-8}$$

$${}^5_6T = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-9}$$

To find the end effector coordination in origin frame all these transformation matrices must be multiplied:

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} & \mathbf{p}_x \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} & \mathbf{p}_y \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} & \mathbf{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2-10}$$

Elements of Eq. 2-10 are quite lengthy and tedious to work with. A MATLAB code is presented in "Appendix A" to numerically and parametrically calculate the final matrix in terms of $\varphi, r, \theta_1, \theta_2, \psi$ and z .

For example, for following random configuration the result of forward kinematics equation is:

$$\varphi = \theta_1 = \theta_2 = \psi = 0.5 \text{ radians}$$

$$R = Z = 1$$

$${}^0_6T = \begin{bmatrix} 0.5992 & -0.6263 & 0.4987 & 0.9781 \\ 0.7534 & 0.6518 & -0.0868 & 0.7908 \\ -0.2707 & 0.4277 & 0.8624 & 0.8624 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

2.5 Inverse Kinematics

Although a completely general manipulator with 6 degrees of freedom does not have a closed-form solution, certain special cases can be solved. For instance in this design all axes have been chosen in a way that most of the parameters are zero. Besides, a closed-form solution is always reachable when 3 consecutive axes intersect at one point, based on Pieper's work on 6 axis robots (Pieper and Roth 1969) (D. Pieper 1968). However, this work was mostly done on all revolute joints. In this case a custom method is needed to solve the inverse kinematics since this manipulator has two prismatic joints.

To approach an inverse kinematics problem two methods can be applied; algebraic and geometric method. In algebraic approach kinematic equations must be put into a form for which a solution is known. In geometric approach the spatial

geometry of the robot must be decomposed into several sub-problems to find the solution for each degree of freedom one by one.

Forward kinematic equations of the proposed robot contain 12 kinematics equations each of which is quite lengthy. For example, the first element r_{11} in the forward kinematic equations, is of the form:

$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} r_{11} = \sin \varphi & \left((\sin \theta_2 (.271 \cos \theta_1 + .653) + .383 \sin \theta_1 (.271 \cos \theta_1 - .653)(\cos \theta_2 - 1))(\cos \psi \cos \theta_1 \right. \\ & - .707 \sin \psi \sin \theta_1) \\ & + (.383 \sin \theta_1 \sin \theta_2 \\ & - (.271 \cos \theta_1 + .653)(.271 \cos \theta_1 - .653)(\cos \theta_2 - 1)) \left(\sin \psi \left(\frac{\cos \theta_1 - 1}{2} \right) \right. \\ & \left. + .707 \cos \psi \sin \theta_1 \right) \\ & + \left(\sin \psi \left(\frac{\cos \theta_1 + 1}{2} \right) + .707 \cos \psi \sin \theta_1 \right) ((.271 \cos \theta_1 - .653)^2 \\ & - \cos \theta_2 ((.271 \cos \theta_1 - .653)^2 - 1)) \left. \right) \\ & - \cos \varphi \left((\sin \theta_2 (.271 \cos \theta_1 + .653) \right. \\ & - .383 \sin \theta_1 (.271 \cos \theta_1 - .653)(\cos \theta_2 - 1)) \left(\sin \psi \left(\frac{\cos \theta_1 + 1}{2} \right) + .707 \cos \psi \sin \theta_1 \right) \\ & - \left(\sin \psi \left(\frac{\cos \theta_1 - 1}{2} \right) + .707 \cos \psi \sin \theta_1 \right) (\sin \theta_2 (.271 \cos \theta_1 - .653) \\ & + .383 \sin \theta_1 (.271 \cos \theta_1 + .653)(\cos \theta_2 - 1) + (\cos \psi \cos \theta_1 - .707 \sin \psi \sin \theta_1) \\ & \left. - \cos \theta_2 (.146 \sin^2 \theta_1 - 1) - .146 \sin^2 \theta_1 \right) \end{aligned}$$

There are 11 more equations like this to try to solve for the unknown angles and prismatic variables. It is pretty obvious that there is no way to solve the inverse

kinematics of this manipulator using these types of trigonometric equations.

However, we can take advantage of the manipulator's geometry and break the inverse kinematics into a few sub-problems to find each joint parameter. This can be done using the key points and axes described in the forward kinematics section.

In The approach first the key point of the mechanism which is the RCM needs to be found. Then by having the RCM coordinate which can be directly linked to other parameters of the design, the problem simplifies to a planar geometric problem to solve for φ and R . In next step, Z which is the distance from the tool's frame origin to the RCM point can be calculated. The only unknowns at this step would be θ_1 , θ_2 and ψ . These parameters can be found using the concept of double-rotor mechanism. A MATLAB code is available in the "Appendix B" that calculates the inverse kinematics of this robot.

2.6 Workspace

The required workspace of this ophthalmic surgery robot is a spherical space with 30 mm in diameter which is the size of the eyeball that must be reachable by its end-effector. The reachable workspace of the robot is the 3D sub-space that the tip of the robot can reach with at least one orientation. The shape of reachable workspace for this robot is found by putting the upper limit and lower limit values of each degree of freedom into forward kinematics equation which is resulted in a shape illustrated in Figure 2-10.

Combination of the planetary ring motion and the top linear motion as the first and second axes gives a workspace in shape of planar disc with its center located on

the surgical base plane and is coincident with the ring axis. This disc shapes the locus of RCM.

The double-rotor mechanism (Axes III to VI) creates a conic workspace with two portions based on the position of instrument tip. When tip of instrument is above RCM which means Z value is positive the workspace is the upper right conic and when it is below RCM and Z value is negative it is the lower left cone. This is shown in Figure 2-11. Therefore the total reachable workspace can be found by moving the conic workspace over this disc, which gives us the mushroom shape in Figure 2-10.

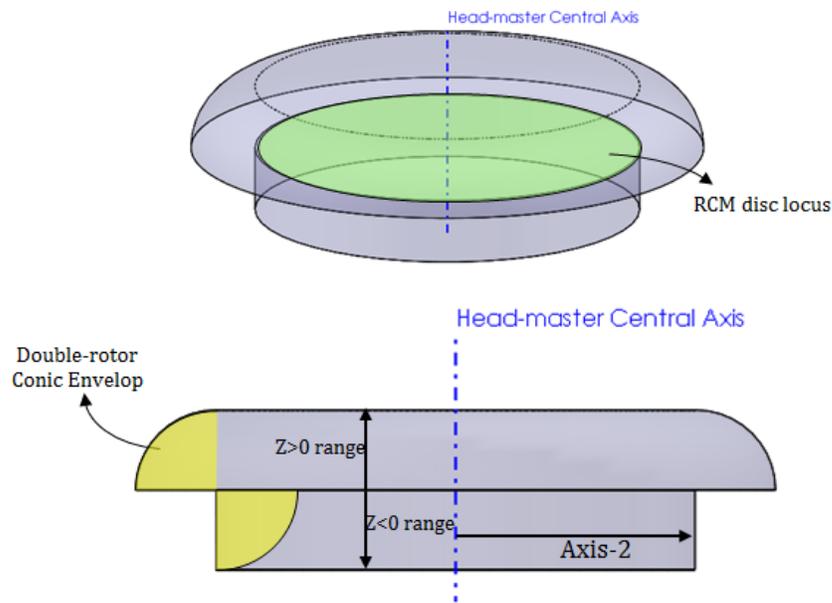


Figure 2-10: Reachable workspace of the robot.

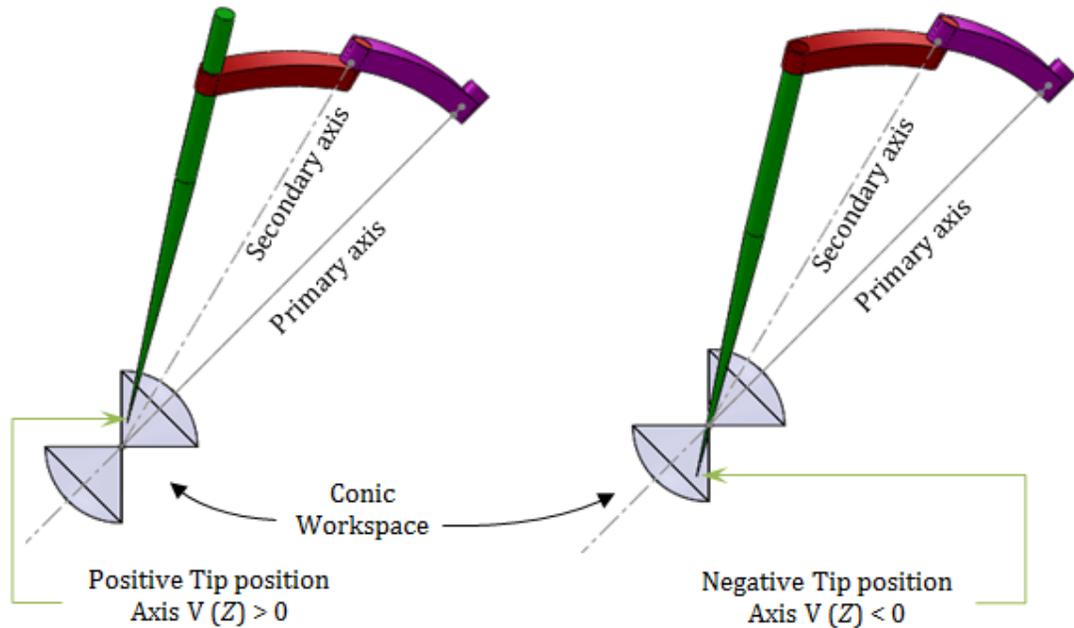


Figure 2-11: Double-rotor conic workspace; positive and negative sections.

2.7 Accuracy

To find the integrated precision of this robotic system the precision of each axis must be used which are calculated in section 2.3. Now that the forward kinematics of the robot is available, the total accuracy of the system can be discussed.

To calculate the precision at the instrument tip, an initial home position is considered. Then each DOF is misplaced by the amount equal to the precision of each axis in a non-compensating direction, meaning that the misplacement in one axis does not make up the error of the misplacement in another axis. Using forward kinematics equation the final position and orientation of the tool tip are compared:

Table 2-5: Misplacement variables

State	Joint variables
Home Position	$\varphi = \theta_1 = \theta_2 = 0$ $\psi = r = Z = 0$
Misplaced	$\varphi = 0.001^\circ$ $\theta_1 = 0.008^\circ$ $\theta_2 = 0.037^\circ$ $\psi = 0.17^\circ$ $r = 0.0025mm$ $Z = 0.0095mm$

$$Home = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Misplaced = \begin{bmatrix} 0.9999933 & -0.0036449 & 0.00034586 & 0.00000333 \\ 0.0036449 & 0.99999336 & 0.00000009 & 0.00250000 \\ -0.000346 & 0.00000116 & 0.99999994 & 0.00949999 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The total tip misplacement is then:

$$\Delta R = \sqrt{0.00000333^2 + 0.0025^2 + 0.00949999^2} = 0.00982344 \text{ mm} = 9.82 \mu\text{m}$$

CHAPTER 3 : CONTROL THEORY OF THE SYSTEM

3.1 Introduction to Theory of Control

Consider a system that receives an input $u(t)$ and in response produces an output $y(t)$. The response $y(t)$ is determined by the nature of the input signal $u(t)$ and by the nature of the system.



Figure 3-1: A simple input/output system.

If $g(t)$ is assumed to be the response of the system to a unit impulse at time $t=0$, then the response of the system $y(t)$ to any input $u(t)$ can be visualized as the response to a train of impulses that approximates the function (see Figure 3-2).

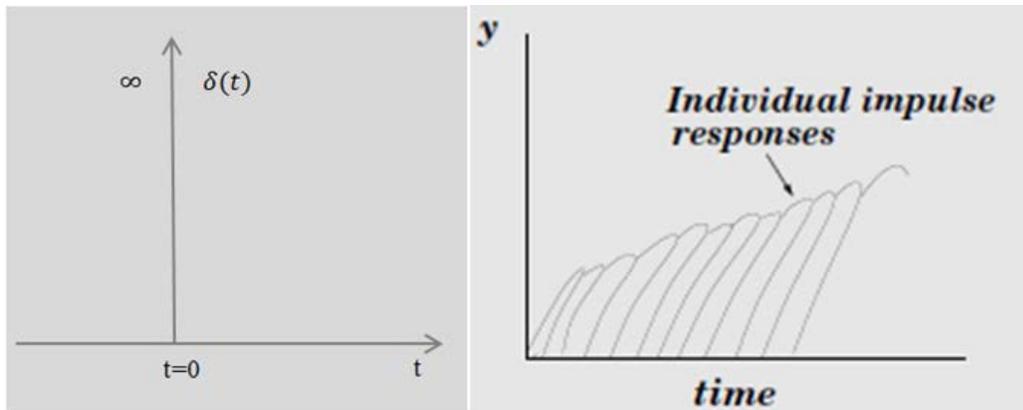


Figure 3-2: Unit impulse $\delta(t)$ and response to a set of individual impulses.

Any one of the individual response curves can be expressed as $u(\tau)g(t - \tau)$, where τ is the time that the impulse is applied (R.Leigh 2004). Then the response to any input $u(t)$ can be given by the convolution integral using superposition:

$$y(t) = \int_0^t g(t - \tau)u(\tau)d\tau = (g * u)(t) \quad \text{Eq. 3-1}$$

If $U(s)$ and $G(s)$ are assumed to be the Laplace transform of $u(t)$ and $g(t)$ respectively, then by taking Laplace transform of both sides of this equation, we get:

$$Y(s)=G(s)U(s) \quad \text{Eq. 3-2}$$

This is how Laplace transform makes it much simpler to find the response to any input impulse $u(t)$.

3.2 Closed Loop Transfer Function

A closed loop system is shown in Figure 3-3 where C is the controller, G is the plant, r is the controller input or desired value and y is the plant output and the transfer functions of these elements are $C(s)$, $G(s)$, $R(s)$ and $Y(s)$, respectively. The system is linear and time invariant which means the elements of transfer functions do not depend on time.

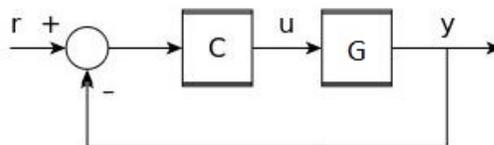


Figure 3-3: Closed loop system.

The characteristics of this system are defined as:

r: Controller Input

C(s): Controller Transfer Function

G(s): System Response

U(s): Controller Output

Y(s): System Output

E: Error

Error is the difference between system output and controller input:

$$E(s) = R(s) - Y(s) \quad \text{Eq. 3-3}$$

Then based on the Eq. 3-2 driven in previous section, the following relations can be expressed:

$$U(s) = C(s) E(s)$$

$$Y(s) = G(s) U(s)$$

Then, by substituting error by its definition from Eq. 3-3, we get:

$$U(s) = C(s) [R(s) - Y(s)]$$

$$Y(s) = G(s) C(s) [R(s) - Y(s)]$$

Eventually, the final result would be:

$$Y(s) = H(s) R(s)$$

Where:

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad \text{Eq. 3-4}$$

H(s) is the transfer function of this closed loop control system.

3.3 Dynamic Model of a DC Motor

Since the robotic surgery system discussed in this thesis is composed of 6 DC motors as the main elements of its control system, it's essential to develop a linear model to analyze the dynamic behavior of the system.

A DC motor as shown in Figure 3-4 consists of stator magnets that provide a constant magnetic field, rotor coils that provide electromagnetic force for rotation. The coils are powered from the commutator and the brushes. The more the numbers of coils are the smoother the rotation gets. The geometrical characteristics and position of the brushes (and the commutator of course) will be responsible for changing the magnetic field of the electromagnets according to the position of the rotor. The brushes are two metallic pieces that act like springs. On one side, they have a piece of conductive material, usually made of carbon to stand against friction. On the other side, they have the pin that the power supply is applied to the motor. The brushes are pushed (by the spring action of the metallic part) against the commutator. The commutator is a metallic ring, also conductive and is fixed on the shaft of the motor. The shaft transfers the rotation to the rotating elements of the mechanism.

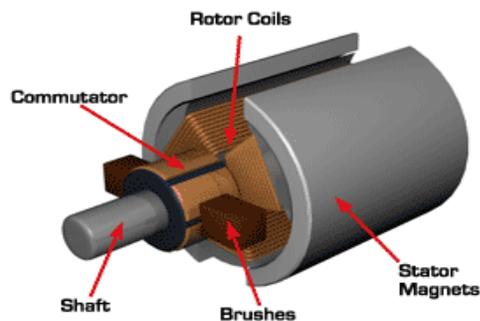


Figure 3-4: A DC motor composition.

Also a DC motor can be modeled as an electric circuit of the armature with mechanical load as shown in Figure 3-5 (Babuska and Stramigioli 1999).

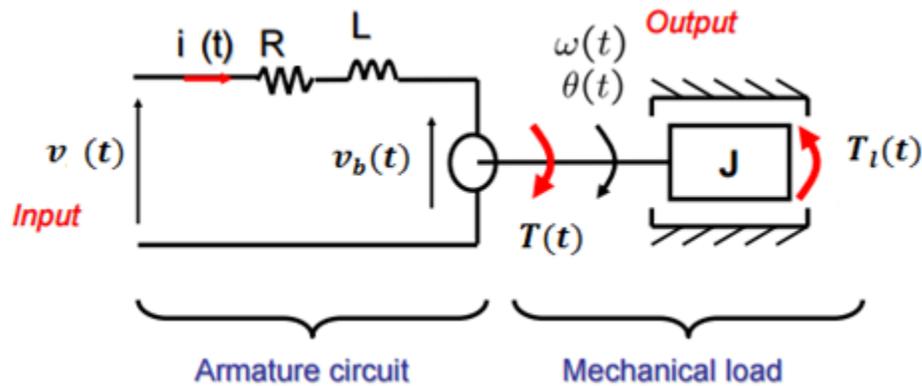


Figure 3-5: A DC motor modeled as an electric circuit of the armature with mechanical load.

The rotor and the shaft are assumed to be rigid. The parameters of this model are defined as follows:

v : applied voltage

i : armature current

v_b : EMF voltage

θ : angular position

ω : angular velocity

J : moment of inertia of the rotor

B : damping (friction) of the mechanical system

The input is the armature voltage v in Volts (driven by a voltage source). Measured variables are the angular velocity of the shaft ω in radians per second, and the shaft angle θ in radians.

For the armature electric circuit the following equation can be written based on Kirchhoff law:

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + v_b(t) \quad \text{Eq. 3-5}$$

Where, v_b is back-electromotive voltage which is a voltage induced in the coil due to its motion inside the magnetic field, as stated by Faraday's law, R is the armature resistance and L is its electric inductance.

The motor torque, T , is related to the armature current, i , by a constant factor k_T :

$$T(t) = k_T i(t) \quad \text{Eq. 3-6}$$

The back electromotive force (emf), v_b , is related to the angular velocity by:

$$v_b(t) = k_b \omega(t) = k_b \frac{d\theta}{dt} \quad \text{Eq. 3-7}$$

Where k_b is the back-EMF constant. Also, for the power in this system the equation $V_i = k_T \omega$ can be written, which results to $k_T = k_b$. Therefore, both these constant factors can be substituted with simple K .

Also, the following equation can be written based on the Newton's law:

$$T(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) + T_l(t) \quad \text{Eq. 3-8}$$

While: $\omega(t) = \dot{\theta}(t)$

Where, T_l is the load torque.

Now that the equations of the system are derived, it's time to take Laplace transform of these equations. Using the Laplace transform, Eq. 3-5 and Eq. 3-7 can be written as:

$$V(s) = RI(s) + LsI(s) + V_b(s) \quad \text{Eq. 3-9}$$

and
$$V_b(s) = K\Omega(s) \quad \text{Eq. 3-10}$$

Where, s denotes the Laplace operator. Also, for Eq. 3-8 we get:

$$T(s) = Js^2\theta(s) + Bs\theta(s) + T_l(s) \quad \text{Eq. 3-11}$$

From combination of Eq. 3-9 and Eq. 3-10, $I(s)$ can be expressed as:

$$I(s) = \frac{V(s) - Ks\theta(s)}{Ls + R} \quad \text{Eq. 3-12}$$

Since $T(s)$ is also equal to $K I(s)$, combining Eq. 3-11 and substituting $I(s)$ from Eq. 3-12, we get:

$$Js^2\theta(s) + Bs\theta(s) + T_l(s) = K \frac{V(s) - Ks\theta(s)}{Ls + R} \quad \text{Eq. 3-13}$$

From this equation, the transferred functions of shaft angle $\theta(s)$ and angular velocity $\Omega(s)$ would be found as follows:

$$\theta(s) = \frac{T(s) - T_l(s)}{s(Js + B)} \quad \text{Eq. 3-14}$$

$$\Omega(s) = \frac{T(s) - T_l(s)}{Js + B} \quad \text{Eq. 3-15}$$

Assuming the free load case ($T_l(s)=0$), Eq. 3-13 for the DC motor is shown in the block diagram in Figure 3-6.

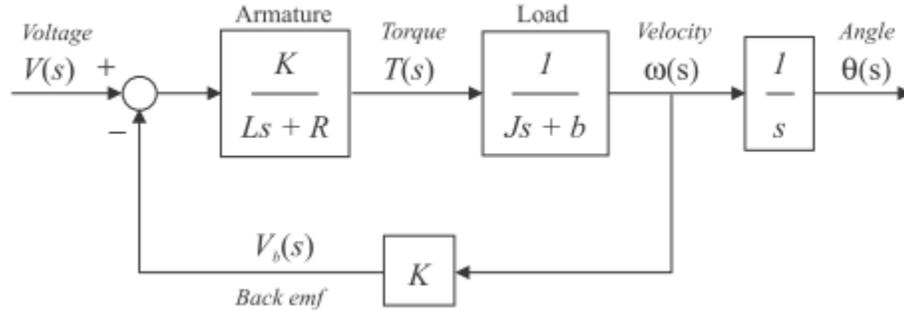


Figure 3-6: A block diagram of the DC motor.

The resulted transfer function for the input voltage, $V(s)$, to the output angle, $\theta(s)$, directly follows:

$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(Ls + R)(Js + b) + K^2]} \quad \text{Eq. 3-16}$$

From the block diagram in Figure 3-6, it is easy to see that the transfer function from the input voltage, $V(s)$, to the angular velocity, $\omega(s)$, is:

$$G_v(s) = \frac{\Omega(s)}{V(s)} = \frac{K}{(Ls + R)(Js + b) + K^2} \quad \text{Eq. 3-17}$$

Now that the position and velocity transfer functions, $G_a(s)$ and $G_v(s)$ are derived, they can be used to analyze the behavior of this control system with real motor parameters and specifications.

3.4 Motor (V) Transfer Function Analysis

The DC brushed motor (V) of the robot is chosen for further analysis. The technical specifications of this motor are:

$$J= 0.13*10^{-7} \text{ (Kgm}^2\text{)}$$

$$R=27.4 \quad \text{(Ohm)}$$

$$b=10^{-7} \quad \text{(Nms/rad)}$$

$$L=0.399*10^{-3} \text{ (H)}$$

$$K=0.01 \quad \text{(Nm/A)}$$

The position transfer function for this motor would be:

$$G_a(s) = \frac{0.01}{5.187 * 10^{-12}s^3 + 3.562 * 10^{-7}s^2 + 0.1027 * 10^{-3}s}$$

Also, the velocity transfer function is:

$$G_v(s) = \frac{0.01}{5.187 * 10^{-12}s^2 + 3.562 * 10^{-7}s + 0.1027 * 10^{-3}}$$

The above transfer functions can be entered into the Matlab code which is presented in "Appendix C" to plot the responses for the velocity and angle position. This code takes the position and velocity transfer functions and returns the plots of response of the system to step and impulse inputs.

3.4.1 Step and Impulse Responses

The step and impulse responses of the system are the first characteristic of the system to be analyzed. The step response is the response of the system to a unit input at $t \geq 0$ which here means to connect the motor to a power source that produces one volt. On the other hand, the impulse response is an infinite input at time $t=0$ and zero otherwise. Note that the Laplace transform of the step function is equal to one over s , $[V(s)_{\text{step}}=1/s]$, while the Laplace transform of the impulse function is one, $[V(s)_{\text{impulse}}=1]$. Also, by definition, the derivation of the step function is the impulse function.

The step and impulse responses of the position transfer function are shown in Figure 3-7.

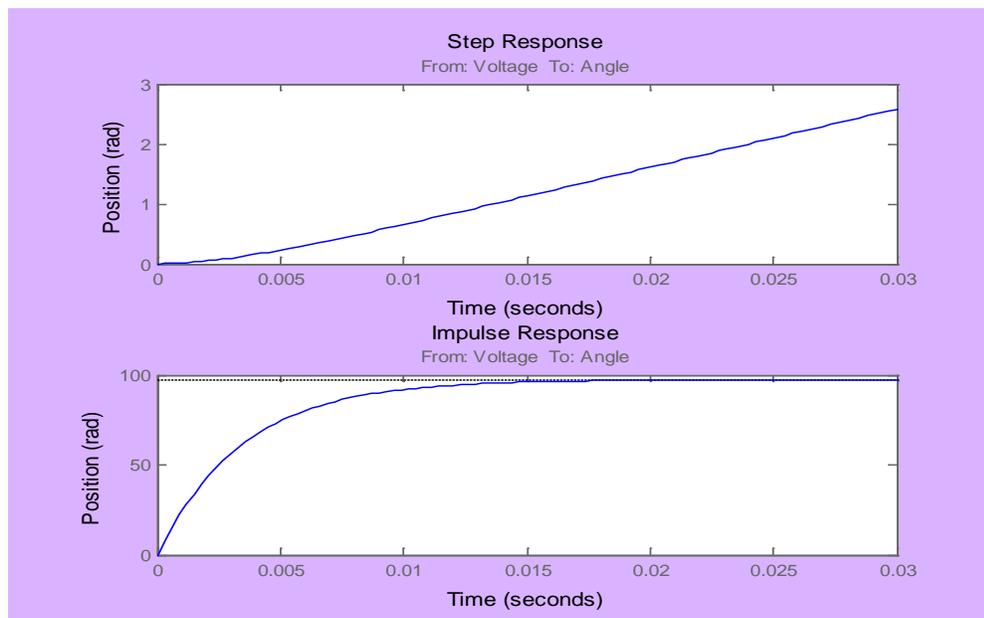


Figure 3-7: The step and impulse responses of the position transfer function.

After giving the input of unit voltage to the motor the step response plot is rising as it is expected due to increase in angular position when the motor starts rotating.

However, the response of the position to the impulse input is converging to 100 rad, which can be validated by taking the inverse Laplace transfer of $G_a(s)$ when $V(s)=1$, in order to find $G_a(t)$ and plot the result.

Also, same responses are plotted for velocity transfer function which is shown in Figure 3-8.

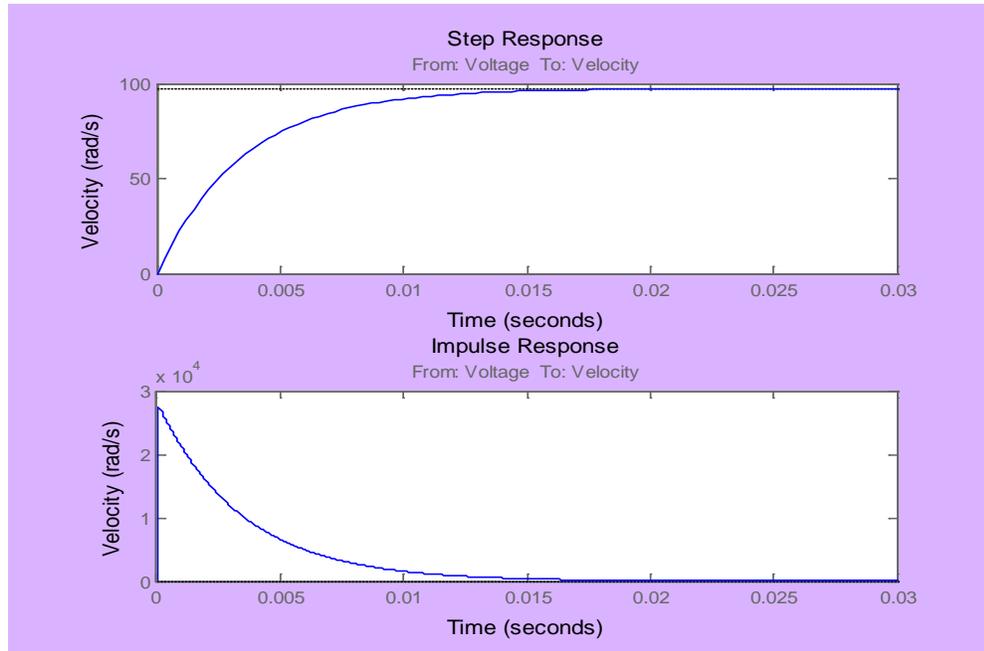


Figure 3-8: The step and impulse responses of the velocity transfer function.

The step response of velocity shows the same behavior as the impulse response of angular position which is expected since the velocity is related to angular position by $\Omega(s) = s\theta(s)$ and the Laplace transfer of step function is related to the Laplace transfer of the impulse function by $V(s)_{\text{step}} = s V(s)_{\text{impulse}}$.

$$G_v(s) = \frac{\Omega(s)}{V(s)_{\text{step}}} = \frac{s\theta(s)}{s V(s)_{\text{impulse}}}$$

The impulse response of velocity also shows an expected behavior by a rapid rise in velocity at instant of impulse and then going back to zero.

3.4.2 Frequency Domain Analysis

In order to analyze the frequency response of the system, the “Bode diagram” is plotted as can be seen in Figure 3-9.

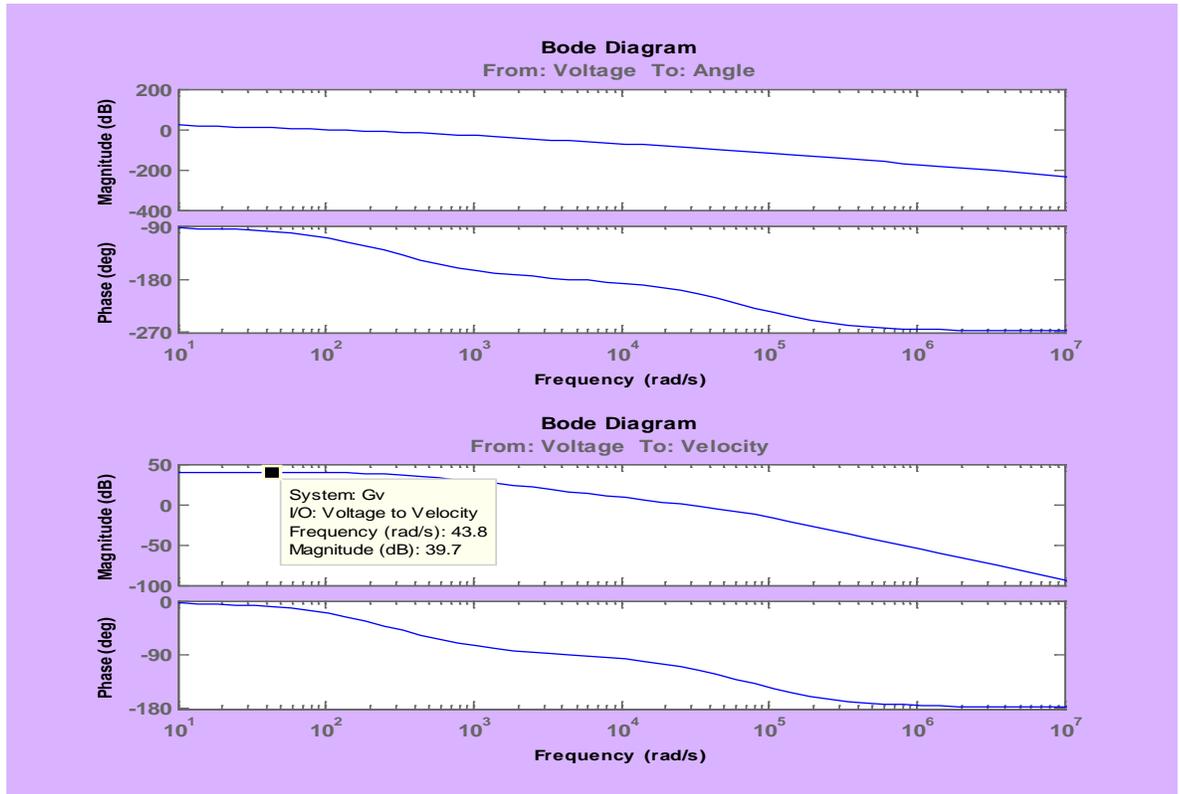


Figure 3-9: The position and velocity responses of the system to the voltage frequency.

A Bode Plot is a useful tool that shows the gain and phase response of a given linear time invariant system for different frequencies. The frequency of the bode plots are plotted against a logarithmic frequency axis. Every tick mark on the frequency axis represents a power of 10 times the previous value. The bode magnitude plot measures the system input/output ratio in special units called decibels, which is in this case can be expressed as: $dB = 20\log G(s)$.

The Bode phase plot measures the phase shift in output with respect to input in degrees. The reason to observe the frequency domain of the system is to make sure there would not be any critical frequency to create excitation in the system.

As it can be seen in these diagrams, the bode magnitude goes down as the frequency increases for both angle and velocity responses. To evaluate the result, for example for frequency between 10 and 100 the magnitude of the velocity response is almost constant and approximately equal to 40 dB which agrees with following calculations:

$$dB = 20 \log G_v(s) = 20 \log \frac{0.01}{5.187 * 10^{-12} s^2 + 3.562 * 10^{-7} s + 0.1027 * 10^{-3}} \cong 20 \log 100 = 40 \text{ dB}$$

Also, from these diagrams it is assured that there is no critical frequency and excitation risk to threaten the system.

3.5 Proportional Controller

Extra space

Now that the behavior of the system has been observed through step, impulse and frequency responses, it's time to add a controller. First, a simple proportional controller would be added to the system as shown in Figure 3-10.

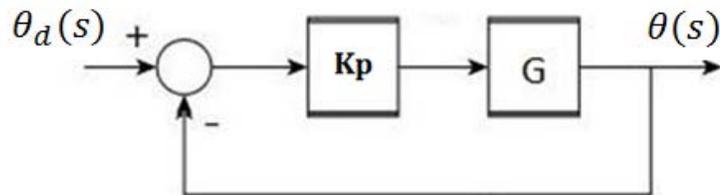


Figure 3-10: A simple closed loop system with proportional controller.

In this system, $\theta_d(s)$ is the desired angular position, K_p is the proportional gain and $\theta(s)$ is the actual position. The error is defined as the difference of actual position and desired position:

$$E(s) = \theta_d(s) - \theta(s)$$

In the proportional controller the error term is multiplied by proportional gain, K_p , to produce the controller output:

$$U(s) = K_p * E(s)$$

Therefore, the closed loop transfer function of proportional controller system according to Eq. 3-4 is:

$$H(s) = \frac{K_p \cdot G(s)}{1 + K_p \cdot G(s)} \quad \text{Eq. 3-18}$$

Then, for the motor 5 that is being discussed in this chapter, the position transfer function of proportional controller will be as followed:

$$H_d(s) = \frac{0.01K_p}{5.187 * 10^{-12}s^3 + 3.562 * 10^{-7}s^2 + 0.0001027s + 0.01K_p} \quad \text{Eq. 3-19}$$

If a big value is been chosen for proportional gain it will result in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when

responding to system disturbances. The goal is to find the best value for K_p that minimizes the error and the response time. To find the best K_p , the step response of this system for $K_p=1, 5, 10$ and 50 has been observed.

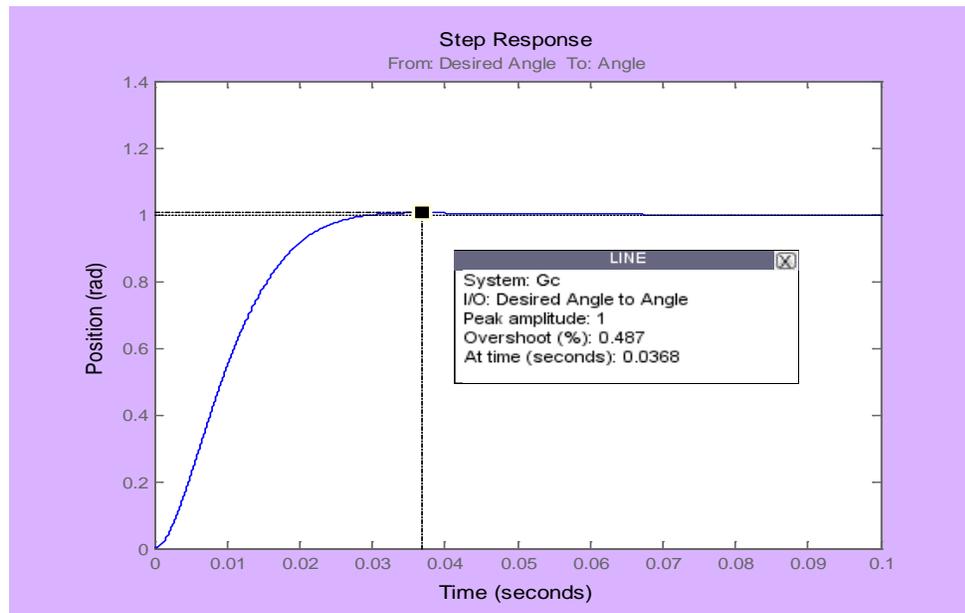


Figure 3-11: The proportional controller step response for $K_p=1$.

For the first plot the value of K_p assumed to be equal one, $K_p=1$, the result shows a quite small overshoot while the system very slowly approaches to the desired value. So, the proportional gain must be increased to give a faster approach.

The next response is plotted for $K_p=5$ as shown in Figure 3-12. Here, the result shows a faster approach to the desired response while gives a bigger over shoot at first which is about 1.27 times of the desired value.



Figure 3-12: The proportional controller step response for $K_p=5$.

To further the study of the effect of proportional gain, the results for $K_p=10$ and 50 are being plotted in Figure 3-13 and Figure 3-14.

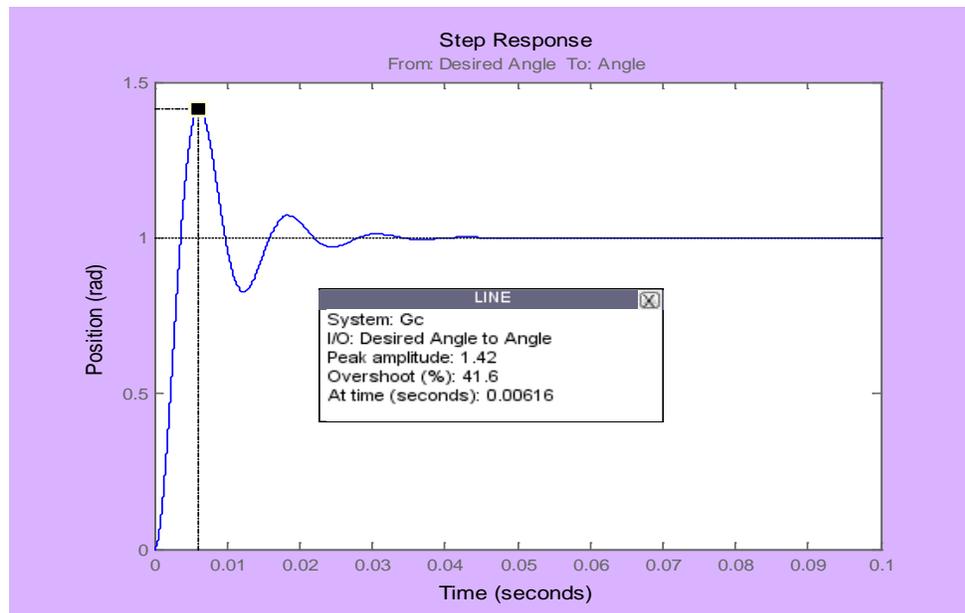


Figure 3-13: The proportional controller step response for $K_p=10$.

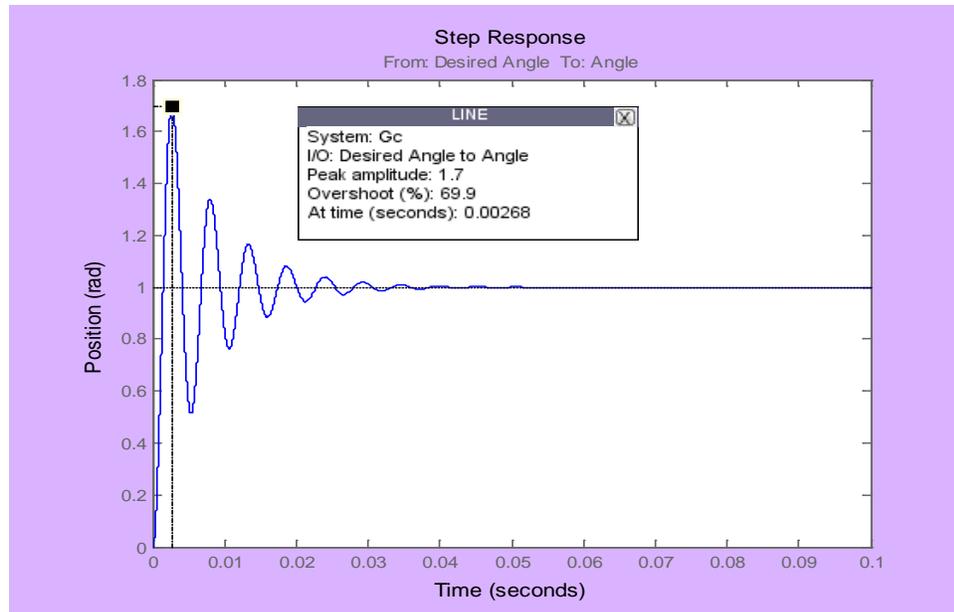


Figure 3-14: The proportional controller step response for $K_p=50$.

As it can be seen in these diagrams, by increasing the proportional gain larger overshoots will be resulted while the response time goes down.

The proportional controller does not deliver a proper performance for the high precision robot which is the interest of this search. Therefore a more advanced control system must be considered.

3.6 PID Controller

PID is an acronym for proportional- integral- derivative control system. This is the most common controller used in many industries. The structure of this three terms feedback controller helps to attain a set point irrespective of disturbances or any variation in characteristics of the plant of any form. It calculates its output based on the measured error and the three controller gains; proportional gain K_p , integral gain K_i , and derivative gain K_d . The proportional gain simply multiplies the error by

a factor K_p . This reacts based on how big the present error is. The integral term is a multiplication of the integral gain and the sum of the recent errors. The integral term helps in getting rid of the accumulation of past errors to eliminate the residual steady state error and causes the system to catch up with the desired set point. The derivative controller determines the reaction to the rate of which the error has been changing. It helps with prediction of future errors based on current rate of change, improves settling time and stability of the system.

The final output of the PID controller (U) is calculated using the following equation:

$$u(t) = K_p * e(t) + K_I \int e(t)dt + K_D \frac{d}{dt} e(t) \quad \text{Eq. 3-20}$$

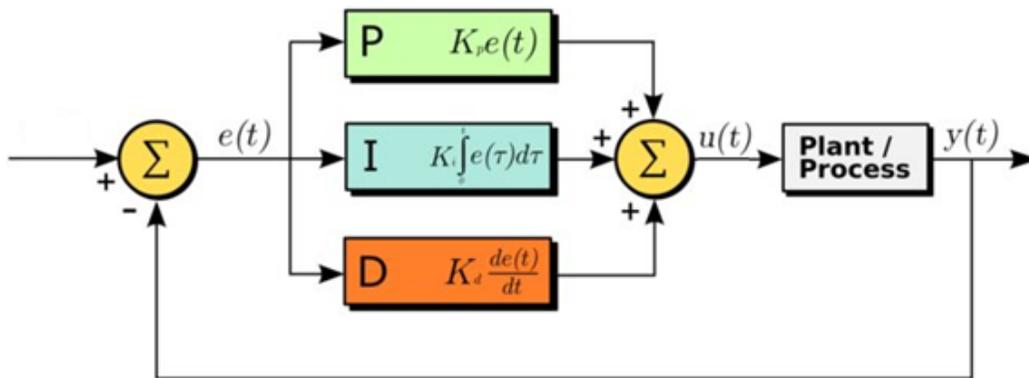


Figure 3-15: The block diagram of closed loop PID controller.

3.6.1 Transfer Function of PID Controller

The transfer function of a PID controller is found by taking the Laplace transform of Eq. 3-20:

$$U(s) = K_p e(s) + \frac{K_I}{s} e(s) + K_D s e(s)$$

Also can be expressed as:

$$U(s) = \left(K_p + \frac{K_I}{s} + s K_D \right) e(s) = C(s) e(s)$$

Where:

$$C(s) = K_p + \frac{K_I}{s} + s K_D$$

$C(s)$ is called the PID controller transfer function. Combining this result with Eq. 3-4, the position transfer function of PID controller is found as follows:

$$H(s) = \frac{\theta_d(s)}{\theta(s)} = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)} \quad \text{Eq. 3-21}$$

With this closed loop transfer function the behavior of the system can be analyzed in order to find the best proportional, integral and derivative gains. This process is called manual tuning. There are a couple of strategies on how PID can be tuned; this includes trial and error tuning method, Ziegler-Nichols tuning method (Meshram and Kanojiya 2012) etc. The goal is to minimize settling time with no overshoot. In this case the best optimum result is obtained by trial and error.

3.6.2 Proportional Gain Effect

To analyze the effect of proportional gain in PID system, the integral and derivative gains assumed to be $K_i=100$ and $K_d=1$. The results are plotted for $K_p=1,50$ and 500.

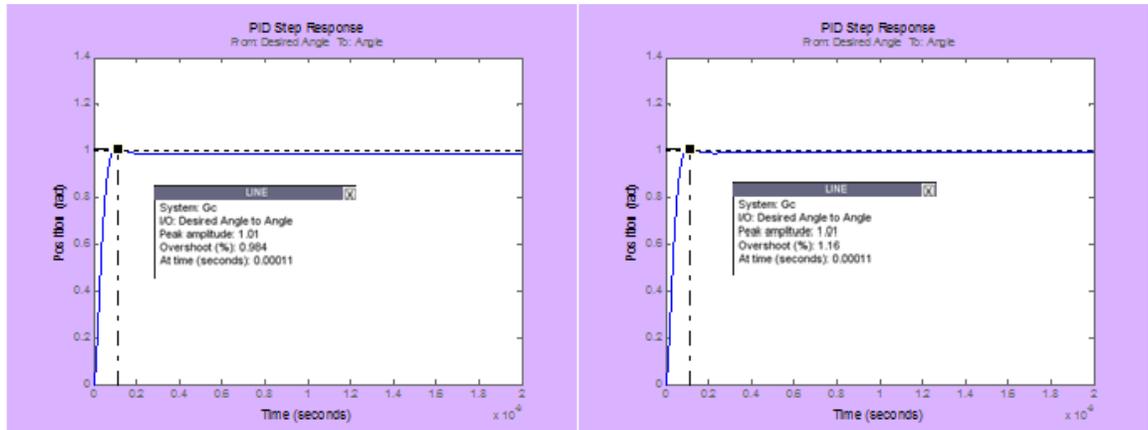


Figure 3-16: The PID controller step response for $K_p=1$ (left) and $K_p=50$ (right).

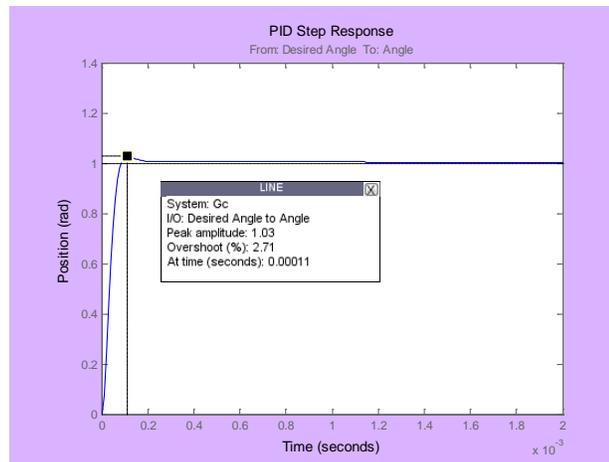


Figure 3-17: The PID controller step response for $K_p=500$.

These plots show even though the system is sensitive to the value of K_p , but the result does not significantly change when K_p value is changed from 1 to 50 and even

to 500. As it was discussed earlier, increasing K_p results in shorter settling time and larger overshoot.

3.6.3 Integral Gain Effect

To study the effect of integral gain in PID system, the proportional and derivative gains assumed to be $K_p=100$ and $K_d=1$. The results are plotted for $K_i=1$ and 100.

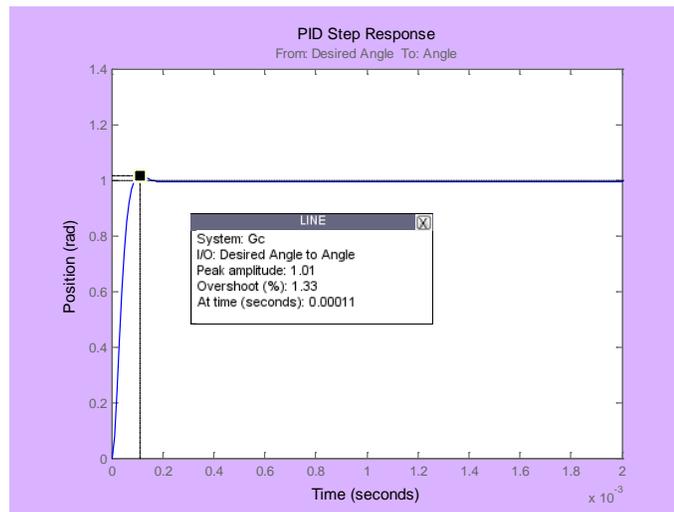


Figure 3-18: The PID controller step response for $K_i=1$.

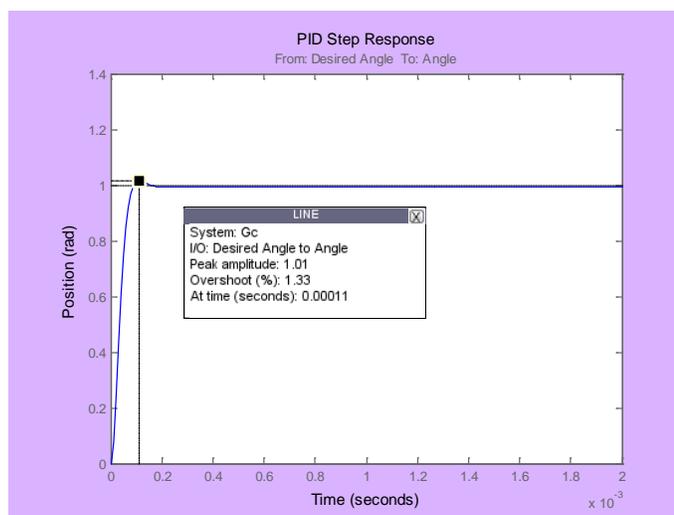


Figure 3-19: The PID controller step response for $K_i=100$.

In this case, even with a big change in value of K_i from 1 to 100, it does not show any significant impact on quality of response.

3.6.4 Derivative Gain Effect

To study the effect of derivative gain in PID system, the proportional and integral gains assumed to be $K_p=100$ and $K_i=10$. The results are plotted for $K_d=0, 1$ and 2 .

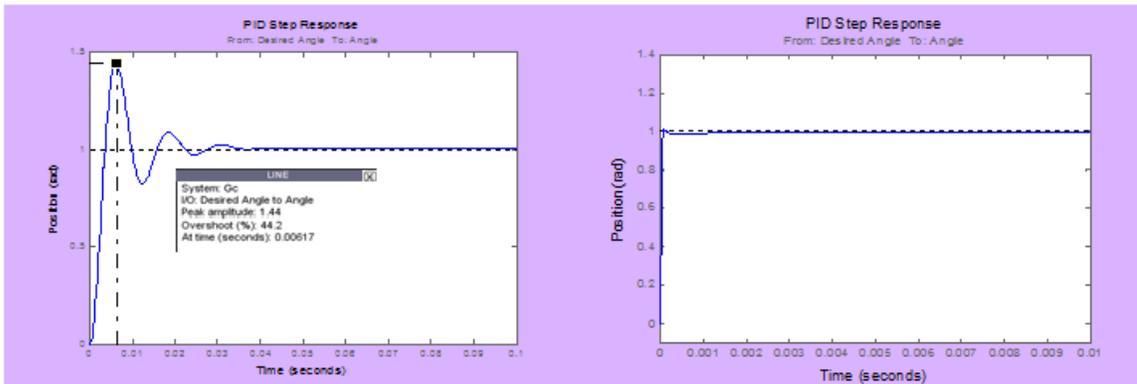


Figure 3-20: The PID controller step response for $K_d=0$ (left) and $K_d=1$ (right).

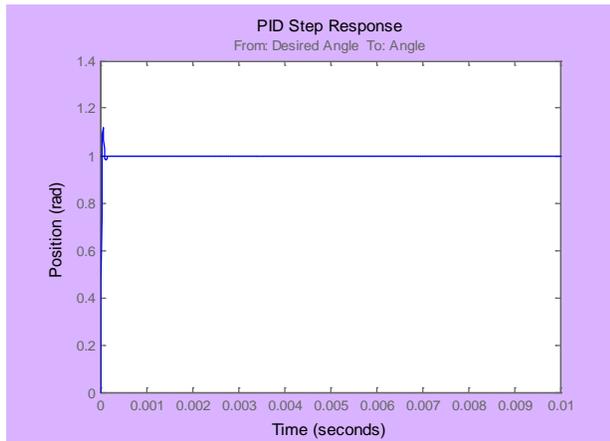


Figure 3-21: The PID controller step response for $K_d=2$

As it can be seen in these diagrams, the response of the system is highly sensitive to the value of K_d that even a very small changes in this value have a big impact on quality of response.

3.6.5 PID Controller Gain Values

The best response for this system is obtained with $K_p=200$, $K_i=100$ and $K_d=0.7$. As it can be seen in Figure the response reached with almost no over shoot (0.01%) in a very short time.

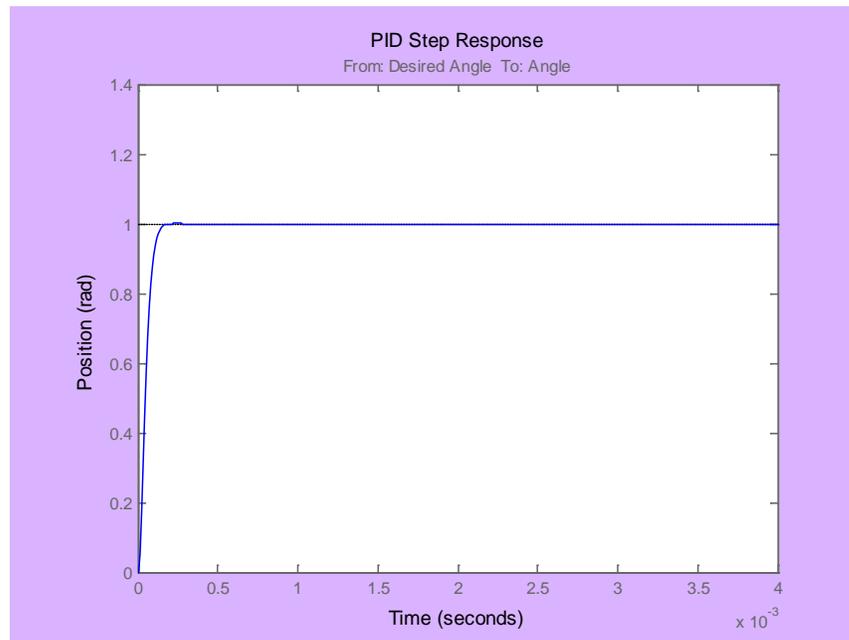


Figure 3-22: The best PID response.

These are the best PID gains could be found for this specific system. If K_p is chosen to be smaller than this value, the system gets to the set point slower and if a bigger value is selected for K_p the over shoot would be resulted. However the system is not pretty much sensitive to the value of K_i , still this value of K_i gives the faster result. If the value of K_d is chosen to be slightly less than this value, like 0.6, it will result in steady state error, and if it is chosen to be slightly more than this value, like 0.8, it will result in over shoot.

CHAPTER 4 : CONTROLLER ARCHITECTURE

In this chapter the control system of the robot with its all elements and features will be discussed. The control system is composed of the encoder, controller module, Controller Area Network (CAN), and controller software, as shown in Figure 4-1.

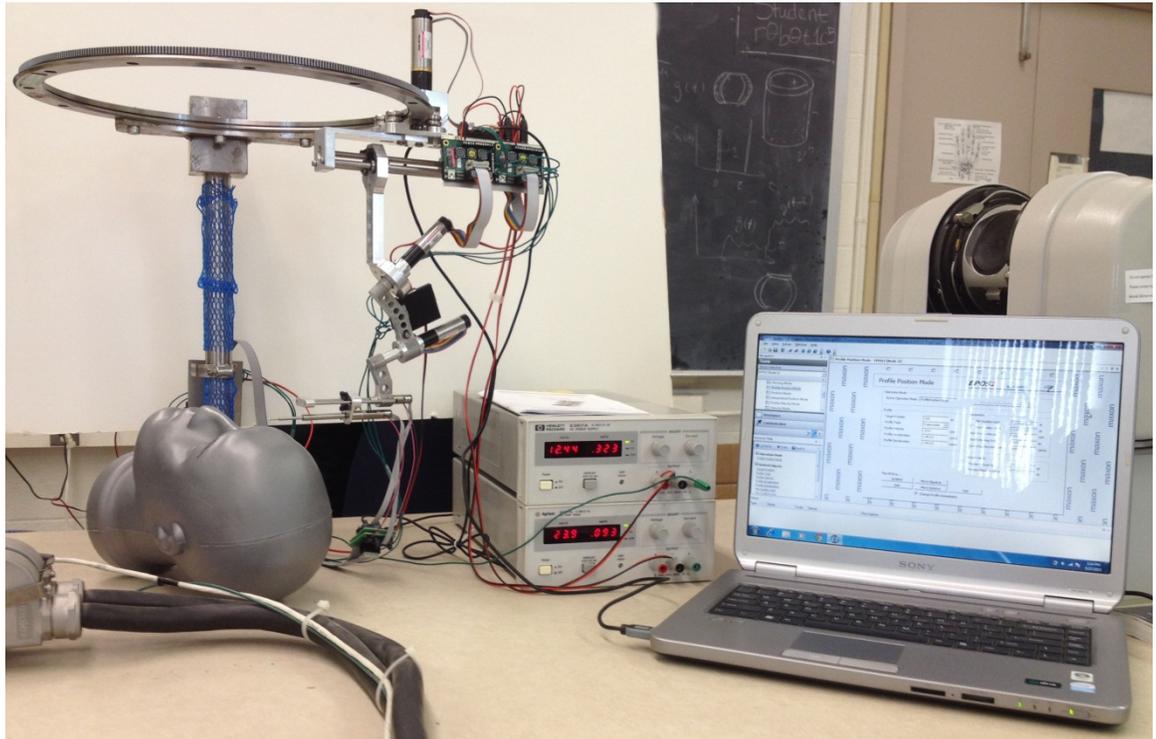


Figure 4-1: The micro-surgery robotic system with its motors and controllers connected through the USB port to the controller software.

4.1 Encoder

The incremental encoder is used for position and speed evaluation (Kafader 2014). Incremental digital encoder is one of the most common sensor type for

measuring speed and position in micro-drive tools. Many various designs with different working principles are using this kind of encoder, but all of them basically subdivide one revolution into a number of steps (increments), and send signal pulses to the controller each time an increment is detected. Incremental encoders generally supply square-wave signals in two channels, A and B, which are offset by one quarter signal length (or 90 electrical degrees).

In this robot the encoder pulse for motors (I) to (V) is 512 cpt (counts per turn) and 128 cpt for motor (VI). The high resolution provided by these encoders gives the robot the ability of the micro positioning control. The encoder part and its connection cable for motor (I) are shown in Figure 4-2 by red arrows.

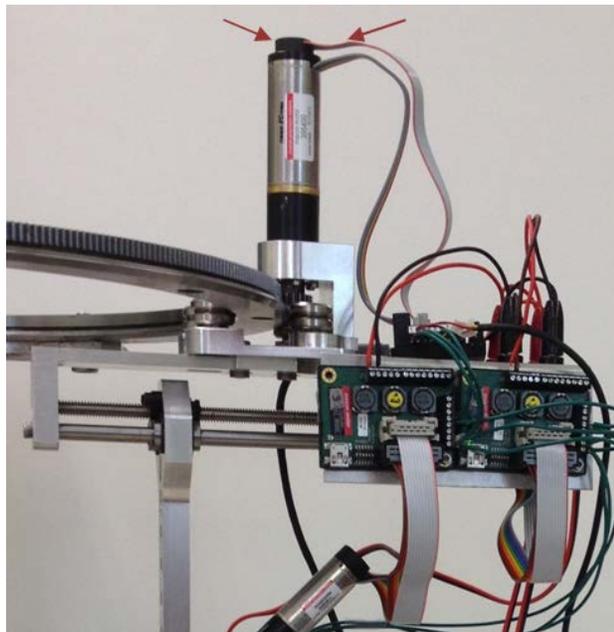


Figure 4-2: Motor (I) encoder and its cable

The characteristic parameter of the encoder is the number of square pulses N per revolution, given in cpt (counts per turn). By counting the number of state transitions in both channels, the real resolution is four times higher. These states

are called quadrature counts or quad counts (qc) and are used as the position unit in EPOS (the controller software) systems.

The higher the resolution, the more accurately the position can be measured, and the more accurately the speed can be derived from the change in position as a function of time. The direction of rotation follows from the sequence of the signal pulses of channel A and B: Channel A leads in one direction, channel B leads in the other direction. As shown in Figure 4-3.

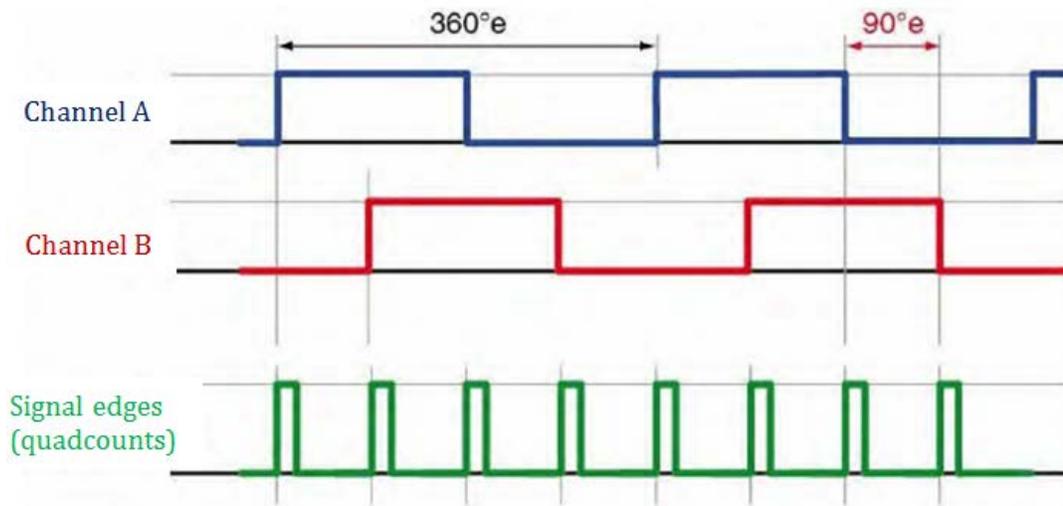


Figure 4-3: The signals of an incremental encoder.

Position evaluation is determined by counting the signal edges (increments) from a defined reference position e.g. home position. The unit of positioning is quad counts (qc).

Speed evaluation is calculated from the position change per sampling interval. In this case, the natural unit of speed is quad counts per millisecond. Therefore, speed can only be expressed in steps of 1qc/ms, which is usually translated in the more

common and practical unit rpm. Acceleration and deceleration values are given in the practical unit rpm/s.

A high encoder pulse count N not only results in a high position resolution, but also improves the control dynamics. Feedback information about a change in the actual position value is obtained more quickly and the control circuit can initiate corrective action much faster.

4.2 Controller Module

All six controllers in this robotic system are modular constructed digital positioning controller. It is suitable for DC and EC motors with incremental encoder with a power range from 1 to 700 watts. A number of operating modes provides flexible application in a wide range of drive systems in automation technology and mechatronics. Motion control is mechatronics; it is the combination of mechanics and electronics of actuators (motors) and sensors all controlled by a software.

The controller device used in this robotic control system is a CANopen motion controller device. CANopen is a standard maintained and supervised by the independent user organization CAN in Automation (CiA). A Controller Area Network, or CAN for short, is a common message based protocol used in control systems (CAN in Automation (CiA) 1992). It is also used in industrial/robotic applications such as motor control.

The following discussion is to provide a better understanding of how communication is organized in this system.

On a physical level, each device - called a node and given its distinctive address (node number) - is mounted in parallel on the bus (Kohanbash 2014). While there is no standard way of wiring a CAN bus there are always two wires/signals, and often another two. The first two are CAN+ (or CAN_H) and CAN- (or CAN_L) which are used for sending the CAN data. The next two wires are for power and ground that are connected to devices on the network. All of CAN devices can be wired in series creating a large bus of devices. The bus network needs proper termination on both sides to avoid signal reflections, which means having a 120 Ohm resistance at each end. A physical layout of this robot with 6 nodes is shown in Figure 4-4.

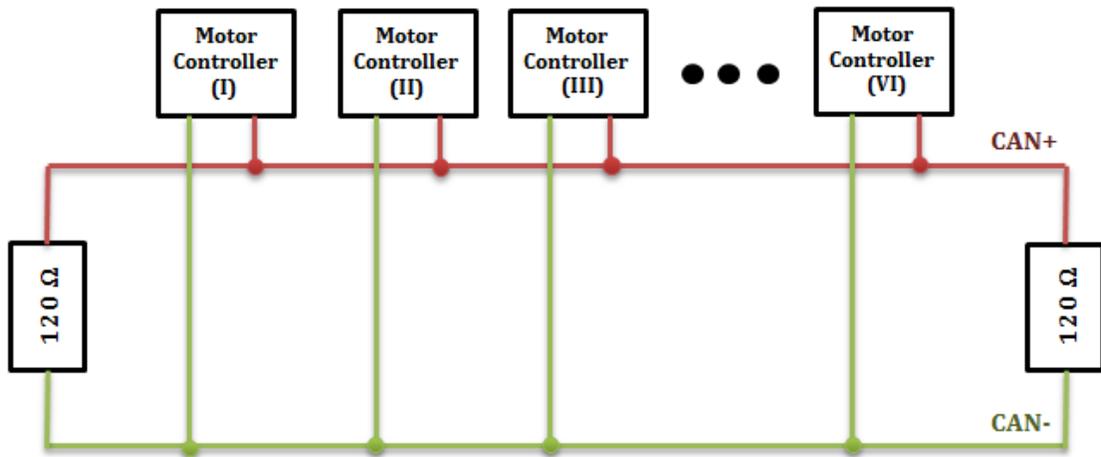


Figure 4-4: Physical layout of the six motor controllers CAN bus.

Signal bits require a certain time to spread over the whole bus. Therefore, the transmission rate depends on the bus length. Typically in robotics each CAN network will have one “master” node which is typically a computer or micro-controller that requests information or listens for alerts/emergencies from all of the CAN devices on the network. In this robot master node is node 1 which is the

controller of motor I. This node sends and receives signals from other 5 controllers and communicates that to the software (computer).

There are two sets of controller used in this design. The first one is “EPOS2 24/2 for Maxon EC motors” which is used for first four motors and the other one is “EPOS2 24/2 for Maxon DC motors” which is used for motor (V) and (VI).

4.2.1 Maxon Controller for EC Brushless Motors

The EC motor controller with its connection ports is shown in Figure 4-5. This is the controller which is used to control first four motors of the robot.

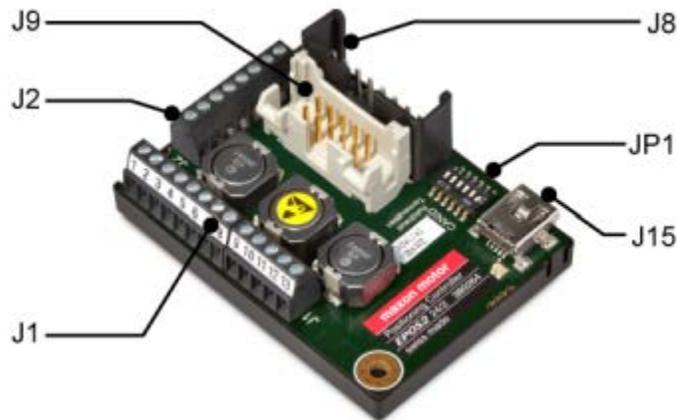


Figure 4-5: The EC brushless controller, interfaces and designations

J1 connectors are for supply and control signals which its pins 12 and 13 are wired to the power supply. **J2** connectors are communication connector that is used for CAN bus communication (Maxon, Application Notes Collection 2013). This connectors are used for send/receive control signals. Pin 1, 2 and 5 are wired to CAN high bus line, CAN low bus line and ground line, respectively. **J8** is motor/hall

sensors connector. **J9** is for the encoder cable connection and finally **J15** is the USB port to connect the controller to the computer.

JP1 is CAN node identification and is used to set CAN ID or node address (Maxon, Application Notes Collection 2013). The CAN ID is set with DIP switches 1 to 4 as shown in Figure 4-6. With these four binary switches, 15 addresses (1 to 15) may be coded.



Figure 4-6: JP1 – Numbering scheme

The CAN ID results in the summed values of “ON” DIP switch addresses. For instance DIP switch setting to identify the six nodes of this robotic system must be like the way shown in the Table 4-1. DIP switches 5 and 6 do not have any impact on the CAN ID.

Table 4-1: DIP switch setting for robotic system with 6 nodes.

CAN ID	DIP Setting	Calculation
1		2^0
2		2^1
3		2^0+2^1
4		2^2
5		2^0+2^2
6		2^1+2^2

DIP switch 5 is to detect CAN bit rate, and DIP switch 6 is CAN bus termination. The CAN bus must be terminated at both ends by a termination resistor of 120 Ω , typically. Depending on utilization of the controller, individual CAN bus termination settings must be performed. Using DIP switch 6, the controller-internal bus termination resistor can be activated/deactivated.

Also there is a Status LED on the controller board to display the current status of the controller as well as possible errors. Green LED shows the operating status

while Red one indicates errors. Figure 4-7 shown the controller (III) with the Green LED “ON”.

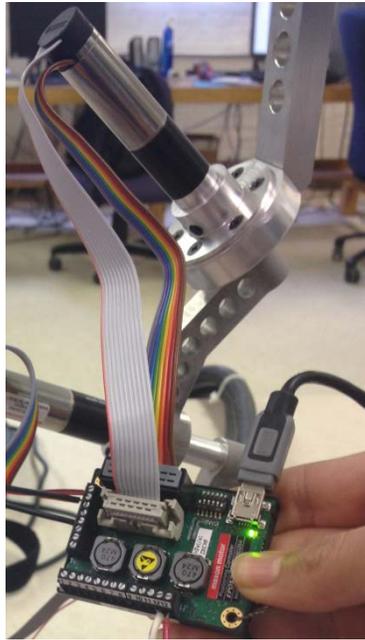


Figure 4-7: Controller (III) in operating status with the Green LED “ON”.

Maxon Controller for DC Brushed Motors

The DC motor controller with its connections is shown in Figure 4-8. This is the controller used for motors (V) and (VI) of the robot.

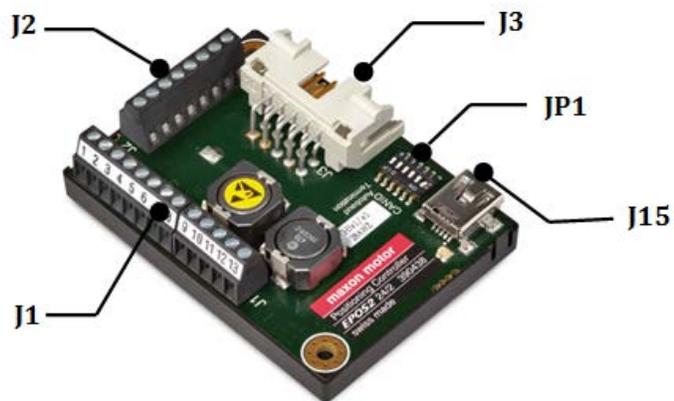


Figure 4-8: DC brushed controller, interfaces and designations.

The **J1**, **J2**, **J15** and **JP1** connectors are the same as discussed in the previous section. **J3** is the motor/encoder connector which connects to the motor and encoder through Adapter encoder connector as shown in the Figure 4-9 with motors (V) & (VI) controller wired to the system.

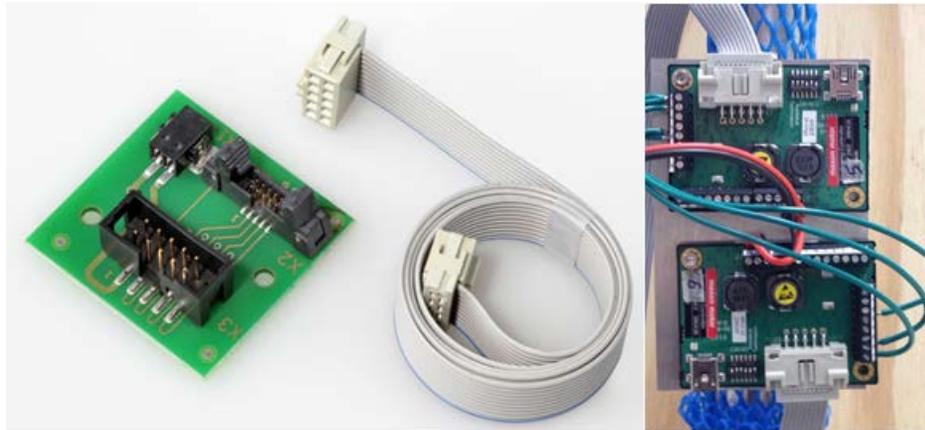


Figure 4-9: Motor/encoder connector (left) and DC brushed controllers (V) & (VI) wired to the system (right).

4.3 Controller Area Network (CAN)

The existing robot control system structures can be roughly divided into two categories, centralized control and decentralized control (Kazanzides 2009). As it was already mentioned, this robotic system benefits from CAN as its control system which is counted as a distributed or decentralized control system where each controller installed on the related joint of the robot and connected to the master processor devices through dedicated wiring. Clearly decentralized control provides more flexibility, asynchronous operation and parallel processing.

By placing the controllers on the robot arm, significant reductions in cabling could be achieved. In particular, the thick cables containing multiple wires for motor power and sensor feedback could be replaced by thin network and power cables.

The other reason that makes CAN network the best option for this case is that it provides an inexpensive, durable network that helps multiple CAN devices communicate with one another. The advantage of this is that electronic control units (ECUs) can have a single CAN interface rather than analog and digital inputs to every device in the system which decreases overall cost and weight of the system.

The other advantage of CAN is its advance Broadcast Communication ability. Each of the devices on the network has a CAN controller and is therefore intelligent. All devices on the network see all transmitted messages. Each device can decide if a message is relevant or if it should be filtered. This structure allows modifications to CAN networks with minimal impact. Additional nodes can be added without modification to the network.

Also, CAN system is very well capable of handling errors. Frames with errors are disregarded by all nodes, and an error frame can be transmitted to signal the error to the network. Global and local errors are differentiated by the controller, and if too many errors are detected, individual nodes can stop transmitting errors or disconnect itself from the network completely.

4.4 Easy Positioning System (EPOS) Controller Software

The EPOS Studio is the graphical user interface for all our Maxon controllers (Kafader 2014). It is capable of communicating with a motion controller with a built-in programmable logic controller (PLC).

Essentially, the controller contains two devices, a PLC and a motion controller, that are internally linked by a CAN bus. The EPOS Studio on the computer communicates with these two devices by means of a USB2.0 serial connection as the default communication. The PLC part is the master of the device. It controls the process flow and all slave units in the network; one of the slave units being the built-in motion controller as shown in Figure 4-10.

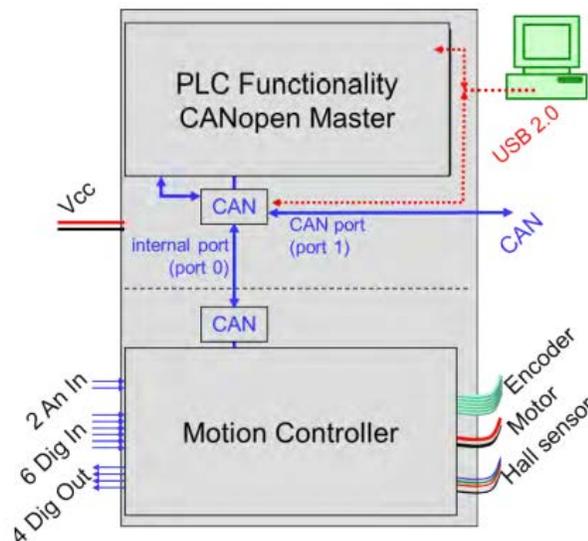


Figure 4-10: Schematic overview of the controller with external connections and communication.

To get connected to the controllers through the EPOS Studio on the computer, first we need to wire all 6 controllers together as it was discussed in section 4.2.

Then to connect this series of controller to the software, one of them need to be chosen (usually the first motor controller) to be connected through USB port to the computer. This one is going to be the master controller.

To get started, we need to create a new project. The controller hardware of the project is displayed in the Workspace tab of the Navigation.

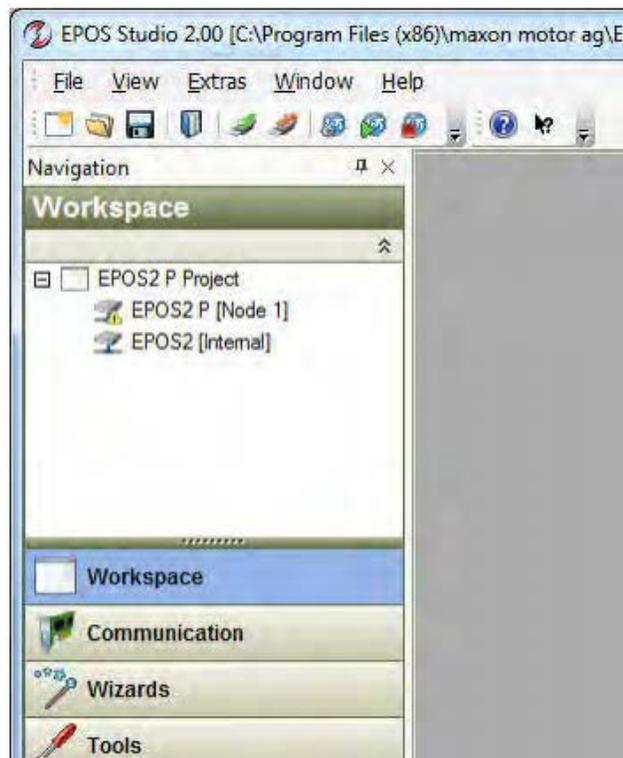


Figure 4-11: The EPOS Studio screen with the Workspace tab open.

The Communication tab shows the setup of the project from a communication point of view. The nice thing about the EPOS Studio is that we can directly communicate with any device in the network once we get connected to them. The first node connected to the computer through USB, usually is quickly recognized by the software. To find the rest of nodes on the network, there is a scanning device available on communication tab which scans for up to 127 nodes, but once it found

the rest of our 5 nodes, we can quit the scanning project and add those nodes to the network.

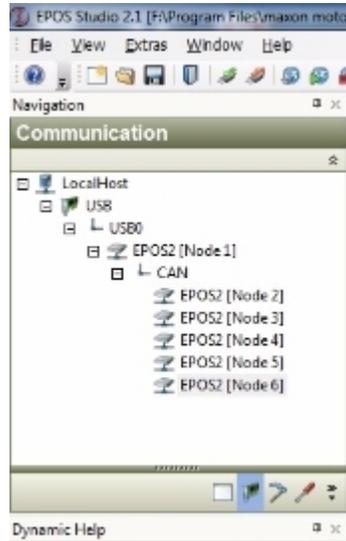


Figure 4-12: Communication set up with six nodes.

4.4.1 The Motion Controller

Now that we have configured the system the EPOS motion controller is now ready for further exploration. The tools for the exploration of the motion controller can be found in the Tools tab of the EPOS Studio.

The standard positioning mode for most applications is the “Profile Position Mode”. To execute a point-to-point movement, we need to use this option and enter required motion parameters such as target position, velocity and acceleration.

As it was discussed in section 4.1, position evaluation is determined by counting the increments from a defined reference position. The unit of positioning is quad counts (qc) which equals four encoder pulse.

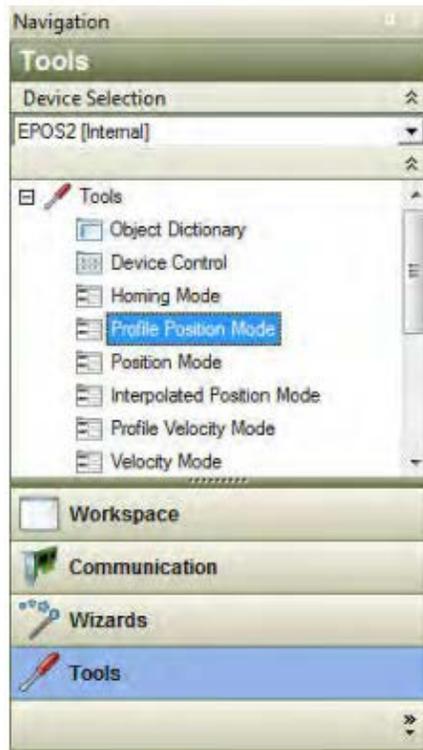


Figure 4-13: Different operating modes under the “Tools tab”.

The required calculation in order to convert the motion of six axes of the robot into the qc unit (which is the acknowledgeable unit by the EPOS Studio program) is summarized in Table 4-2.

For example to drive the robot arm around the ring for 45 degree, the target position for node 1 (the first Axis- rotating axis) would be 116736 “qc” which is resulted from:

$$\frac{4 * Encoder Pulse * Gear box ratio * 45 (target position)}{15 (the distance traveled each pinion turn)}$$

For the second axis which is a linear one, to move the arm 20 mm along its axis the target position would be 16126 “qc” which is calculated from:

$$\frac{4 * Encoder Pulse * Gear box ratio * 20 (target position)}{2 * 1.27 (the distance traveled per turn)}$$

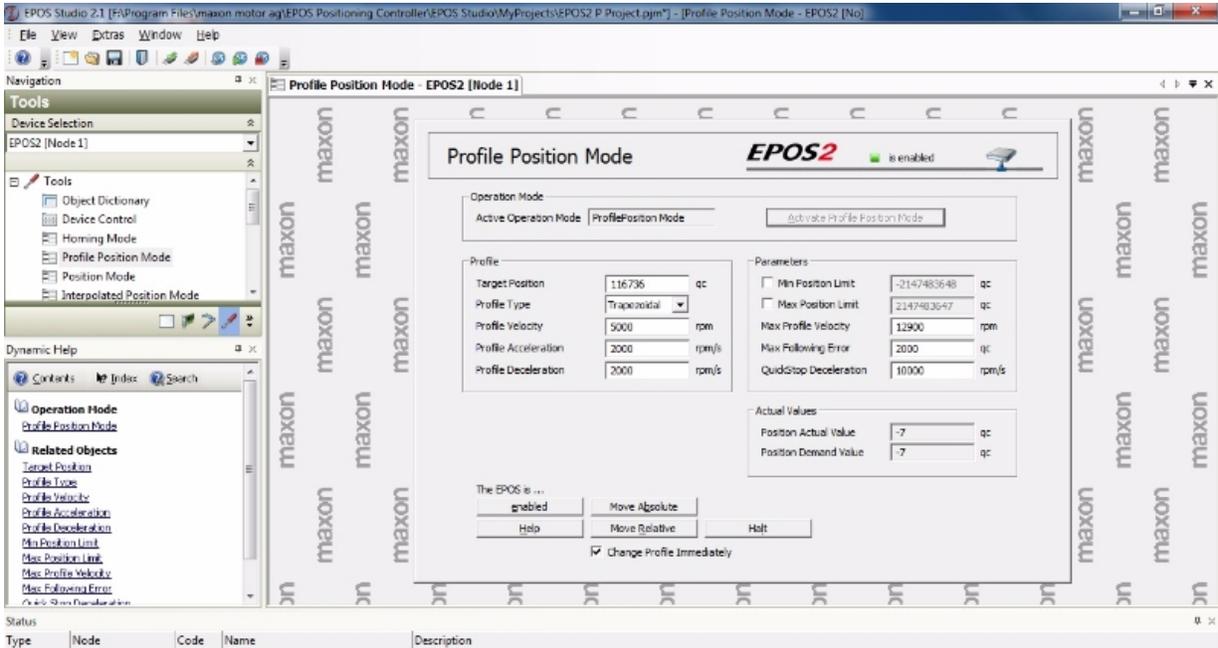


Figure 4-14: Data entry for 45 degree motion of node 1 (around the ring) in profile position mode.

Table 4-2: Data sheet to convert the motion of six axes to quad count (qc) unit.

Motor	Designed Range	Encoder Pulse	Gear box	Traveling Distance (each turn)	Input Data Angle/Distance	Quad Counts (qc)
Axis I	-45° to 225°	512	19	15 Each pinion turn travel 15° around the ring	45 °	4*512*19*45/15 116736
Axis II	0 to 90 mm	512	1	1.27 mm per turn	20 mm	4*512*20/2*1.27 16125.98
Axis III	-180° to 180°	512	84	1 °	90 °	4*512*84*90/360 43008
Axis IV	-180° to 180°	512	19	1 °	90 °	4*512*19*90/360 9728
Axis V	-30 to 30 mm	512	1	1.22 mm per turn	10 mm	4*512*10/2*1.22 8393.443
Axis VI	-180° to 180°	128	16	1 °	180 °	4*128*16*180/360 4096

4.4.2 Position Accuracy

To move an object to a specified final position, the object must first be set in motion (accelerated), moved and then decelerated. Finally, the specified position must be maintained against any interfering forces. All these operations require a motor torque that must be made available to the position control system. Ultimately, position control relies on the lower level current control loop for creating the necessary torque.

Here what happens is the master system sends motion commands to the motion controller. They are processed by the path generator that calculates intermediate positions at given time intervals (1ms on EPOS systems) on the path to reach the final position. These set values are transmitted to the position control loop, which, by comparison with the actual position, determines the set (or demand) values for the current controller. In the EPOS the current control loop has a cycle time of 0.1 ms or 10 kHz.

On a well-tuned system, end positions are reached to within 1 encoder quad count (qc). As soon as the position is more than 1 qc off the target position control loop will take corrective actions.

It must be noted that, the system reaction on deviations from the required positions depends on the torque and speed capabilities of the system and also on the friction, mass inertias and the set feedback and feed-forward parameters. The controller parameter values have been established during the tuning process based on the system properties during tuning process.

The other important position accuracy aspect is how the target position is reached. Again, the results depend on how the system was tuned and which aspect of accuracy was given the most weight during the tuning: e.g. no position overshoot permitted, or very fast motion regardless of overshoot. Based on the application of the device in high precision microsurgery, obviously the tuning as it was discussed in chapter three has been done with no overshoot permitted.

CHAPTER 5 : CONCLUSION

In this chapter, the main conclusions are briefly discussed and recommendations for future work are presented.

5.1 Conclusions

In this thesis work, first it was explained how robotic surgery can benefit ophthalmic surgery, and then different ophthalmic surgeries were studied in order to understand required workspace, function and precision. Based on those requirements a robotic system with 6 DOF is proposed and designed.

This system features a 350 mm ring that can hold up to three different instrument manipulators and provide a rotary motion about the optic axis of the eye, φ , a linear motion in the radial direction that determines the entry point to the eye, r , the double-rotor mechanism that provides a remote center of motion for the surgical manipulator at the entry point to the eye with two lateral rotations about the entry point to the eye θ_1 and θ_2 , a stroke motion along the instrument axis, Z , and a rotation about the instrument axis, ψ .

This design is compact and offers wide range of motion and is capable of adjusting remote center of motion. The theoretical accuracy of the system is calculated to be 10 μm . Also, forward and inverse kinematics of the system were defined.

Since the robotic surgery system discussed in this thesis is composed of 6 DC motors as the main elements of its control system, a linear model to analyze the dynamic behavior of the system was developed.

To run 6 DOF system, six miniature DC motors with PID controllers were used. To perform the motion control of the system, first the dynamic behavior of a DC motor was studied and modeled and then the position and velocity transfer functions were derived. By substituting real motor parameters and specifications into those transfer functions, the control diagrams were plotted and used to study the behavior of the system. Then, a proportional controller was introduced to the system and the process of finding the best proportional gain to give the optimum response with fast convergence and no overshoot was investigated. To manually tune the system, the same process was repeated to find the effect of integral and derivative gains. Eventually, the PID control system with the optimum gains was found and the result was presented.

The elements of the control system including encoder, controller modules, Controller Area Network (CAN) and the controller software were discussed next. To organize communications among these elements, a network which is here called CAN bus needed to be created. In order to do so an ID number must be assigned to each motor controller as a node on the system and then it must be properly connected to the rest of nodes on network. The method of node number assignment and wiring is also described.

The EPOS Studio is the graphical user interface that can provide connection with controllers in order to run the motors. This program offers a wide range of control options and provides a very well defined master for this robotic system.

Motion commands must be expressed in understandable unit which is here called quadratic counts (qc). This unit is the smallest motion achievable by this design that pretty much guarantees the required precision for intended surgeries. The procedure to convert all sorts of motions to qc unit is tabulated and presented.

5.2 Recommendations for Future Work

The surgical robot which is the purpose of this thesis is fully assembled and motorized and ready to function at this point. The master control system offers options to adjust the magnitude of velocity, acceleration and deceleration in order to produce smooth motion on all joints.

As mentioned earlier in the body of this thesis, this robot is the slave robot, which does the operation. The input to the robot should be generated by a master robot. The main part of the future work is to design a master robot and the communication method between these two robots. Although designing such system requires another comprehensive research, the studies on the eye surgery procedures in this work can be used as a baseline for designing an efficient and ergonomic master console.

Since the forward and inverse kinematics of the slave robot is fully determined, the mapping between surgeon moves on the master console and slave robot manipulation could be achieved with high accuracy.

The control system that receives the input from the master console and translate it to slave robot movement can be designed in a way to scale the surgeon's hand motion down and also reduce the tremor.

The next step would be design verification and validation based on regulatory agents standards such as FDA. The results of the verification and validation tests can be used to optimize the design of the proposed slave robot.

BIBLIOGRAPHY

- American Society of PeriAnesthesia Nurses. *PeriAnesthesia Nursing Core Curriculum: Preprocedure, Phase I and Phase II PACU Nursing*. 2nd. Edited by Lois Schick and Pamela E. Windel. Saunders Elsevier, 2012.
- Ang, W. T., C. N. Riviere, and P. K. Khosla. "An active hand-held instrument for enhanced microsurgical accuracy." *MICCAI*, 2000: 878 – 886.
- Babuska, Robert, and Stefano Stramigioli. "Matlab and Simulink for Modeling and Control." 1999.
- Biographicsmedia. *Eye anatomy drawing sketch image illustration*. 2006. <http://www.biographixmedia.com/human/eye-anatomy.html> (accessed 2013).
- Bourla, D. H., J. Hubschman, M. Culjat, A. Tsirbas, A. Gupta, and S. D. Schwartz. "Feasibility study of intraocular robotic surgery with the da vinci surgical system." *Retina* 28, no. 1 (2008): 154-158.
- Buckley, Sean. *iRobot, InTouch Health unveil RP-VITA*. 2012. <http://www.engadget.com/2012/07/24/irobot-intouch-health-unveil-rp-vita-telepresence-robot/> (accessed 2013).
- Butter, M., et al. *Robotics for healthcare, Final report*. TNO Informatie en Communicatietechnologie: TNO, The Netherlands Vilans, The netherlands EuroAct, Japan FhG ISI, Germany VTT, Finland, 2008.
- CAN in Automation (CiA)*. March 1992. <http://www.can-cia.org/>.
- Charles, S., et al. "Dexterity-enhanced telerobotic microsurgery." *International Conference on Advanced Robotics (ICAR)*, 1997: 5-10.
- Craig, J. J. *Introduction to Robotics: Mechanics and Control*. 3rd. New Jersey: Pearson Prentice Hall, 2004.

- Dalvand, Mohsen Moradi, and Bijan Shirinzadeh. "Motion control analysis of a parallel robot assisted minimally invasive surgery/microsurgery system (PRAMiSS)." *Robotics and Computer-Integrated Manufacturing* 29, no. 2 (2013): 318-327.
- Das, H., H. Zak, J. Johnson, J. Crouch, and D. Frambach. "Evaluation of a telerobotic system to assist surgeons in microsurgery." *Computer Aided Surgery* 4, no. 1 (1999): 15-25.
- Diana, Alison. *12 Advances in Medical Robotics*. January 27, 2011. <http://www.informationweek.com/healthcare/patient-tools/12-advances-in-medical-robotics/d/d-id/1095720?> (accessed November 1, 2013).
- Donelson, S. M., and C. C. Gordon. *Matched anthropometric database for U.S. Marine corps Personnel: summery and statistics*. Newton Centre, MA: Geo Centers Inc., 1996.
- Encyclopedia of Surgery*. 2015. <http://www.surgeryencyclopedia.com/La-Pa/Ophthalmologic-Surgery.html> (accessed 2015).
- Engel, D., J. Raczkowski, and H. Worn. "A safe robot system for craniofacial surgery." *IEEE International Conference on and Automation (ICRA)* 2 (2001): 2020-2024.
- Farsi, M., K. Ratcliff, and M. Barbosa. "An introduction to CANopen." *Computing & Control Engineering Journal, Volume 10*, 1999.
- Fei, B., W. Sing Ng, S. Chauhan, and C. K. Kwoh. "The safety issues of medical robotics,." *Reliability Engineering & System Safety* 73, no. 2 (2001): 183-192.
- Ghani, Khurshid R., Quoc-Dien Trinh, Jesse Sammon, Wooju Jeong, Ali Dabaja, and Mani Menon. "Robot-assisted urological surgery: Current status and future perspectives." *Arab Journal of Urology* 10 (March 2012): 17-22.
- Gheshmi, Siavash. *Design and Development of Advanced Surgical Robotic Systems*. Orono: M.Sc. thesis, University of Maine, 2012.

- Goroll, Allan H., and Albert G. Mulley. *Primary Care Medicine: Office Evaluation and Management of the Adult Patient*. 6th. Philadelphia: Lippincott Williams & Wilkins, 2012.
- Graham, J. H. "Research issues in robot safety." *IEEE International Conference on Robotics and Automation (ICRA)* 3 (1988): 1854-1855.
- Gupta, P.K., P.S. Jensen, and E. de Juan. "Surgical forces and tactile perception during retinal microsurgery." *Medical Image Computing and Computer-Assisted Intervention*, 1999: 1218-1225.
- Hendrix, Ron. *Robotically assisted eye surgery: A haptic master console*. Eindhoven: Ipskamp Drukkers, TU/e, 2011.
- Higson, G. R. *Medical Device Safety: The Regulation of Medical Devices for Public Health and Safety*. 1st. London: The Institute of Physics (IoP), 2001.
- Ida, Yoshiki, Naohiko Sugita, Takashi Ueta, Yasuhiro Tamaki, Keiji Tanimoto, and Mamoru Mitsubishi. "Microsurgical robotic system for vitreoretinal surgery." *International Journal of Computer Assisted Radiology and Surgery* 7, no. 1 (January 2012): 27-34.
- InTouch Health Inc. *RP-VITA Remote Presence Robot*. 2012. <http://www.intouchhealth.com/products-and-services/products/rp-vita-robot/> (accessed November 2013).
- Intuitive Surgical. *Intuitive Surgical - da Vinci Products FAQ*. 2013. http://www.intuitivesurgical.com/products/products_faq.html (accessed 2013).
- Jagtap, A. D., and C. N. Riviere. "Applied force during vitreoretinal microsurgery with handheld instruments." *26th Annual IEEE International Conference of the Engineering in Medicine and Biology Society*. 2004. 2771-2773.
- Jensen, P.S., P.K. Gupta, and E. de Juan. "Quantification of microsurgical tactile perception." *Engineering in Medicine and Biology, 21st Annual Conf. and the*

Fall Meeting of the Biomedical Engineering Society BMES/EMBS Conference. 1999.

Kafader, Urs. "Motion Control for Newbies." 2014.

Katalyst Surgical Inc. *Ophthalmic Instruments.* 2013.
<http://www.katalystsurgical.com/> (accessed 2013).

Kazanzides, P. "Safety design for medical robots." *IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, 2009: 7208-7211.

Kazazides, Peter, and Paul Thienphrapa. "Centralized processing and distributed I/O for robot control." *Technologies for Practical Robot Applications.* IEEE International Conference, 2008. 84-88.

Kirby, Daniel B. "Presentation of Instruments—Instruments for Cataract Surgery." *Trans Am Ophthalmol Soc*, 1947: 423-432.

Kohanbash, David. "CAN bus (CANopen & CiA) for Motor Control." 2014.

Kumar, R., et al. "Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation." *IEEE International Conference on Robotics and Automation (ICRA)* 1 (2000): 610-617.

Lambaudie, E., et al. "Role of robot-assisted laparoscopy in adjuvant surgery for locally advanced cervical cancer." *European Journal of Surgical Oncology (EJSO)* 36, no. 4 (April 2010): 409-413.

Lynch, Patrick J. "Medical Illustrator." *Lateral eye and orbit anatomy with nerves.* 2006.

Mahpeykar, Navid. *Design, Development and Fabrication of an Advanced high precision Robotic System for Microsurgery.* Orono: M.Sc. thesis, University of Maine, 2013.

Maxon. "Application Notes Collection." 2013.

- Maxon. *CANopen Basic Information*. Maxon Corporation, 2008.
- Meenink, Thijs. *Vitreo-retinal eye surgery robot: sustainable precision*, PhD diss. Eindhoven: Ipskamp Drukkers, 2011.
- Meshram, P.M., and R.G. Kanojiya. "Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor." *Advances in Engineering, Science and Management (ICAESM)*, 2012.
- ORBIS Cyber-Sight. *Manual Small Incision Cataract Surgery*. New York: ORBIS International, 2006.
- Patel, K. "Robotics the future of surgery." *International Journal of Surgery* 6, 2008: 441-442.
- Pieper, D. *The Kinematics of Manipulators Under Computer Control*. Stanford University: Unpublished, 1968.
- Pieper, D., and B. Roth. "The Kinematics of Manipulators Under Computer Control." *Proceedings of the Second International Congress on Theory of Machines and Mechanisms 2* (1969): 159-169.
- Pitcher, John D., Jason T. Wilson, Tsu-ChinTsao, Steven D. Schwartz, and Jean-Pierre Hubschman. "Robotic Eye Surgery: Past, Present, and Future." *Computer Science Systems Biology, Published Online*, 2012: doi:10.4172/jcsb.S3-001.
- R. Leigh, James. *Control theory*. 2004.
- Rahimy, E., J. Wilson, T. C. Tsao, S. Schwartz, and J. P. Hubschman. "Robot-assisted intraocular surgery: development of the IRISS and feasibility studies in an animal model." *Eye* 27 (August 2013): 972-978.
- Robinson, Stephen A. "Cataract Extraction." Department of Surgery, GHSU. *Cataract Extraction*. Augusta, GA, 2012.

Shahinpoor, M., and S. Gheshmi. *Robotic Surgery: Smart Materials, Robotic Structures, and Artificial Muscles*. New York and London: Taylor and Francis Publishers, PAN Stanford Publishing, 2014.

Tameesh, M. K., R. R. Lakhanpal, and et al. "Retinal vein cannulation with prolonged infusion of tissue plasminogen activator (t-PA) for the treatment of experimental retinal vein occlusion in dogs." *American Journal of Ophthalmology* 138, no. 5 (2004): 829-839.

Taylor, R. H., et al. "Taming the bull: safety in a precise surgical robot." *Fifth International Conference on Advanced Robotics, ICAR*, 1991: 865-870.

Ueta, T., et al. "Robot-Assisted Vitreoretinal Surgery: Development of a Prototype and Feasibility Studies in an Animal Model." *Ophthalmology* 116, no. 8 (August 2009): 1538-1543.

APPENDIX A: MATLAB CODE FOR FORWARD KINEMATICS

To find symbolic forward kinematic equations:

```
clear all
clc;
syms Phi Psi Th1 Th2 R Z H
T10=[cos(Phi) sin(Phi) 0 0;
     -sin(Phi) cos(Phi) 0 0;
     0 0 1 H;
     0 0 0 1]
T21=[1 0 0 0;
     0 0 -1 R;
     0 1 0 0;
     0 0 0 1]
T32=[cos(Th1) -sin(Th1) 0 0;
     -sin(Th1)*cosd(45) -cos(Th1)*cosd(45) cosd(45) -H;
     -sin(Th1)*cosd(45) -cos(Th1)*cosd(45) -cosd(45) H;
     0 0 0 1]
T43=[cos(Th2) -sin(Th2) 0 0;
     sin(Th2)*cosd(22.5) cos(Th2)*cosd(22.5) -sind(22.5) 0;
     sin(Th2)*sind(22.5) cos(Th2)*sind(22.5) cosd(22.5) 0;
     0 0 0 1]
T54=[1 0 0 0;
     0 cosd(22.5) -sind(22.5) -Z*sind(22.5);
     0 sind(22.5) cosd(22.5) Z*cosd(22.5);
     0 0 0 1]
T65=[cos(Psi) -sin(Psi) 0 0;
     sin(Psi) cos(Psi) 0 0;
     0 0 1 0;
     0 0 0 1]

T60=T10*T21*T32*T43*T54*T65
```

To find numerical forward kinematic equations with input arguments in degrees:

```
function T60=F_Kine(Phi,Th1,Th2,Psi,r,Z)

Phi=deg2rad(Phi);
Th1=deg2rad(Th1);
Th2=deg2rad(Th2);
Psi=deg2rad(Psi);

H=1;
R=H+r;
T10=[cos(Phi) sin(Phi) 0 0;
     -sin(Phi) cos(Phi) 0 0;
     0 0 1 H;
     0 0 0 1];
T21=[1 0 0 0;
     0 0 -1 R;
     0 1 0 0;
     0 0 0 1];
```

```

    0 0 0 1];
T32=[cos(Th1) -sin(Th1) 0 0;
     -sin(Th1)*cosd(45) -cos(Th1)*cosd(45) cosd(45) -H;
     -sin(Th1)*cosd(45) -cos(Th1)*cosd(45) -cosd(45) H;
     0 0 0 1];
T43=[cos(Th2) -sin(Th2) 0 0;
     sin(Th2)*cosd(22.5) cos(Th2)*cosd(22.5) -sind(22.5) 0;
     sin(Th2)*sind(22.5) cos(Th2)*sind(22.5) cosd(22.5) 0;
     0 0 0 1];
T54=[1 0 0 0;
     0 cosd(22.5) -sind(22.5) -Z*sind(22.5);
     0 sind(22.5) cosd(22.5) Z*cosd(22.5);
     0 0 0 1];
T65=[cos(Psi) -sin(Psi) 0 0;
     sin(Psi) cos(Psi) 0 0;
     0 0 1 0;
     0 0 0 1];

T60=T10*T21*T32*T43*T54*T65;

```

end

APPENDIX B: MATLAB CODE FOR INVERSE KINEMATICS

```

function [Phi Th1 Th2 Psi R Z]=Inv_Kine(T)

r11=T(1,1); r12=T(1,2); r13=T(1,3);
r21=T(2,1); r22=T(2,2); r23=T(2,3);
r31=T(3,1); r32=T(3,2); r33=T(3,3);
Px=T(1,4); Py=T(2,4); Pz=T(3,4);

Phi=atan2(Px-Pz*r13/r33,Py-Pz*r23/r33);
R=sqrt((Px-Pz*r13/r33)^2+(Py-Pz*r23/r33)^2);
Z=Pz/r33;

Rz=[cos(Phi) -sin(Phi) 0 0;
sin(Phi) cos(Phi) 0 0;
0 0 1 0;
0 0 0 1];

%Calculating the rotated {Tool} to obtain {Tool} in {RCM}
Tprime=Rz*T;
r11p=Tprime(1,1); r21p=Tprime(2,1); r31p=Tprime(3,1);
r13p=Tprime(1,3); r23p=Tprime(2,3); r33p=Tprime(3,3);

Th2=2*asin(sqrt((r23p+r33p-1)/0.41421356));

%NOTE: The positive value is chosen
%negative Th2 results in another set of solutions for Th1 and
Psi

A=(0.35355339+0.35355339*cos(Th2));
B=0.38268343*sin(Th2);
C=r13p;

E=(1-sin(Th2/2)^2);
F=0.2705980501*sin(Th2);
D=r23p-0.7071067812*sin(Th2/2)^2;

Sine_Th1=(2*B*D-B*E+E*C)/(2*F*B+A*E);
Cosine_Th1=C/B-A/B*Sine_Th1;

Th1=atan2(Sine_Th1,Cosine_Th1);

Cosine_Psi=((cos(Th1)*cos(Th2)-
0.92387953*sin(Th1)*sin(Th2))*r11p...
+(0.27059805*sin(Th2)+0.65328148*cos(Th1)*sin(Th2)+0.70710678*cos(Th2)*
sin(Th1))*r21p...
+(0.27059805*sin(Th2)-0.65328148*cos(Th1)*sin(Th2)-
0.70710678*cos(Th2)*sin(Th1))*r31p);

Sine_Psi=((0.14644661*sin(Th1)-0.92387953*cos(Th1)*sin(Th2)-
0.85355339*cos(Th2)*sin(Th1))*r11p...
+(2.4142136*sin(0.5*Th1)^2*sin(0.5*Th2)^2-1.0*sin(0.5*Th1)^2-
1.7071068*sin(0.5*Th2)^2-0.65328148*sin(Th1)*sin(Th2)+1.0)*r21p...

```

```

+(sin(0.5*Th1)^2-
2.4142136*sin(0.5*Th1)^2*sin(0.5*Th2)^2+0.70710678*sin(0.5*Th2)^2+0.653
28148*sin(Th1)*sin(Th2))*r31p);

```

```

Psi=atan2(Sine_Psi,Cosine_Psi);

```

```

Phi=Phi*180/pi;

```

```

Th1=Th1*180/pi;

```

```

Th2=Th2*180/pi;

```

```

Psi=Psi*180/pi;

```

```

disp('          Phi          Th1          Th2          Psi          r          Z')

```

```

disp([ Phi Th1 Th2 Psi R Z])

```

```

end

```

APPENDIX C: MATLAB CODE TO PLOT RESPONSES OF THE DC MOTOR

SYSTEM

```
function Arezoo=PIDC(Kp,Ki,Kd,t)

J=0.13*10^-7;      % kgm2 %
b=10*10^-8;       % Nms/rad %
K=0.01;           % Nm/A %
R=27.4;           % Ohm %
L=0.399*10^-3;    % H %

s = tf([1 0],1);
Gv = K/((L*s + R)*(J*s + b) + K^2);
Ga = Gv/s;

Gv.InputName = 'Voltage';
Gv.OutputName = 'Velocity';
Ga.InputName = 'Voltage';
Ga.OutputName = 'Angle';

% G = [Gv; Ga];
G = ss(Ga);
set(G,'c',[0 1 0; 0 0 1],'d',[0;0],'OutputName',{'Velocity';'Angle'});

% Plotting system properties: response to step input, impulse input
% and bode plot(frequency domain)

% figure(1);
% subplot(2,1,1); step(Ga, 0.03);title('Step
Response','FontSize',10);ylabel('Position
(rad)','FontSize',10);xlabel('Time','FontSize',10);
% subplot(2,1,2); impulse(Ga, 0.03);title('Impulse
Response','FontSize',10);ylabel('Position
(rad)','FontSize',10);xlabel('Time','FontSize',10);
% set(gcf,'Color',[0.85,0.7,1]);
% set(gcf,'InvertHardcopy','off');
% print('-dmeta');

% figure(2);set(gcf,'Color',[0.85,0.7,1])
% subplot(2,1,1); step(Gv, 0.03);title('Step
Response','FontSize',10);ylabel('Velocity
(rad/s)','FontSize',10);xlabel('Time','FontSize',10);
% subplot(2,1,2); impulse(Gv, 0.03);title('Impulse
Response','FontSize',10);ylabel('Velocity
(rad/s)','FontSize',10);xlabel('Time','FontSize',10);
% set(gcf,'InvertHardcopy','off');
% print('-dpng');

% figure(3);
% subplot(2,1,1); bode(Ga);
% subplot(2,1,2); bode(Gv);
% set(gcf,'Color',[0.85,0.7,1]);
% set(gcf,'InvertHardcopy','off');
% print('-dpng');
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % % Proportional controller

% % % Position feedback
Gc = feedback(Ga*Kp,1);% Angle
Gc.InputName = 'Desired Angle';

% figure(5)
% rlocus(Gc)
%     set(gcf,'Color',[0.85,0.7,1]);
%     set(gcf,'InvertHardcopy','off');
%     print('-dpng');

% figure(6);
% step(Gc,0:0.00001:t);title('Step
Response','FontSize',10);ylabel('Position
(rad)','FontSize',10);xlabel('Time','FontSize',10);
%     set(gcf,'Color',[0.85,0.7,1]);
%     set(gcf,'InvertHardcopy','off');
%     print('-dpng');

% % % Velocity feedback
Gc = feedback(Gv*Kp,1);% Velocity
Gc.InputName = 'Desired velocity';

% figure(7); step(Gc,0:0.0005:t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% PID controller

C = tf([Kd Kp Ki],[1 0]);

% Position feedback
Gc = feedback(Ga*C,1);% Angle
Gc.InputName = 'Desired Angle';

figure(8); step(Gc,0:0.00001:t);title('PID Step
Response','FontSize',10);ylabel('Position
(rad)','FontSize',10);xlabel('Time','FontSize',10);
    set(gcf,'Color',[0.85,0.7,1]);
    set(gcf,'InvertHardcopy','off');
    % print('-dpng');

% Velocity feedback
Gc = feedback(Gv*C,1);% Velocity
Gc.InputName = 'Desired velocity';

% figure(9); step(Gc,0:0.0005:t);
end

```

BIOGRAPHY OF THE AUTHOR

Arezoo Ebrahimi was born in Booshehr Port, Iran on January 3rd, 1985. She attended Bentolhoda Sadr high school in Shiraz. She then studied Industrial Engineering at Sharif University of Technology and graduated in 2006 with a Bachelor's degree in Industrial Engineering.

In September 2013, she started her studies at University of Maine to obtain a Master of Science degree in Mechanical Engineering. She joined the graduate program as a teaching and research assistant at Biomedical Engineering and Advanced Robotics Laboratory.

Arezoo is a candidate for the Master of Science degree in Mechanical Engineering from the University of Maine in December 2015.