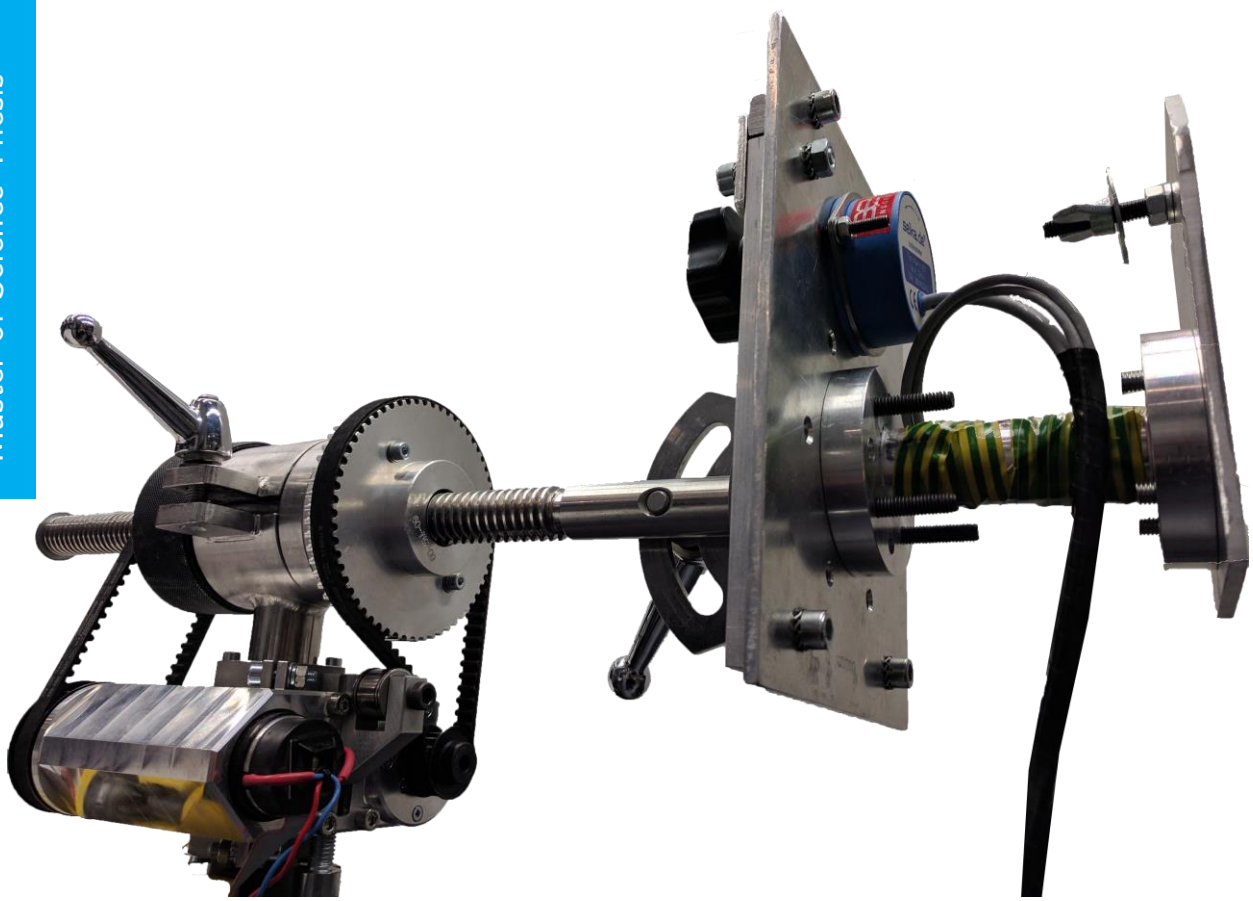
CONFIDENTIAL

Hip Fracture Reduction Robot

Design and Low-level Control

Shekhar Gupta

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

MSCCONFIDENTIAL

Hip Fracture Reduction Robot
Design and Low-level Control

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Shekhar Gupta

November 30, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Hip Fracture Reduction Robot

by

Shekhar Gupta

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: November 30, 2016

Supervisor(s):

Prof. Dr. Robert Babuška

Mr. Maarten Griffioen

Reader(s):

Dr. ir. J. W. van Wingerden

Dr. R. Ferrari

Dr. T. M. A. L. Klem

Summary

Conditions such as osteoporosis, in combination with the fall-related injuries, lead to hip fractures, especially among the older population. Fracture reduction is the surgical procedure used to restore the broken femur to its correct alignment. The hip fracture reduction procedure is performed by the surgeon, by first applying the traction and then rotating the patient's leg either in 'endo' or 'exo' direction. The surgical procedure for pertrochanteric femur fracture does not involve any lateral forces and therefore a robot with two Degree(s) of Freedom (DoF) can perform the reduction by imitating the reduction procedure as done by the surgeon. The goal of this thesis is to develop a compact, low-cost and specialized two DoF robotic solution with low level control.
A 2-DoF robot is designed to deliver linear motion of 100 mm and rotation of 360°. DC motors combined with a planetary gearbox and belt transmission system is used to achieve the actuation. A two axis force/torque sensor allows accurate measurement of the force and the torque at the end effector.
The Hip Fracture Reduction Robot (HFR) demands interaction between the robot and the patient's leg; hence various types of interactive control have been implemented and compared. In impedance control, the robot imposes a force to move the leg; virtual mass, spring, and damper systems are used to generate this force. The purpose of impedance control is to obtain a dynamic relationship between the position and the force, rather than controlling either of these variables independently. Admittance control is designed not just to track the force trajectory, but also to regulate the mechanical admittance of the robot. By controlling the mechanical admittance of the robot, it can be constrained to have a particular mechanical behavior. On the other hand, direct force control will keep the contact force constant; this is particularly beneficial for situation with fluctuations in contact forces.

A KUKA robotic arm is programmed to behave like a spring and a damper system to mimic the patient's leg during fracture reduction. The programmed KUKA robotic arm and the spring based lower limb phantom are used to test the fracture reduction robot. Initial experiments attaching the device to KUKA have indicated great potential for the device. The fracture reduction robot can apply the required amount of force and the torque to achieve the desired fracture reduction.

Table of Contents

Acknowledgements

I would like to thank my supervisor Prof. Dr. Robert Babuška for his support and guidance during this thesis.

I would also like to thank Mr. Maarten Griffioen for the stimulating discussions, invaluable support, and guidance during the difficult phase of this project.

I extend my sincere gratitude to Dr. T. M. A. L. Klem for facilitating this project.

This thesis would not have been possible without the help and support of Mr. J. Frankenhuyzen, Mr. C. J. Slinkman, and Mr. W. J. M. van Geest. I would like to extend my sincere gratitude for their guidance.

I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Finally, I would like to place on record my appreciation for all those who have helped me during this endeavor.

Delft, University of Technology — Shekhar Gupta
November 30, 2016

Chapter 1

Introduction

Surgical history dates back to as early as the stone age, during which period a surgical procedure called trephination was performed on the human skull (see Figure 1-1). Archaeological evidence which dates back to 10,000 years, suggests that a hole was drilled into the human skull during the human's lifetime. Even though the reason behind performing such a procedure might have been superstitious, it is believed to have some clear benefits to the patient.

Figure 1-1: Trephination performed on human skull during stone age, Source: Ancient History [31]

Thousands of years later, during the 19^{th} century, the popular modern day surgery, which is perceived safe, originated. Before this period the surgical procedure was very crude and had

a mortality rate well above 50% due to infections. The discovery of anesthesia and aseptic surgical technique created a paradigm shift in the way surgery was performed. Similarly, development in the field of surgical robotics can pave the way to create a new era in the surgical history. The innovations in minimally invasive surgery, due to the advent of surgical robots demonstrates this statement. However, surgical robotics is still an emerging technology and is still in its infancy.

"Arthrobot" is the first known surgical robot and was used in an orthopedic surgical procedure in the year 1983 [41]. The robot was developed by a team of biomedical professors and engineering students in collaboration with an orthopedic surgeon. Since then, researchers all around the world have performed extensive research on surgical robotics. However, robots designed for the surgical procedure are far from being used in everyday procedures at the hospitals across the world. Issues such as patients' safety, the risk of malfunction or failure, bulkiness, and incompatibilities with conventional instruments are a few barriers to the adoption of surgical robotics. Furthermore, the disadvantages such as prolonged hours, due to the technical difficulties in setting up robotic instruments, higher initial and maintenance cost of robotic devices than the standard tools, discourage the hospitals from adapting to the new technology. However, robotic surgery is an emerging technology which has both its period of support and resistance from the surgeons' as well as from the patients' side. As with any technology, people need time to become familiar with, trust it and finally accept it.

From its inception, surgical robots have been envisioned to aid and enhance the surgeon's capability during surgery. In the recent past, impressive surgical robots, such as the Robodoc® [15] and Da Vinci have been used to perform surgical procedure extensively. Robodoc® was designed to machine the femur with greater precision in the hip replacement surgeries and was the first surgical robot to get approved by the Food and Drug Administration (FDA). However, lately Robodoc® has gained negative publicity. Robodoc® achieves a similar level of precision, compared to surgeons but with the cost of technical complications [25]. The Da Vinci surgical system was designed to facilitate complex surgery using a minimally invasive approach while being remotely controlled by a surgeon. Even though Da Vinci has been used in over 200,000 operations, it has received criticism for its cost and surgical performance [30]. Given the above facts, it is crucial for the technologists to understand and realize that the surgical robotics needs to be impartially evaluated for its drawbacks and benefits before putting it on the clinical trial.

A project has been initiated by the Delft Center for Systems and Control (DCSC) department at TU Delft in collaboration with Sint Franciscus Gasthuis hospital to develop a Hip Fracture Reduction Robot (HFR). The guideline mentioned above was at the forefront for evaluating the potential benefits of such a robot. The following sections will describe the motivation and objective for this project. The designed robot and the low-level control implementation will be described in detailed, from Chapter 2 onwards.

1-1 Motivation

Fall-related injuries are among the highest injuries experienced among the older population [42]. They are the source of major public health problems in aging societies worldwide. One of the main reasons for this is osteoporosis. Osteoporosis is a medical condition in which

the bones become brittle and fragile from the loss of tissue, typically as a result of hormonal changes, or deficiency of calcium or vitamin D. It becomes more common with age and is the most common reason for a broken bone. A study conducted by Hartholt et al. [1] has shown that from the beginning of 1981 till 2008, about 355,320 patients aged $\geq$ 65 years were admitted due to a hip fracture in the Netherlands. It is expected that this number will continue to increase in the future due to a rising number of older people in the population. Today, the treatment of choice for femur fracture is the standard technique of intramedullary nailing.

1-1-1 Intramedullary Nailing

Intramedullary nailing is a surgically optimized minimally invasive surgery. It is done in two phases, namely: fracture reduction and intramedullary fixation. The complete description of intramedullary nailing procedure is as follows:

Fracture reduction: When the femur breaks, the muscles that surround the injured area contract and displaces the broken fragments of the bone. Figure 1-2 shows the x-ray image of a broken hip. The broken bone needs to be aligned to its correct anatomical position, and this procedure is called as fracture reduction. During reduction, the patient is first made to lie down on a fracture table in a supine position as shown in Figure 1-3a. The patient's leg is attached to the fracture table, and a shoe is put on the foot corresponding to the broken femur. The alignment of the broken femur is achieved by applying traction and endorotation or exorotation through the patient's leg as shown in Figure 1-3b. The surgeon first pulls and then rotates the patient's leg about the longitudinal axis until the fracture is aligned. The alignment is confirmed by fluoroscopic imaging in axial and anterior-posterior views. Figure 1-3c shows the final state of patient's leg.

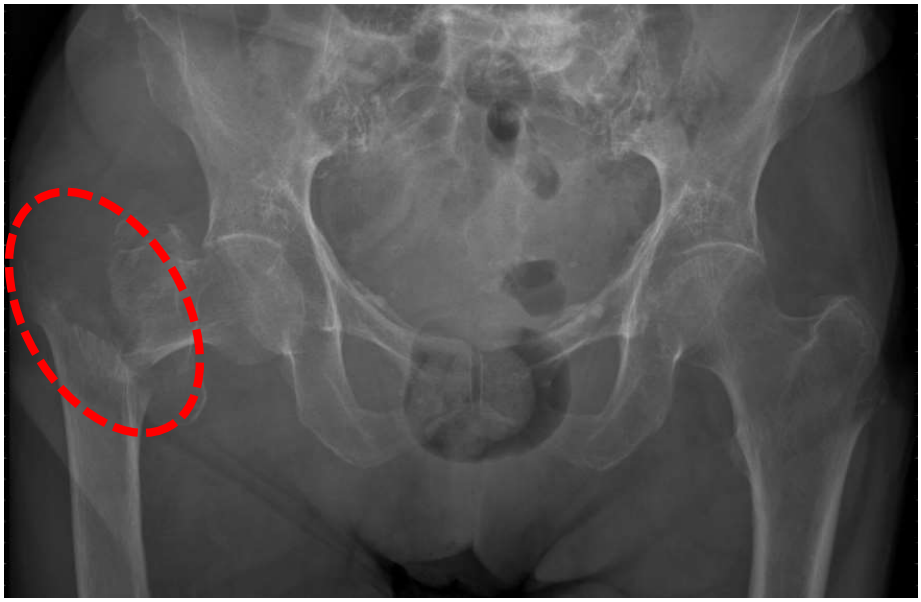
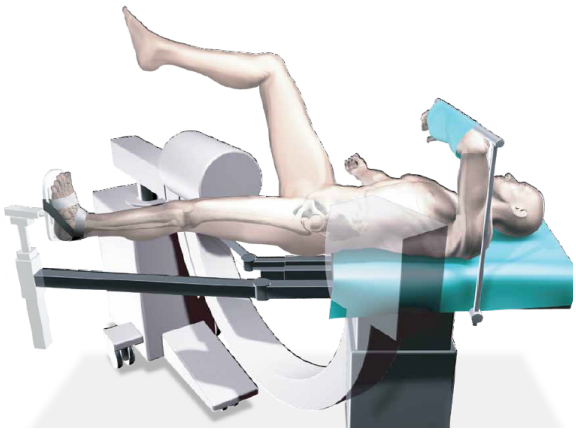
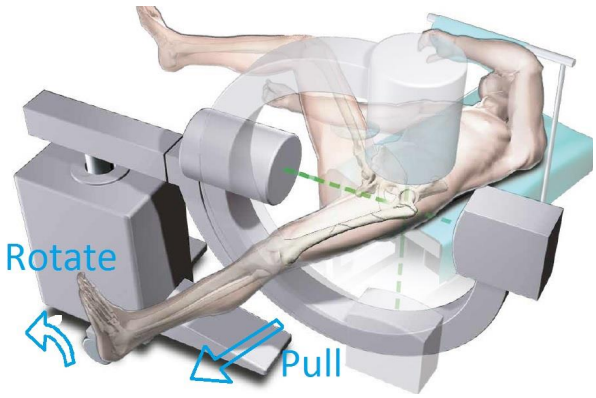


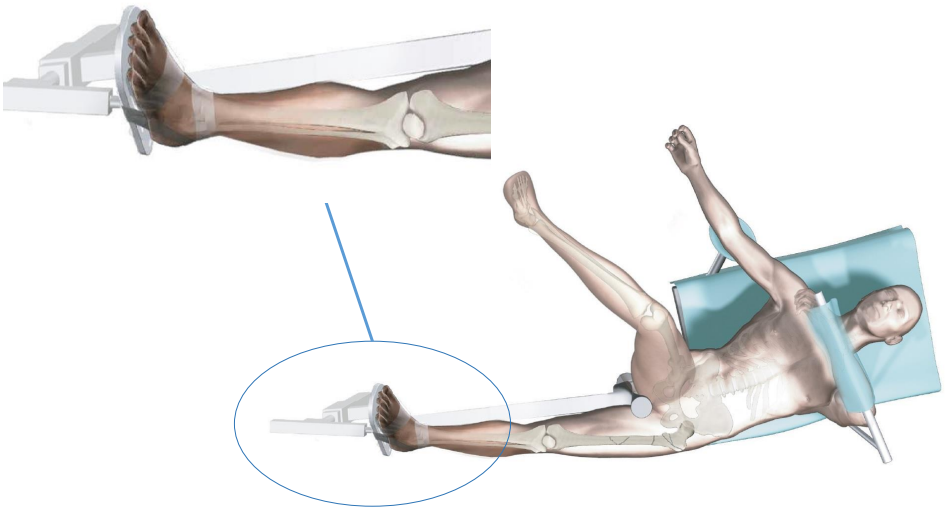
Figure 1-2: Fluoroscopic image of a broken femur on the left vs. a solid femur on the right



(a) Patient's position during fracture reduction.



(b) *Pull* and *Rotate* action to align bones.



(c) Patient's leg position after reduction.

Figure 1-3: Stages in Fracture reduction procedure. Source [19]

Intramedullary Fixation: Intramedullary fixation is the process of attaching the bone fragments to each other by forcing and fixating a nail in the medullary cavity. Subsequently, the two major bone fragments are aligned according to their correct anatomical positions. Figure 1-4 shows the gamma nail and set screw insertion into the femur's medullary cavity during the intramedullary fixation procedure. The whole process is supervised using X-ray imaging. A detailed description of this surgical procedure can be found in [24].

This procedure has many advantages such as preservation of soft tissues, high primary union rate and small chances of surgery-induced infection. Intramedullary fixation allows immediate full weight bearing postoperatively and hence it is popular for the treatment of the proximal femur.

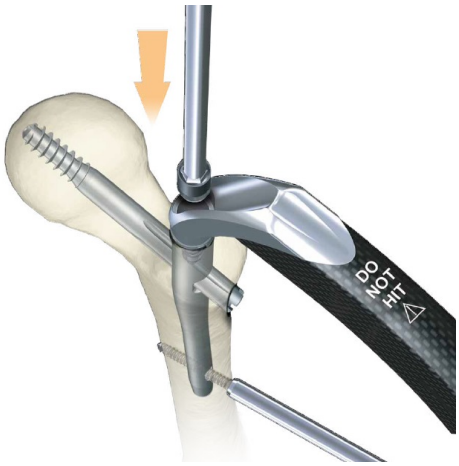
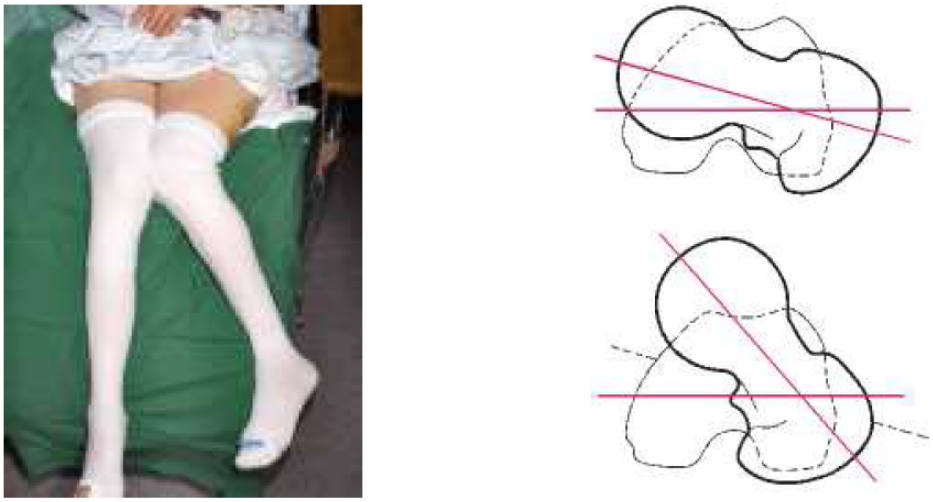


Figure 1-4: Gamma nail assembly during intramedullary fixation Source: Stryker [19]

1-1-2 Drawbacks of State of the Art Fracture Reduction Procedure

Besides its advantages, fracture reduction obtained from the aforementioned procedures has various drawbacks which are widely discussed in the literature. One of the biggest disadvantage being malalignment (ref. Figure 1-5), which has a high impact on functional biomechanics. Blomfeldt et al. [28] reported that 47% of elderly people had to re-operated due to malalignment. Other disadvantages include high radiation exposure to the patient and operating room (OR) staff and soft tissue trauma to the patient. Radiation exposure to the patient is limited to one operation. However, the surgeon and the OR staff are repeatedly exposed to radiation during multiple procedures. It is estimated that on an average between 158 and 316 seconds of X-ray imaging radiation is common [29]. The forces during fracture reduction can go up to 600 N [10] which can lead to physical fatigue for the surgeons and more importantly such large forces cannot be applied accurately using the manual procedure, causing soft tissue trauma to the patient.

The intramedullary nailing procedure for treating fractures is being used from the time of the second world war [27], and the technology has remained relatively constant. Now with the advancement in the technology, it is time to overcome some of these disadvantages which could not have been done before. One of the emerging technology that one can use is surgical robotics. Since robots are more dexterous than humans, malalignment can be avoided or minimized. It is possible to reduce the soft tissue trauma and radiation exposure on the patient and the OR staff by automating the whole procedure. However replacing current state of the art reduction technique with a robot does raise safety issues. The robot should be equipped with multiple levels of safety features to address this problem. The safety issues and requirements have been discussed in detailed in the Section 2-1. However, before going any further, it is important to understand the existing technology in the field of fracture reduction robot. The next section presents a short survey on fracture reduction robots.



(a) (b)

Figure 1-5: (a) An example of rotational malalignment leading into a non-physiological condition. (b) Malalignment caused by the relative rotation of axis between the femoral neck and the knee joint.

1-2 State-of-the-art Fracture Reduction Robots

Even though remarkable research has been done in medical robotics in the recent past, the robotic fracture reduction is one field which has seen a little research. In fact, up until now there are no commercially available fracture reduction robots.

Bouazza-Marouf et al. [3] were the first to use robotic assistance during femur fracture reduction. However, they did not develop any robot for performing the fracture reduction procedure. Instead, they focused on developing an image-guided robot for the intramedullary fixation procedure.

Füchtmeier et al. [7] were also working on Robotic-assisted fracture reduction. They made a robot which was manually controlled i.e. they have used a trackball of a 3-D mouse to control the robot. Their research focused on evaluating industrial robots so that they can be employed in the process of fracture reduction. To assess safety requirements, they studied on the reaction of the industrial robots in the case of overload. Even though the robot was able to achieve precise fracture reduction, many modifications were made to the robot as well as the OR. Their robot was designed particularly for long bone reduction and they have discontinued their work. Their study was based on the assumption that only 240 N of force is required during reduction, according to the measurement done by Huiskin [10], the forces can go up to 600 N. Industrial robots, when used for such high payloads can weigh up to 1000 kgs [33] and are extremely bulky.

1-2-1 Non-invasive Femur Fracture Reduction Robot

Warisawa et al. [4] were working on the automated femur fracture reduction robot (Figure 1-6). Their initial publication in 2004, was on automating non-invasive fracture reduction. However, their automated fracture reduction procedure did not satisfy the error limits of 2 mm in translation direction, which was required for the navigation system to reduce the fracture. Due to this reason, it could not be used for the hip fracture reduction surgery, and subsequently, a modified robot based on invasive reduction was developed. However, a control mode called 'power assist mode' is accepted by medical doctors and has given excellent efficiency and operability of the surgical operation. In this mode, the robot augments the surgeon's force to generate the power required for fracture reduction. Their approach to automated fracture reduction has been described below.

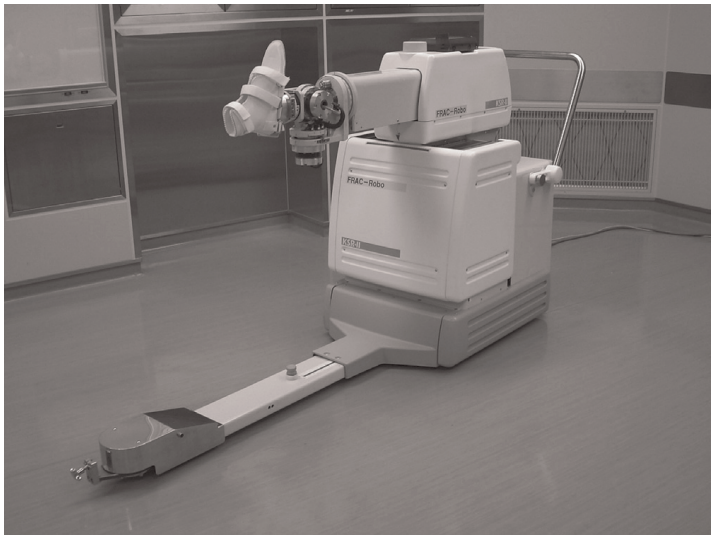


Figure 1-6: Robotic surgical assistants devloped by Warisawa et al. [4]

The femur reduction is carried out, by reconstruction of 3D bone image from CT scan. After the 3D reconstruction of the bone, path planning for the reduction of fracture fragments is done. Both of these steps are conducted on the computer pre-operatively. Then a six-axis robot with fail-safe mechanism is used to carry out the fracture reduction. A touch panel type surgeon robot interface is used to provide all the necessary information to the surgeon. Safety is provided both by hardware and software fail-safe mechanisms.

1-2-2 Invasive Femur Fracture Reduction Robot

Joung et al. [6] designed an automated femoral fracture reduction robot with high accuracy, but requires a small incision on the patient's leg to perform fracture reduction. This new robot was obtained by modifying the robot described in Section 1-2-1. The foot fixator device was replaced by a customized jig (Figure 1-7(b)). This new robot which performs open fracture reduction was called FRAC-Robo and has six degrees of freedom. To achieve the reduction, a customized jig is directly attached to the bone of the patient by inserting two screws.

Figure 1-7(c) provides the kinematic model of FRAC-Robo. Three rotational axes intersect

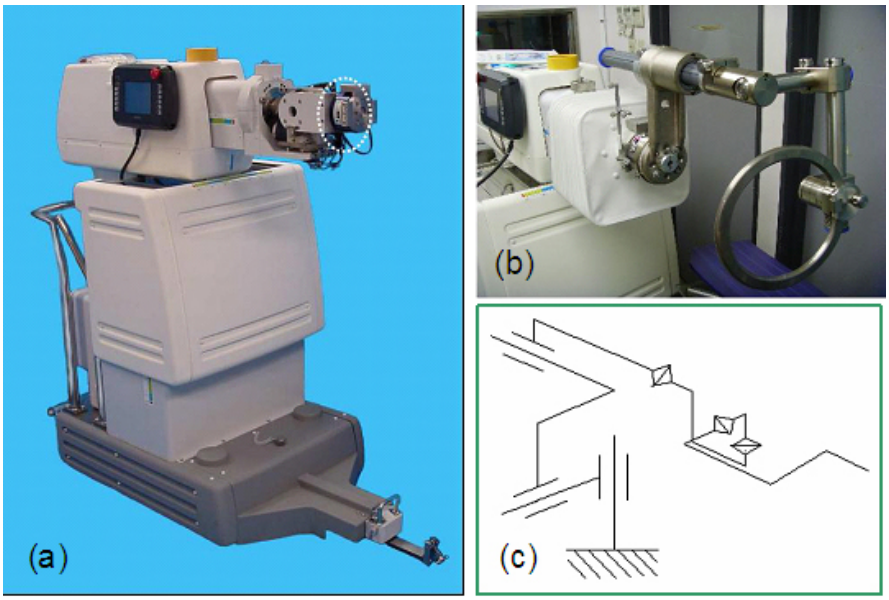


Figure 1-7: FRAC-Robo(a)6 DoF FRAC-Robo system (b)Customized jig connected with the distal fragments(c)Kinematic model of the robot. Source: Joung et al. [6]

at one point, from kinematics point of view this makes control easy. Four steerable wheels are attached for easy movement of the robot which weighs 315 kg and is of size $640 \times 1084 \times 1317$ mm ($width \times length \times height$). The jig is made from carbon fiber and duralumin to make it lightweight and sturdy.

FRAC-Robo is integrated with a navigation system that tracks the relative position of the bone fragments and generates the reduction path. The three-dimensional surface data of the bone fragments are estimated from CT scans of the injured hip, before surgery. A reference frame is attached to the pelvis, and the position of the femur head is calculated using the pelvis tracking and then using kinematic knowledge of hip joint. The position of the distal bone is calculated using the robot's position since the distal bone is rigidly attached to the robot using a customized jig. The reduction path is generated using the acquired relation of the bone fragments. The generated reduction path is interactively edited by the surgeon, and the generated path is transmitted to the robot. Control is done using position control, and the control loop is closed using vision.

This setup was tested on a femoral neck fracture test model, which simulated the influence of the muscles around the femur. It showed good results but the evaluation method still needs to be further developed to confirm the safety and accuracy of the FRAC-Robo. Figure 1-8a and 1-8b shows the test femur model used to carry out reduction using FRAC-Robo.

1-2-3 Serial Robot for Fracture Reduction

Hung and Lee [14] designed a serial robot that can be mounted on the fracture reduction table. Figure 1-9 shows their robotic setup. It contains two supporters, the hip support, and the knee support. The hip support supports the patient's pelvis and provides a counter force to avoid distal displacement of the patient during traction. The knee supporter functions

(a) Test femur model before reduction **(b)** Test femur model after reduction

Figure 1-8: Femoral neck fracture test model used by Joung et al. [6] to test FRAC-Robo

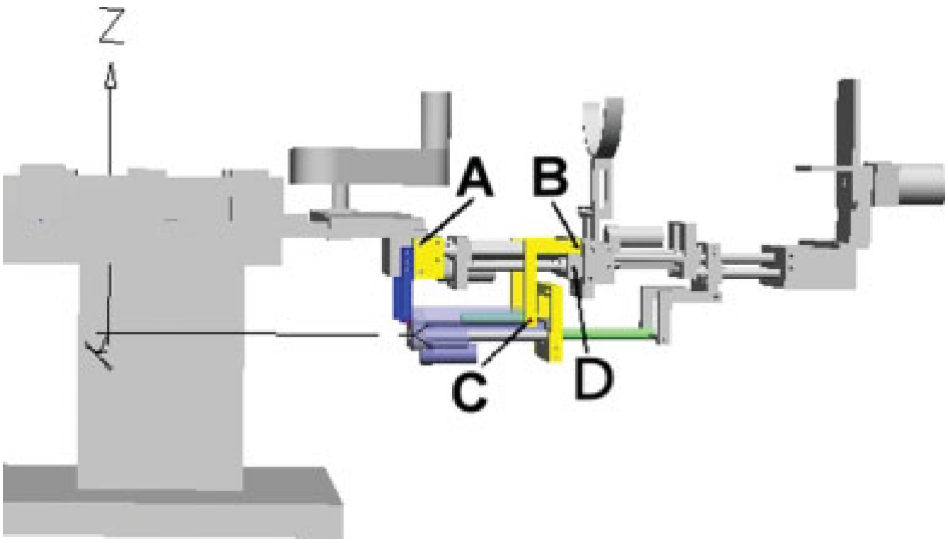


Figure 1-9: Fracture reduction robot by Hung and Lee [14]

as a stabilizer for the lower limb and prevents motion during surgery. The reduction of the proximal femur is performed by application of the traction force and the rotational torque along the longitudinal axis of the leg. To achieve thigh traction, the actuator connecting at location D (Figure 1-9) will elongate, causing an increase in the length of the proximal segment. The rotation of the foot is achieved by rotating the foot plate. Here 180° of rotation is allowed via a motor. The limitation of their robot is that it has not been automated yet, and control is done manually by motor control, through an open switch relay. They have not described the load capacity of the manipulator, and have not provided any fracture reduction results.

1-2-4 Parallel Robot for Femur Shaft Reduction

Graham et al. [8] have developed a six degree of freedom parallel robot which can be attached to the reduction table to achieve femur fracture reduction. Figure 1-10 shows the prototype of the integrated robot system. In this robot, a foot holster is used to attach the platform to

the recipient's leg to perform reduction, or the robot can be attached directly to the femur with a pin through the femur head.
The control of the robot is done through the navigation trajectories consisting of a number of discrete points which are processed by a six-axis control card and are amplified to move the platforms with six individual ball screw actuators. Optical encoders on each actuator provide position feedback for closed loop control of the trajectory. This robot was designed

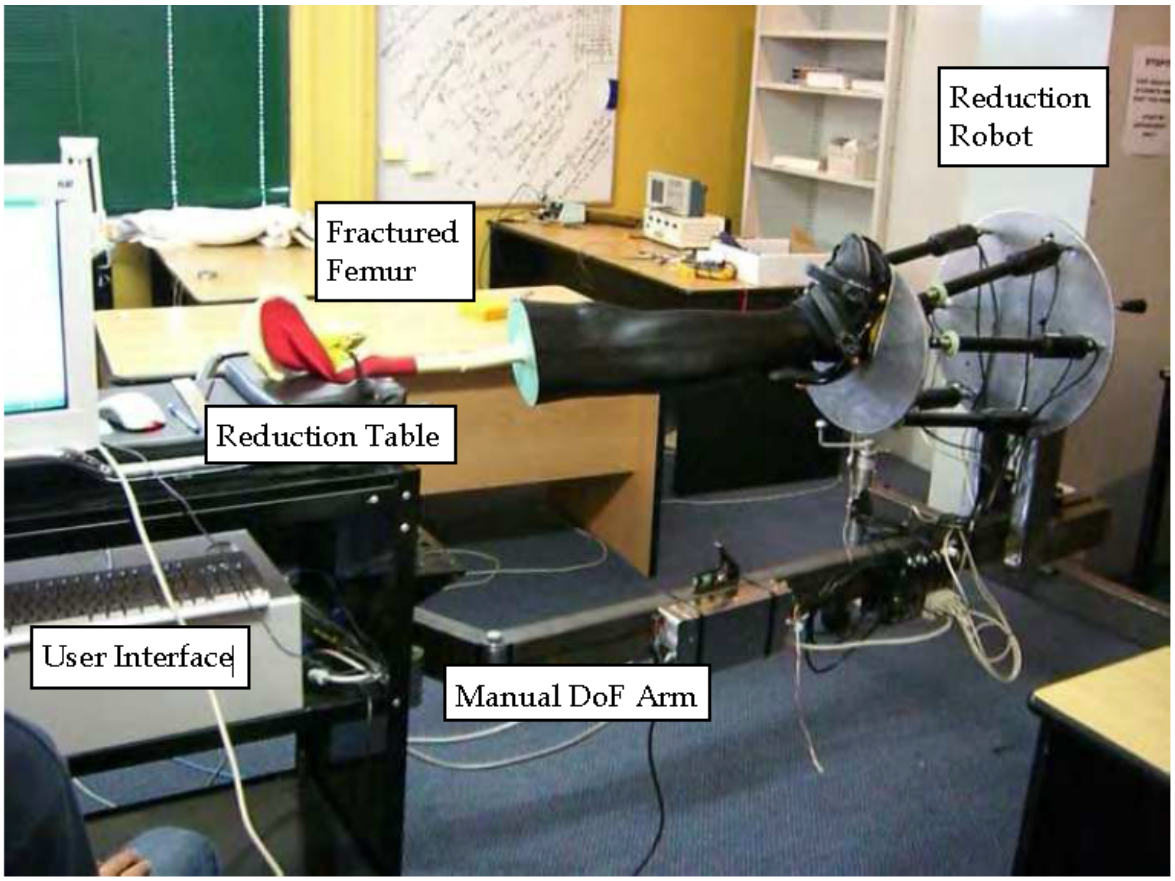


Figure 1-10: A prototype of Integrated Fracture reduction system of the Parallel Robot. Source: Graham et al. [8]

specially to meet the needs of existing OR. Instead of using CT scan to get the 3D data, 2D fluoroscopic images were used, which is usually available in hospitals. Keeping in mind that the space in the OR is limited they used a parallel robot, which gives high payload to weight ratio in comparison to the serial robots. The design requirements and accuracy were derived by taking the OR requirements and the model of the existing fracture table into account. Table 1-1 shows the design requirements and accuracy that were used to develop the robot. The x, y and z are the translation motion requirements and θ, ϕ and φ are the rotations about the three axes. A computer model of the parallel robot is shown in Figure 1-11 and it comprises of six powered links joining the base and the top plate. They have not described the load capacity of the robot or the control method that they want to use. It was mentioned that position control is currently used, but they intend to implement force control in their future work. They did not mention the load capacity of their robot neither did they show

Figure 1-11: CAD model of a parallel robot used by Graham et al. [8]

Table 1-1: Design requirements for the fracture reduction robot

Parameter	Range (mm/deg)	Accuracy (mm/deg)
x	± 100	< 1
y	± 100	< 1
z	± 150	< 1
θ	± 10	< 1
ϕ	± 10	< 1
φ	± 30	< 1

any results. The limitation of the parallel platform is its limited workspace. For example, the maximum rotation movement that it can provide is $\pm 30^\circ$ whereas, according to the measurement performed by Huiskin [10], the rotation along the longitudinal axis during reduction can go up to $\pm 60^\circ$.

From the above literature survey, it is clear that most of the research work in this field is done using industrial six degrees of freedom robotic arms. However, the industrial robots are extremely bulky and expensive. Even the small robot arm's which are commercially available today do not meet the force (600 N) and the torque (18 Nm) requirements [10], to perform the fracture reduction procedure. Due to the space and the mobility constraints of the OR, bulky and cumbersome robotic arms are not practical. Also, additional modifications need to be done on the existing industrial robot to cater to the unique set of the surgical procedure requirements. There exists no commercial autonomous robot for closed femur fracture reduction. It was also found out that in the literature only development of six degrees of freedom robot for fracture reduction is discussed. A six DoF robot can perform multiple types of fracture reduction. However, our area of focus is a very specific type of pertrochanteric femur fracture reduction which does not involve any lateral movement. To obtain the bone alignment of such a fracture the surgeon first *pulls* and then *rotates* the patient's leg about the longitudinal axis until the fracture is aligned [26]. Therefore a two degrees of freedom robot can perform the reduction by imitating the reduction procedure as done by the surgeon. To correctly dimension the system and to select the actuators, we need to know the force, torque, and the displacement during reduction. Hence the measurement

data from Huiskin [10] and Ravishankar [32] will be used as the reference while deriving the requirements and constraints for the robot.

1-3 Objectives

Given the absence of any commercially available fracture reduction robot and with drawbacks mentioned above in employing industrial robots to perform femur fracture reduction, developing a customized robot should be the next logical step. The goal of this thesis is to develop a compact, low-cost and specialized two Degree(s) of Freedom (DoF) robot solution with low-level control. The robot needs to be mounted on the existing fracture table and should perform the particular task of *pull* and *rotate* action in a compliant manner.

Moreover, this design needs to preserve the current manual fracture reduction mechanism in the fracture table; this is important because if the robot malfunctions, the surgeon should still be able to carry out the fracture reduction without worrying about the robot.

1-4 Outline

Given the above objectives, this master thesis has been divided into two parts. In part I, the design of the HFR is discussed. Chapter 2 starts by presenting the pre-design analysis and clearly defines the design objectives and the requirement for the robot. The detailed design description and the hardware development of the first prototype of 2 DoF HFR are furnished in Chapter 3. In part II, conceptual implementation and experimental validation of the appropriate compliant control strategies is discussed. Chapter 4 gives a detailed overview of the state-space modeling of the robot and identification of parameters. Chapter 5 begins with the implementation of different types of compliant controllers on the Simulink simulation platform, followed by Chapter 6 which provides a comparison of these controllers between the test setup and the simulation. Finally, the conclusions and the recommendations for the future improvements are proposed in Chapter 7.

Part I

Design of Hip Fracture Reduction Robot

Chapter 2

Design Objectives, Constraints and Requirements

This Chapter lists the requirements for the design of the robot and the constraints that have been imposed to define boundaries for certain aspects of the design. The constraints and requirements are categorized in Sections 2-1 to 2-4. After the implementation of design prototype, the prototype will be tested on to check if it meets all the requirements and constraints. In Chapter 7, the results of these tests will be discussed.

The task of automating, femur fracture reduction by designing a robot capable of being deployed in an operating room (OR), presents many design challenges. Balancing the requirements of each of the technical requirements involved in the construction, control, and safety of operation is one of the most relevant ones. The equilibrium between these can be seen from the following sections.

2-1 Safety

Safety is of utmost importance while designing surgical robots. Medical robots are in constant contact with the patient, unlike industrial robots. So proper safety design and full proof fail-safe mechanism should be built into them. Safety standards can be divided into hardware and software based [35]. Safety should include software and hardware force limiters. That is, excess force and torque applied while performing fracture reduction can harm the tissues surrounding the bone. Hence fracture reduction robots must always have mechanical fail-safe such that when the force or torque exceeds the threshold value the end effector should get detached from the actuator thereby removing the excessive force. When the excessive force is removed, the system should return to its normal state. There should also be an emergency stop, which should stop any further motion of the robot and at the same time maintain the position before the stop. Standard safety features which are used in the medical robots are watchdog timers, dead-man switch, force monitoring, encoder redundancy and software motion limits. Apart from these safety features, the robot can be designed to have less drastic

effects in case of breakdown by adding features like increased robot stiffness and gear ratio or applying restriction on unnecessary movements as the design requirement.

2-2 Mechanical Constraints and Requirements

As mentioned earlier the robot needs to mimic the current state of the art fracture reduction procedure which involves the *pull* and *rotation* action of the leg. It is possible to achieve this task by designing a two Degree(s) of Freedom (DoF) robot. To accurately dimension the system and select the actuators, actual measurement data regarding the force, torque and the distance moved by the leg during reduction, is required. Data from Huiskens [10] and Ravishankar [32] were used to address these requirements. Huiskens [10] carried out measurements and reported that the maximum force, torque and the angle of rotation of leg during the fracture reduction procedure were 548 N, 17.4 Nm and 60° respectively. Ravishankar [32] also performed measurement such as distance and angle moved by the leg during the procedure with respect to time. From these measurements, it was possible to find the range and speed of motion of the robot to be designed. An accuracy of 2 mm for linear and 2° for rotation of bone as seen from a 2D fluoroscopic image is medically accepted [18] as a good reduction pose. The complete mechanical design requirements are presented in Table 2-1.

2-3 Electrical and Software Constraints and Requirements

In consideration of the fact that the medical robots directly interact with the patients and surgeons, electrical constraints form the most significant part of the design constraints of the medical robots. The design objective should address the requirement that the robot should be able to perform the fracture reduction autonomously. Hence it is important to have software constraints as well. Some of these constraints are mentioned below:

- Sufficient computing power for simultaneous autonomous control and data processing coming from the sensors.
- Possibility to shut down the robot at any time using an emergency switch
- Electrical and thermal protection (Overheating in motors)
- Protection against excessive currents on motors

2-4 General Constraints and Requirements

User involvement is a fundamental element in the development process of any technology and is particularly important in an environment such as in an OR. During this project, discussion with several surgeons and OR staffs were conducted. To understand the procedure accurately, the surgery was witnessed in person. The feedback from the surgeons suggested that present reduction procedure has many disadvantages and that a robotic solution could be useful in an appropriate setting. But, some of the OR staffs expressed their hesitation in using such

a device. After brainstorming with OR staffs and witnessing surgery, few common themes emerged. These themes included topics such as safety, complexity, current limitation, ease, and understanding of state of the art reduction techniques, etc. These themes have been translated into general constraints and requirements that one should keep in mind before designing the robot and are mentioned below.

2-4-1 Compliance

In today's financial climate, the cost will be a limiting factor for the acceptance of the robotic system by the medical community. It is recommended to develop a cost-effective design, taking into account the safety and hygiene constraints of the OR. The robot should be fully compatible with the OR environment without the need for modifications. This will minimize the total cost of implementation of this robotic system. While designing the robot it should be kept in mind that the end product will be used in an OR. Hence it does not suffice to have a functioning robot, but it should also be simple and safe so that OR staff feels comfortable in using it.

2-4-2 Known Problems of Fracture Reduction Procedure

As discussed in Chapter 1, the state of the art fracture reduction procedure has drawbacks such as malalignment, radiation exposure, and soft tissue trauma. The robot designed should be able to overcome these drawbacks while minimizing the technical complexity. If the results obtained by adopting the robot can be proven beneficial without any bias, then the acceptance of such a robot becomes much easier.

2-4-3 Structure and Location

The designed robot has to be capable of being easily positioned and moved around in the presence of other OR instruments. Similarly, it should provide little to no obstruction to the OR staff when performing the surgery. A requirement such as this implies that it is impractical to use an existing industrial robot as the basis of the system. Most commercially available robots which have the required range of payload capacity, are too large to be used in an OR environment. While, commercial robots of appropriate size and mass cannot provide the required payload. The problem is however simplified by the requirement that the procedure requires just a *pull* and *rotate* action. This action could be achieved by a customized, relatively small 2 DoF robot. From the feedback of the OR staff, it is found that the same OR is used for many other operations. The layout of the OR is dynamic, and the fracture table can be moved if required. Hence it is a requirement that the robot should be designed such that it can be mounted on an existing fracture table.

2-4-4 Scope for Manual Control

The aim of hip fracture reduction robot is to eventually carry out the reduction autonomously, by applying the traction force and rotation torque in a way similar to the procedure currently executed by the surgeon. It is possible that the early prototype will not be perfect. Hence

it is required that the current manual control mechanism of the fracture table need to be preserved. This would come in handy in the case of robot malfunction, as the surgeon can perform the procedure manually.

2-4-5 Data collection

As mentioned earlier, the early prototype of the robot is not without its flaws and will not be perfect. It will not be possible to use the first prototype in an entirely autonomous mode to carry out fracture reduction procedure. However, such a robot can be used to perform the fracture reduction procedure, provided that an appropriate feedback mechanism is established between the surgeon and the robot. If the robot malfunctions or displays the intent to carry out an unnecessary movement, the surgeon should be able to shut down the robot and perform the procedure manually without any hassle. In such cases, the sensor data should be recorded and stored in a database. This data should include both the motion trajectory as well as force trajectories. The data can then be used to analyze and learn from the mistakes. This same data can also be used to look for variations between individual surgeons, and also individual patients. Development of such monitoring concepts can then form a part of the continuing research program.

2-5 Conclusion

This chapter presents the design requirements and constraints that one should keep in mind before design the Hip Fracture Reduction Robot (HFR). Various aspects of design were considered and feedback from the surgeons and OR staffs were taken. Given the above measurements, safety and maneuverability conditions, the mechanical design requirements are furnished below.

Table 2-1: Mechanical design requirements for the design of 2 DoF HFR.

Parameters	Values for Linear Motion	Values for Rotary Motion
Peak Force/Torque	600 N	18 Nm
Range of Motion	200 mm	$\pm 60^\circ$
Speed of Motion	32 mm/s	20°/s
Accuracy	2 mm	2°

Other then the above mentioned mechanical requirements it should be emphasized that there should also be redundancy in safety critical system. The design should include safety features such as, watchdog timers, current limiters, force monitoring, emergency stop button and software motion limits. The performance including the functional workspace, smoothness of movement and robustness should be considered, while designing such a robot. It is important to apply good engineering design concepts and adhere to regulatory requirements, and general medical device standards. Rigorous design analysis needs to be done, precise documentation needs to be maintained and at the same time testing needs to be done as defined by the testing protocol.

Chapter 3

Design Implementation

Chapter 1 presented state of the art femur fracture reduction procedure and its associated drawbacks like malalignment, radiation exposure, and soft tissue trauma. Automation can avoid some of these disadvantages. Chapter 2 dealt with the design requirement for this automation. Since the above procedure requires the robot to perform *pull* and *rotate* action on the patient's leg, a 2 Degree(s) of Freedom (DoF) robot is sufficient to achieve this. The idea is to build an 2 DoF robot that can mimic the surgeon's action by applying a pull and rotate action. This chapter presents the design of one such robot that can be used to automate the fracture reduction procedure. Section 3-1 starts by describing the design of the fracture table, currently used in the operating room (OR). Section 3-2 shows the calculation for the selection of actuation unit. Mechanical design of the entire robot is then demonstrated in Section 3-3. Finally, the electronic design is discussed in Section 3-4. The result is an autonomous 2 DoF robot that can mimic the surgeon's action by applying a *pull* and *rotate* action on the patient's leg.

By using 2 DoF it is possible to provide the *pull* action by making use of a linear actuator and the task of *rotation* can then be achieved by a rotary actuator. Selection of such an actuator not only depends on the mechanical requirements as described in Chapter 2, but also on the cost, simplicity, and safety of its operation. In this regard, some inspiration can be drawn from the current manual fracture reduction setup in this regards. Hence it is important to understand the existing setup first before designing a robot to automate it.

3-1 Fracture Reduction Table

Figure 3-1 shows the current fracture reduction table setup. It consists of following four parts.

Table (A): The patient is made to lie on the table in a supine position during the reduction procedure. Figure 3-2 shows the actual fracture table setup.

Pelvis support (B): This part of the table supports the pelvis of the patient and stops it from moving when the traction is applied.

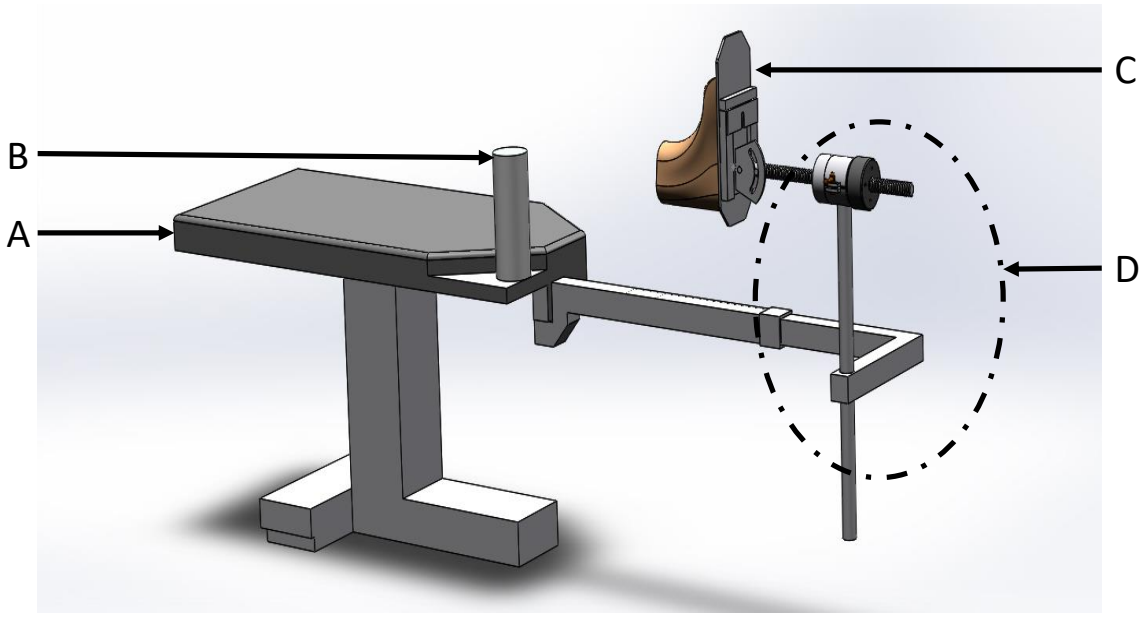


Figure 3-1: Parts of fracture reduction table in drawing. (A) Table, (B) Pelvis support, (C) Foot traction boot, and (D) Actuation mechanism

Figure 3-2: The fracture reduction table supports.(A) Fracture table and (B) Pelvis support

Foot traction boot (C): The foot traction boot is used to fix the patient's foot to the fracture table. Figure 3-3 shows the foot traction boot attached to a fracture table.

Actuation mechanism (D): Fracture table has one rotary actuator and two linear actuators. These actuators are powered manually and are described below. Figure 3-3 shows the actuation mechanism present in the fracture table.

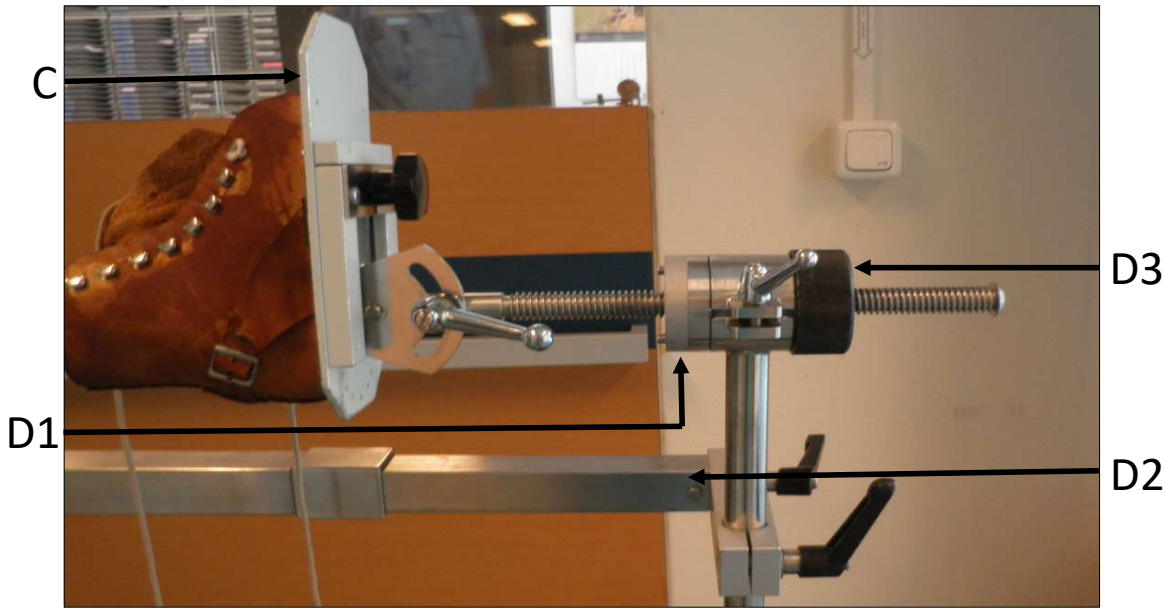


Figure 3-3: Parts of fracture reduction table. (C) Foot traction boot, (D1) Pivot joint, (D2) Gross traction mechanism, and (D3) Fine traction mechanism

Pivot joint (D1): This joint allows rotary movement along the longitudinal axis during the fracture reduction. It is housed with a clamping lever to lock the rotation of the foot at any angle.

Gross traction mechanism (D2): This type of mechanism is used when the surgeon wants to apply gross traction force on the patient's lower limb. It consists of a slider mechanism composed of slide rails over a cantilever supported bar.

Fine traction mechanism (D3): This is a screw thread mechanism with a knurled nut, housed within a traction unit assembly. It is used by the surgeon when a fine or small traction force has to be applied to the patient's lower limb.

The design of fracture table looks simple, yet intuitive. The only drawback being it has to be powered manually. Forces and torque applied during fracture reduction can go up to 600 N and 18 Nm respectively. From a practical point of view, it's hard to apply such a high force and torque to the patient's leg manually and maintain accuracy. The key engineering problem is to design a robot which can apply the required amount of force and torque to the patient's leg while keeping the weight to a minimum. One of the requirement is that robot needs to be mounted on the fracture table. As the weight increases, the stress on the cantilever support bar of the fracture table will increase, making it unstable. Cost is also one of the limiting factors while designing such a robot. Even though pertrochanteric femur fractures have a reasonable incidence, it is very likely, that an expensive robotic system will not be cost-covering for a hospital. Hence the design needs to cost-effective such that the hospitals will be encouraged enough to buy such a solution.

With the above requirements in mind, It is believed that current fracture table design should be the starting point for the Hip Fracture Reduction Robot (HFR) design. It is proposed

that since the existing fracture reduction table can already perform the whole surgery, why not use it as the base model and automate it by automating the actuators. Since the lead screw mechanism and pivot joint are already present in the fracture table, only motors will be required to actuate it. Eliminating the need for any additional actuator mechanism, which makes the design simple, lightweight, cost-effective and compact. The next section presents the calculation for the selection of motors that can be attached to the fracture table to automate it.

3-2 Actuation Unit

3-2-1 Background

For autonomous systems, it is crucial to apply lightweight actuators. Electric motors were chosen to power the lead screw and pivot joint setup. Other alternatives include hydraulics and pneumatic actuators. However, they are used in cases where extremely high torques are required. Hydraulic/pneumatic actuators have problems such as fluid leaks and require frequent refueling making it cumbersome. Problems such as fuel leaks are not suitable for an OR. The electric motor, on the other hand, is compact, powerful and is cost effective, when low to medium torque is required. If the load stalls an electric motor, the windings may burn and permanently damage the motor. Between different types of electric motor Direct Current (DC) motor was selected, since the speed of the motor can be controlled smoothly and the direction of motion can be changed without power circuit switching. Selection of an appropriate DC motor is necessary as an undersized motor can stall which in turn can damage the motor while oversized motor will increase the cost and will be inefficient. In the following section calculation for the selection of a correct size of motors is presented.

3-2-2 Selection of Motor for Linear Motion

The measurement performed during fracture reduction procedure shows that the maximum amount of linear displacement of the leg was 180 mm [32]. Figure 3-4 illustrates the plot of displacement of the patient's leg during one of the surgeries. From the plot, it can be seen that even though the amount of duration taken by the surgeon to perform the overall reduction is around 4 minutes. The time taken to apply the "pull" motion using the gross traction mechanism is around 40 s and the rest of the reduction is carried out using fine traction mechanism. The function of the robot is to imitate the behavior of both the gross and the fine traction motions.

Speed Required

The goal is to make use of the lead-screw present in the fracture table, to achieve the linear motion. Hence, it is necessary to understand the dimension of the lead screw. The lead screw present at the fracture table is a TR20x4 trapezoidal thread screw with a diameter of 20 mm and pitch length of 4 mm.
From Figure 3-4 it can be seen that the speed required for linear motion is small. However, for the homing maneuver of the robot, it is ideal to have a much higher speed. It was decided

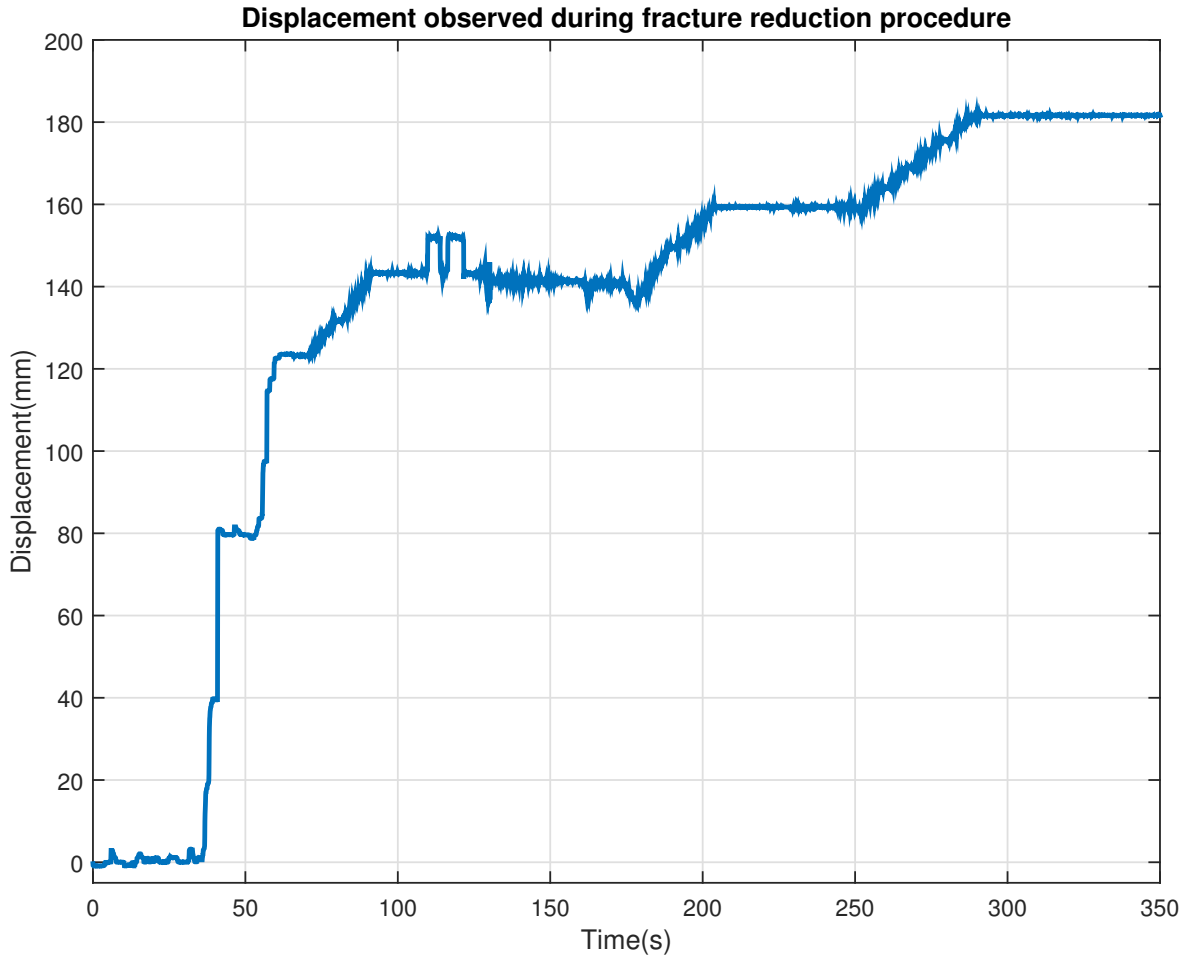


Figure 3-4: Displacement of a patient's leg during fracture reduction. Source: Ravishankar [32]

to select speed of at least 32 mm/s. With this speed, the robot should be able to perform the complete linear motion of 200 mm in 6.25 s. To achieve a linear motion of 4 mm, the knurled nut has to perform one complete rotation. Hence, to achieve a linear speed of 32 mm/s, a motor speed of 8 rotations per second (RPS) or 480 rotations per minute (RPM) will be required.

Torque Required

The amount of input torque required depends on the amount of linear output force that needs to be produced.

Force Calculation

The total output force is a combination of external force, force due to acceleration and frictional force.

$$F_T = F_e + F_a + F_f \tag{3-1}$$

where, F_T is the total linear force, F_e is the external force, F_a is force due to acceleration and F_f is the frictional force.

In the present case, the force due to acceleration is minuscule, $F_a \approx 0.3$ ($F = ma, 10 \times 0.032$) as the acceleration is $0.032\,\mathrm{ms}^{-2}$ and hence it can be neglected. There is no friction in the load (patient's leg) and the friction in the thread screw will be accounted while calculating the torque.

Torque Calculation

$$T = F_T \times \frac{L}{2\pi e}$$

where, F_T is the maximum linear force of 600 N, L is the pitch of the lead screw, 4 mm in this case, and e is the efficiency of the screw (0.6 used for lead screw)

$$T = 600 \times \frac{0.004}{2\pi 0.6} = 0.637 \text{ Nm} \tag{3-2}$$

Power Calculation

The power required by the motor can be calculated using the equation,

$$\text{Power} = \text{Torque (Nm)} \times \text{Angular velocity (rad/s)} \tag{3-3}$$

Speed of 480 rpm in rad/s is $480 \times 0.105 = 50.4$ rad/s
Hence,

$$\begin{aligned}\text{Power} &= 0.637 \times 50.4 \\ &= 32.08 \text{ W}\end{aligned}$$

However, this is the power required before accounting for the efficiency of the timing pulley, gear and motor. After taking the efficiency of timing pulley, gear and motor to be 0.9, 0.75 and 0.75 respectively and a safety factor of 1.2, the power required by the motor will be,

$$\begin{aligned}\text{Power} &= \frac{32.08 \times 1.2}{0.9 \times 0.75 \times 0.75} \\ &= 76.05 \text{ W}\end{aligned}$$

3-2-3 Selection of Motor for Rotary Motion

The goal is to use the revolute joint or pivot joint of the fracture table and a motor to achieve the required rotary motion during fracture reduction.

During the fracture reduction procedure, the angle of rotation and torque applied by the surgeon on patient's leg can go up to $\pm 60^\circ$ and 18 Nm [10] respectively.

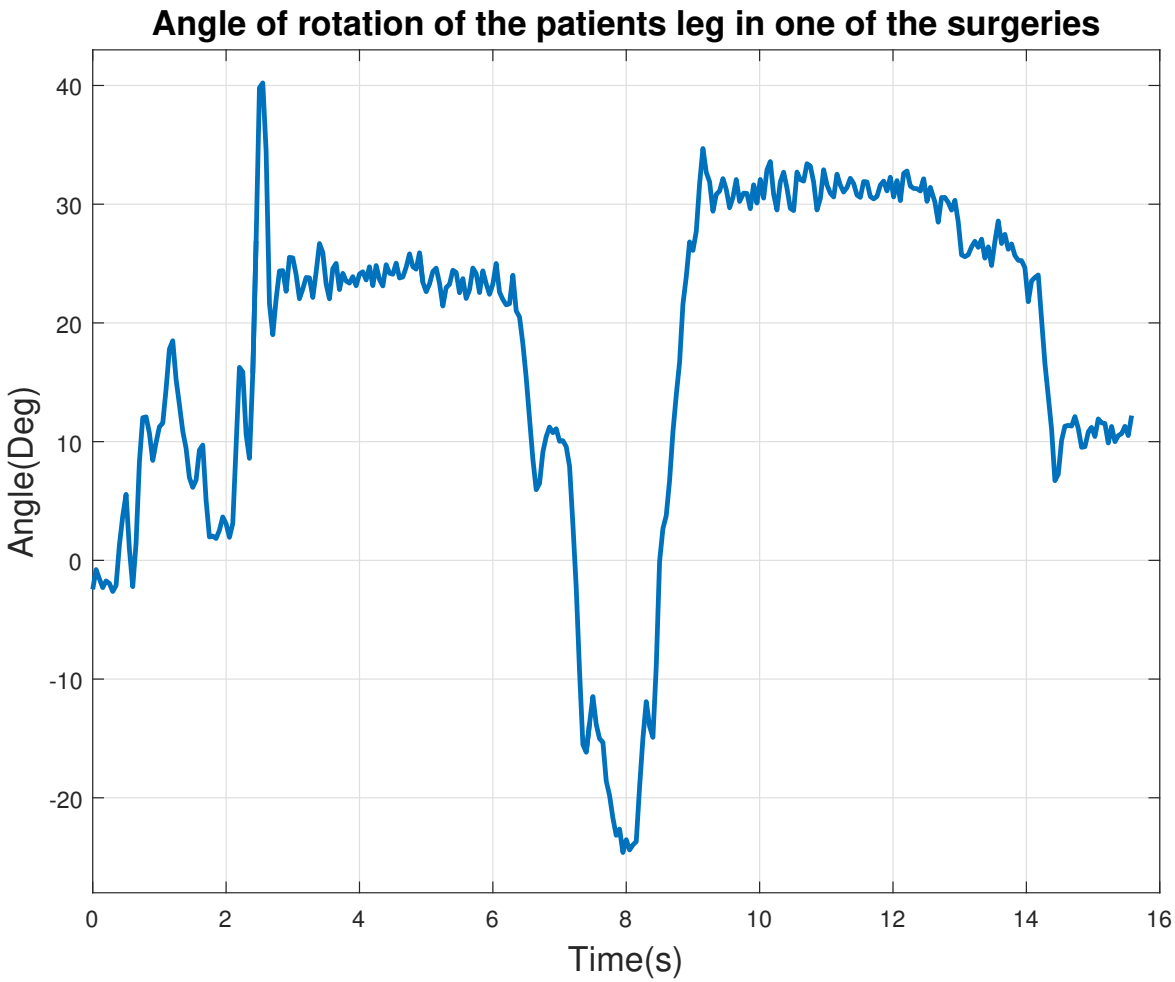


Figure 3-5: Angle of rotation (in degrees) of the patient's leg during fracture reduction. Source: Ravishankar [32]

Speed Required

Figure 3-5 shows the angle of rotation (in degrees) of patient's leg during fracture reduction performed by the surgeon. It can be seen that the surgeon performs this motion on an average around 10 s. Our goal is to achieve the required amount of rotation as fast as possible. To achieve a rotation of 60° in 5 s the rotation speed required is 2 rpm. However, for the homing maneuver of the robot, it is ideal to have a much higher speed. It was decided to dimension the motor such that it can perform the rotary motion in 2.5 s at 4 rpm.

Torque Required

Measurement data of Huiskens [10] has been used to identify the torque requirement. It was found that during the surgery, a maximum torque of 17.4 Nm was applied by the surgeon on the patient's leg. A torque requirement of 18 Nm was thus selected.

Power Calculation

Equation 3-3 can be used to calculate the power. The requirement of 4 rpm speed needs to be converted to SI unit in rad/s, i.e.

$$= 4 \times 0.105 = 0.42 \text{ rad/s}$$

Hence power required will be

$$\begin{aligned} \text{Power} =& \text{Torque} \times \text{Speed} \\ =& 18 \text{ Nm} \times 0.42 \text{ rad/s} = 7.56 \text{ W} \end{aligned}$$

This is the power required to achieve the rotary motion before accounting for the efficiency in pulley, gear and motor. Taking efficiency of pulley, gear and motor to be 0.9, 0.75 and 0.75 respectively, power required by the motor will become,

$$\begin{aligned} \text{Power} =& \frac{7.56}{0.9 \times 0.75 \times 0.75} \\ =& 14.93 \text{ W} \end{aligned}$$

However, this calculation still does not account for the large amount friction present in the revolute joint of the fracture table. To account for this, a safety factor of 3 is used on the power calculation. The final required power of the desired motor is

$$\begin{aligned} \text{Power} =& 14.93 \times 3 \\ =& 44.79 \text{ W} \end{aligned}$$

The required motor can now be selected as the speed, torque, and power requirement of the motors are known. However, choosing a motor is a compromise between the requirement and affordable cost. The torque requirement for the rotary actuator is high, while the speed required is rather low. Motors with high torques are expensive. It is a logical step to buy relatively low torque motor with higher speed and take advantage of speed reduction in power transmission system, to meet the desired speed and torque values. Next section presents the mechanical design and power transmission system that is used to achieve this requirement.

3-2-4 Power Transmission

To transfer the mechanical power from the motors to the actuators of the fracture table, power transmission system is required. Belt transmission system was used for the design. It was chosen because belt transmission systems are quite and smooth, hence suitable for an OR. Gear makes the robot stiff while belt transmission systems lower the overall stiffness of the robots. Since the HFR will be directly interacting with the patients, the stiffness of the robot needs to be small. However, belts tend to slip at high torques, to avoid this, timing belts are selected. The torque requirement for the rotary motion is high. Hence a reduction ratio of four is used between the motor shaft and the final revolute joint shaft of the fracture table.

3-3 Mechanical Design

The hardware configuration of the designed HFR is shown in the Figure 3-6. The 2 DoF HFR comprises of three main parts, a linear actuator, a rotary actuator and a two axis Force/Torque (F/T) sensor. The mechanical design is inspired from the current fracture reduction table setup. The linear actuator and rotary actuator are designed by making use of the existing screw thread mechanism and the revolute joint mechanism present in the fracture table. The actuators are powered by a DC motor combined with the planetary gearbox and belt transmission system. Two motors are used, corresponding to the two actuators. The motors are housed inside the motor frame, which is directly mounted on the fracture table. The required thickness of the motor frame is determined, using a Finite Element Method (FEM) analysis. A F/T sensor is attached to the foot fixator device as shown in Figure 3-6. The F/T sensor enables, accurate measurement of the amount of force and torque being applied to the patient's leg. The F/T sensor also acts as a safety device when the threshold force or torque value is crossed.

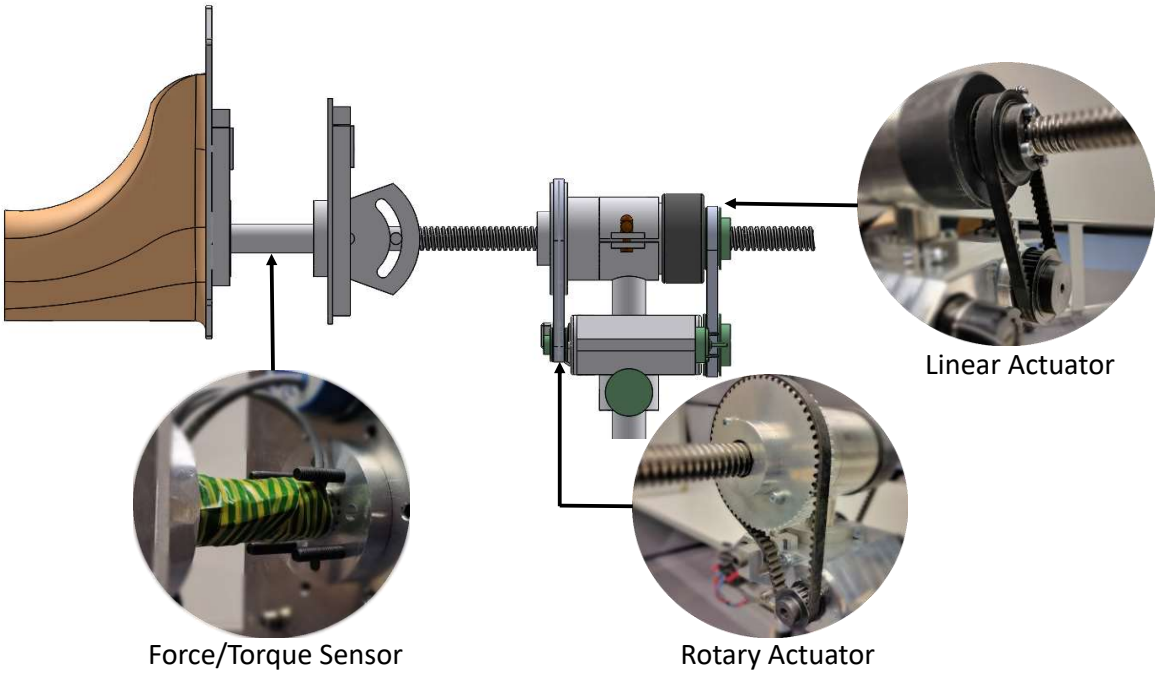


Figure 3-6: Designed HFR consisting of linear and rotary actuator, with a custom made two axes Force/Torque (F/T) sensor

3-3-1 Motor Frame

The motor frame is an ellipse-shaped aluminum block as shown in Figure 3-7. Aluminum is used because it is an inexpensive, lightweight, and relatively strong material. The motors are placed symmetrically at the two focal points of the ellipse, which provides stability to the structure. The aluminum block completely encloses the two motors, making it sturdier as well as safe for the use in an OR. The FEM-analysis (see Section 3-3-2) shows that the thickness of 6mm of the motor frame around the shaft of the motor is sufficient. Figure 3-8

indicates the placement of motors in the motor frame and the complete final assembly of the HFR.

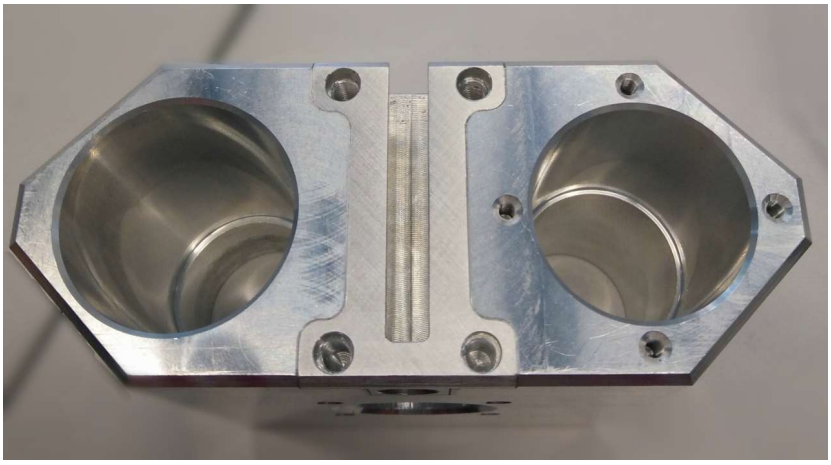


Figure 3-7: The designed motor frame for HFR

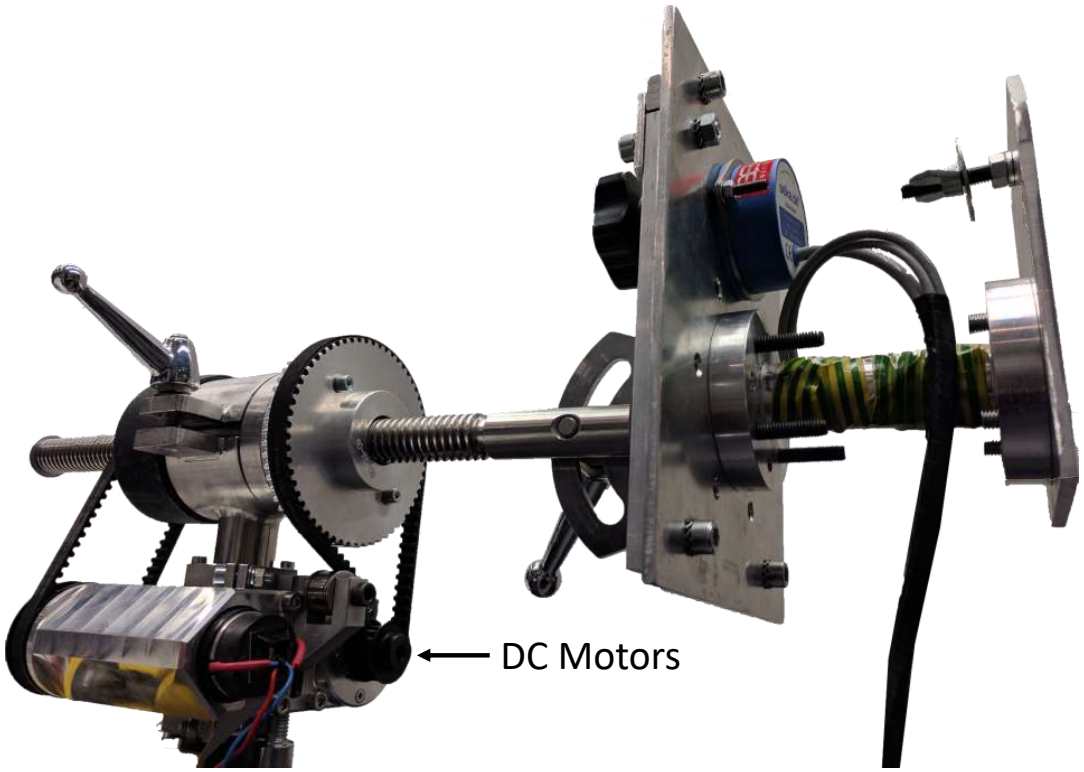


Figure 3-8: The complete designed assembly of the HFR

3-3-2 FEM-Analysis of the Motor Frame

The required thickness of the motor frame was determined with a FEM Analysis using a SolidWorks model. The motor frame was loaded with maximum torque applied by the two motors. The resulting stress distribution is shown in Figure 3-9. It can be seen that the stresses are maximum around the motor mounts. However, this analysis does not represent the real situation accurately because of stress concentration around the bolts. In reality the maximum stress will possibly be higher. The thickness of the motor mount has been designed with safety margins and was narrowed down to 6 mm, to keep the weight low.

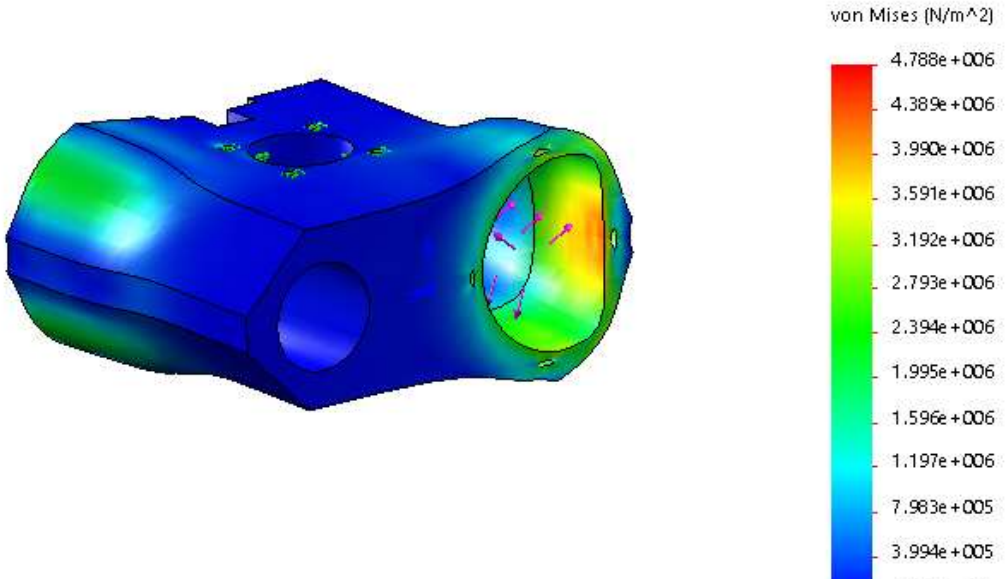


Figure 3-9: Hip Fracture reduction Robot SolidWorks assembly

3-4 Electronics Design

The electronic design of the HFR includes the hardware interface, sensors, data processing and low-level control unit. These are described below.
The hardware interface is made up of two motor drivers based on an H-bridge gate driver integrated circuit (IC). Hence the voltage to the motor driver can be precisely controlled using the Pulse Width Modulation (PWM) technique. For the design of HFR, custom-built motor controllers, developed by Delft Center for Systems and Control (DCSC) were used. The motor drivers can supply current of up to 50 A and voltage up to 24 V. The current and the voltage rating of the motor driver are within the power requirements of the selected motors. Motor drivers also provide safety features like current limit and watchdog timer. These features make motor driver ideal for the use of HFR.
Safety is of utmost importance for the design of HFR. Hence, the HFR is equipped with both position and force measurement sensors. The position measurement is obtained from the optical encoders attached to the two motors, while accurate measurement of force and

torque is achieved from the two axes F/T sensor connected to the foot fixator device. A custom-built F/T sensor was designed for the prototype of HFR. This reduced the overall cost of the first prototype of acHFR. The analog to digital conversion of the signal coming from F/T is done using a Data acquisition (DAQ) box. The designed sensor can measure up to 2000 N of force and 24 Nm torque. In the appendix detailed, discussion of construction and performance of the designed load cell is presented.
A personal computer (PC) with Matlab software is used as the control unit. The values of position and current are collected from the motor driver while the DAQ box collects the F/T data and sends it to the controller for real-time feedback. PC runs the control strategy and sends the output duty cycle of Pulse Width Modulation (PWM) to the motor driver.

3-5 Conclusion

The design of HFR was presented in this chapter. The robot was developed with the aim of providing a compact and low-cost solution to perform the hip fracture reduction procedure. Cost is one of the quintessential requirements while designing such a robot. Hence, motorizing the existing fracture table was found to be the most optimal design solution. Two motors, one for each of the linear and rotary motion are attached to the fracture table. This is done in order to automate the existing linear and rotary actuator, present in the fracture table. The designed robot is theoretically capable of applying a maximum force and torque value of 1284 N and 24 Nm respectively. The end effector of the designed robot can provide a range of 150 mm and 360° motion in the linear and rotary direction respectively. This provides a cylindrical workspace.Table 3-1 gives the specification of the final design. The design is simple and makes use of existing fracture table without making any changes on it. This will encourage the OR staff to use it, without any hesitation. It was possible to achieve an accuracy of 1 mm and 2° in the translational and the rotational directions, respectively, using the position control. The implementation of the position controller is provided in Appendix B-7. A custom built F/T is attached to the foot fixator device. The F/T sensor allows accurate measurement of the force and torque, which are being applied to the patient's leg and is used to limit excessive force or torque. Safety features such as the current limiters, the watchdog timer, and the emergency stop button are also provided.

Table 3-1: Specification of the designed HFR.

Parameters	Linear Motion	Rotary Motion
Peak Force/Torque	1284 N	24 Nm
Range of Motion	150 mm	±360°
Speed of Motion	28.67 mm/s	108°/s
Accuracy	1 mm	1°

Part II

Low-level Control Implementation and Validation

Chapter 4

Identification of 2-DoF Robot Setup

This chapter discusses the modeling and identification of the 2-Degree(s) of Freedom (DoF) Hip Fracture Reduction Robot (HFR). The first section describes the setup of the system, followed by measurements of Direct Current (DC) motor parameters that are required for the modeling. Section 4-6 describes the physical modeling of the setup. However, the friction parameter is not available for the modeling. Consequently, subsequent sections explicate the conducted experiment to determine the friction parameter. Finally, in Section 4-6-4 the obtained continuous state space model of the HFR is converted to discrete time by choosing an appropriate sample time.

There are two basic approaches to modeling: 'Black box modeling' and 'First principle modeling'. Both the approaches have their advantages and disadvantages. In black box modeling, the same algorithm can be used to identify different types of systems. Black box modeling is used for systems which are complex in nature due to its complicated mathematical equations or when there is no prior knowledge of the system model. Models such as these are not easy to identify using the first principle. However, since the black box modeling is done by analyzing the input and the output data of the plant, the model is valid only for the signal from which it was obtained. For example, if a step signal was used to identify the plant, this plant should not be utilized for the analysis of high-frequency signals. Plants obtained by using the first principle are more robust as the whole range of input signals can be used to analyze it. However, the drawback of the first principle is that there are a lot of unknown constants and relations in the model description. These constants need to be obtained by performing real-time experiments on the plant. For the modeling of HFR, first principle modeling method has been used. This chapter includes the various steps involved in system parameter identification for every element of the system. Some of the system parameters were obtained from the datasheets while the others were determined experimentally.

4-1 Setup

As described in Chapter 3 the setup is a 2-DoF robot driven by DC motors using belt transmission system. Figure 4-1 shows the complete setup. The linear motion is achieved by

driving a lead screw setup by the DC motor. The rotary motion is achieved by rotating the whole lead screw setup freely around an axis that is parallel to the linear motion. The two motions are completely decoupled and can be modeled and controlled separately. Since, DC

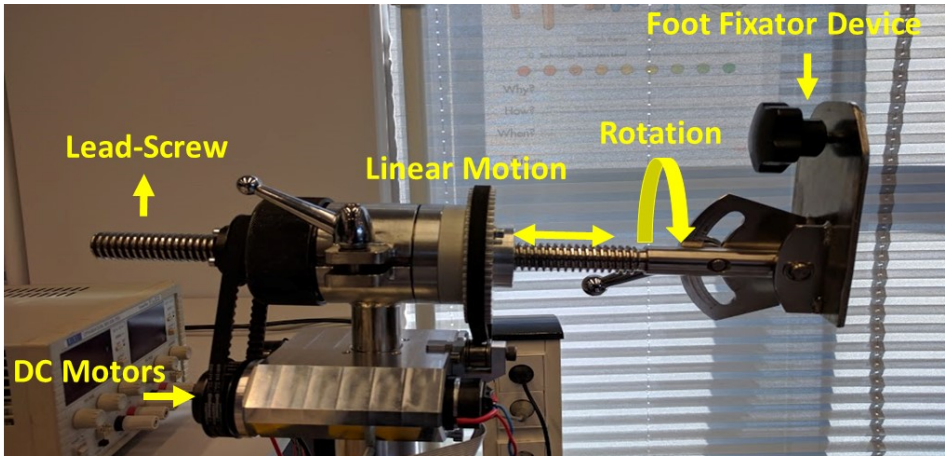


Figure 4-1: 2-DoF HFR

motors are used, the control input is voltage. This input is commanded from the computer and is controlled by providing the Pulse Width Modulation (PWM) signal between -1 and +1 (corresponds to the maximal -24V to +24V moving the motor clock and anti-clockwise with maximum speed). There are two measured outputs: θ-the angle of rotation of the motor and the Force/Torque (F/T) measurements corresponding to the linear and rotary motion. Physical parameters of the system have to be determined to model a system. The following sections discuss some of these parameters.

4-2 Inertia

4-2-1 Linear Motion Inertia

Inertia in the linear motion is contributed by the DC motor, gearbox, pulleys and knurled nut (thumb nut). All of these elements are cylindrical in shape. Their inertia can be calculated by using the formula

$$J = \frac{1}{2}mr^2 \tag{4-1}$$

where m is the mass and r is the radius of the cylinder. The inertia of the motor and gearbox needs to be translated to the shaft side by multiplying with the square of the gear ratio n. The effective inertia of the system can then be calculated by adding inertia contribution from the individual elements. Measurements were made to calculate the inertia of each of these elements. The measured values and the inertia have been tabulated in Table 4-1.

The inertia of the motor and gear box was obtained from the datasheets of the manufacturer.

Hence the overall inertia of robot for linear motion at the gear output shaft is given by

$$\begin{aligned} J_{eq} &= 2 \times J_p + J_{kb} + n^2(J_m + J_{gb}) \\ &= 3.2 \times 10^{-3}\ \mathrm{kg\,m^2} \end{aligned} \tag{4-2}$$

Table 4-1: Inertia in linear motion

Parameter	Part Name	Mass (kg)	Radius (m)	Inertia ($\mathrm{kg\,m^2}$)
J_p	Pulley	0.2	0.023	5.7×10^{-5}
J_{kb}	Knurled nut	0.95	0.035	5.82×10^{-4}

Table 4-2: Inertia in the DC motor used for linear motion

Parameter	Physical Property	Supplied Value ($\mathrm{kg\,m^2}$)
J_m	Rotor inertia	7.92×10^{-6}
J_{gb}	Gear box inertia	8×10^{-8}

4-2-2 Rotary Motion Inertia

For the rotary motion, the inertia is contributed by the motor, gearbox, pulleys, clamping shell, knurled nut, F/T sensor, sensor mounting plate and the lead-screw setup. Inertia for these elements is calculated similarly to as done in the Section 4-2-1. The inertia of motor, gearbox, pulleys, clamping shell, knurled bolt, F/T sensor, and the lead screw setup is obtained by performing measurements. These measurement values and the corresponding calculated inertia have been tabulated in Table 4-3.

Table 4-3: Inertia in rotary motion

Parameter	Part Name	Mass (kg)	Radius (m)	Inertia ($\mathrm{kg\,m^2}$)
J_{sp}	Small Pulley	0.2	0.023	5.7×10^{-5}
J_{bp}	Big Pulley	0.28	0.047	3.2×10^{-4}
J_{kb}	Knurled Bolt	0.95	0.035	5.82×10^{-4}
J_{rb}	Clamping block	0.725	0.03	3.3×10^{-4}
J_s	F/T	0.22	0.03	9.9×10^{-5}
J_{ls}	Lead screw setup	1.5	0.01	7.5×10^{-5}

However, inertia for sensor mounting plate is calculated using parallel axis theorem with the following formula.

$$J = mr^2 + \frac{1}{12}m(w^2 + h^2) \tag{4-3}$$

where w is the width of the plate, h is the height of the plate and r is the distance from the axis of rotation to the center of mass of the plate. Hence the inertia of sensor mounting plate with $0.564\,\mathrm{kg}$, $0.05\,\mathrm{m}$, $0.08\,\mathrm{m}$, and $0.02\,\mathrm{m}$ as mass, width, height and distance from axis of rotation to the center of mass respectively is

$$\begin{aligned} J_{smp} &= 0.564 \times 0.02^2 + \frac{1}{12}0.564(0.05^2 + 0.08^2) \\ J_{smp} &= 6.44 \times 10^{-4}\ \mathrm{kg\,m^2} \end{aligned} \tag{4-4}$$

Table 4-4: Inertia in DC motor use for rotary motion

Parameter	Physical Property	Supplied Value ($\mathrm{kg\,m^2}$)
J_m	Rotor inertia	3.35×10^{-6}
J_{gb}	Gear box inertia	7×10^{-8}

The inertia of the motor and the gearbox needs to be translated to the gear output shaft by multiplying with the square of the gear ratio. The inertia of the motor and gear box was obtained from the data sheet of the manufacturer.

There is also a pulley reduction of 4:1 on the rot ary motion. Hence the overall inertia of at the gear output shaft is given by

$$\begin{aligned} J_{eq} &= \frac{J_{bp} + J_{rb} + J_{kb} + J_{ls} + J_{sp} + J_s + J_{smp}}{4} + n^2(J_m + J_{gb}) \\ &= 5.5 \times 10^{-4} \end{aligned} \tag{4-5}$$

4-3 Brushed DC Motor

DC motors parameters are required to obtain the dynamical model of the HFR. These parameters are supplied by the manufacturer and can be obtained from the datasheet. The parameters for the motors used for the linear and rotary motion are tabulated in Table 4-5. The brush frictions are identified in Section 4-6-2

Table 4-5: DC motor parameters

Parameter	Physical Property	Supplied Value for motor used for linear motion	Supplied Value for motor used for rotary motion
K_t	Torque constant	$29.2 \times 10^{-3}\,\mathrm{Nm/A}$	$25.9 \times 10^{-3}\,\mathrm{Nm/A}$
K_e	Electrical constant	$29.2 \times 10^{-3}\,\mathrm{V/rads^{-1}}$	$25.9 \times 10^{-3}\,\mathrm{Nm/A}$
R	Motor terminal resistance	$0.583\,\Omega$	$0.611\,\Omega$
L	Motor terminal inductance	$0.191 \times 10^{-3}\,\mathrm{H}$	$0.119 \times 10^{-3}\,\mathrm{H}$

4-4 Backlash

Datasheet of the gear box is used to obtain the amount of backlash in the motor. The backlash found from the datasheet of the DC motors for the linear as well as the rotary motion is 0.8°

and 1° respectively. Since, lead screw setup with a 4 mm pitch, is used to provide the linear motion, a 0.8° causes an error of 0.009mm. The error obtained is minuscule and can be neglected. For the rotational motion, belt transmission system is used, with a pulley ratio of 1 : 4. Hence, the backlash error of 1° gets reduced to 0.25°. However, the rotation motion has also backlash error of 5° in foot fixator device; this backlash error was measured using an inclinometer. Since during fracture reduction procedure, the robot only moves in a particular direction, the effect of backlash is avoided.

4-5 Friction

Understanding friction is critical for the control engineer as friction is highly non-linear and may result in steady state errors, limit cycles, and reduced performance. Control engineer needs to understand the effect of friction on closed loop response and design control laws that can take care of these errors caused by friction. The amount of friction on the DC motor due to motor brushes contributes considerably high friction to the end effector side. The effect of friction gets amplified due to the presence of gear reduction. If the gear ratio is n then the friction on the end effector will be n^2 times the motor friction. Gearbox also has sliding friction which cannot be neglected for higher loads. The friction due to DC motor brushes can be assumed to be *dry friction*. The *dry friction* consists of static, kinetic and stribeck friction. In the literature, there are several static as well as dynamic friction models available [23]. For a DC motor, Coulomb friction does not provide good model at high voltages and modeling the friction as stribeck friction gives better results [34]. However, for modeling of HFR, a viscous friction model is used because of its linearity.

4-6 Physical Modeling

The robot can be modeled by rotating masses connected to a DC motor as in Figure 4-2. From the motor's perspective, it looks as if it moves the whole weight of the robot as if it was spread around the circumference of the gear shaft. The angular position and velocity on the gearbox side are denoted by θ and ω. The process input is the voltage v, supplied by the motor driver and the measured process output is the angular position. Finally, linear friction is represented by the damping constant b.

The torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. Since brushed DC motors are used, the magnetic field can be assumed to be constant. The motor torque T is then proportional to the armature current i as shown in the Eq. 4-6.

$$T = K_t i \tag{4-6}$$

where K_t is called the torque constant of the motor.

The electrical model of DC motor can be expressed by the following equations:

$$L\frac{di}{dt} + Ri + K_e\omega = v \tag{4-7}$$

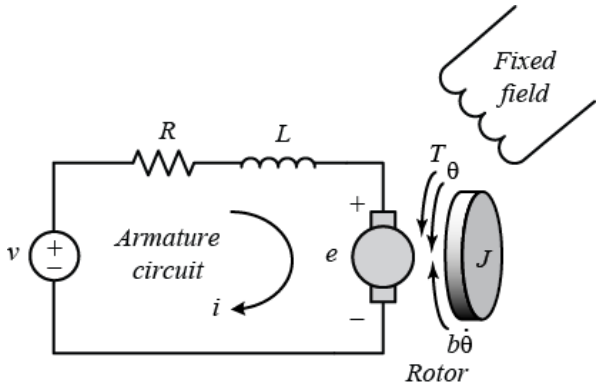


Figure 4-2: Schematic of a DC motor. Source Control Tutorials [22]

where L is the inductance, R is the resistance, v is the input voltage, $\omega = \dot{\theta}$ is angular speed and K_e is back emf or electrical constant of the motor.
The mechanical equation of DC motor can be written as:

$$J_{eq}\frac{d\omega}{dt} - b\omega = T \tag{4-8}$$

where J_{eq} is the equivalent load inertia, b is the damping constant of the modeled linear friction and T is the input torque. Substituting equation 4-6 in 4-8 we get

$$J_{eq}\frac{d\omega}{dt} - b\omega = K_t i \tag{4-9}$$

Rearranging equation 4-7 and 4-9 and introducing state variables

$$\begin{aligned} x_1 &= i \\ x_2 &= \theta \\ x_3 &= \omega \end{aligned} \tag{4-10}$$

and the input $u(t) = v$, the system can be written in state-space form as

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -R/L & 0 & -K_e/L \\ 0 & 0 & 1 \\ K_t/J_e & 0 & -b/J_e \end{bmatrix} x(t) + \begin{bmatrix} 1/L \\ 0 \\ 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x(t) \end{aligned} \tag{4-11}$$

4-6-1 Simulation Model

The Equation 4-11, derived in the previous section will be used to model the system. For this equation, equivalent inertia and motor's electrical parameters have been already identified in the Section 4-2 and 4-3. However, b (viscous friction) needs to be identified experimentally.

4-6-2 Parameter Estimation

Since the modeling of the system is done using the first principle, the unknown parameters of the model has to be determined by performing real-time experiments on the HFR. The Coulomb, static and stribeck friction were neglected for this model. There will always be some difference between the model and the real setup. The advantage of modeling by the first principle is that the model is valid for all sets of input-output data, not just in the data set that was used for parameter estimation. Due to this, the choice of the input signal for parameter estimation is not crucial. A signal as simple as an impulse signal should be enough in this case. However, for identification a random PWM signal (Figure 4-3) was generated for the identification with the constraints that acceleration and deceleration cannot be more than 0.05 and 0.5 respectively. This limitation was imposed on acceleration and deceleration because the motor driver cannot accelerate or decelerate faster than these values. Due to the presence of friction, a better estimate of the parameters is obtained by using the non-linear least square approach. The model (refer to Eq. 4-11) that was derived earlier is utilized for this estimation. Matlab function *lsqnonlin* was used for the identification.

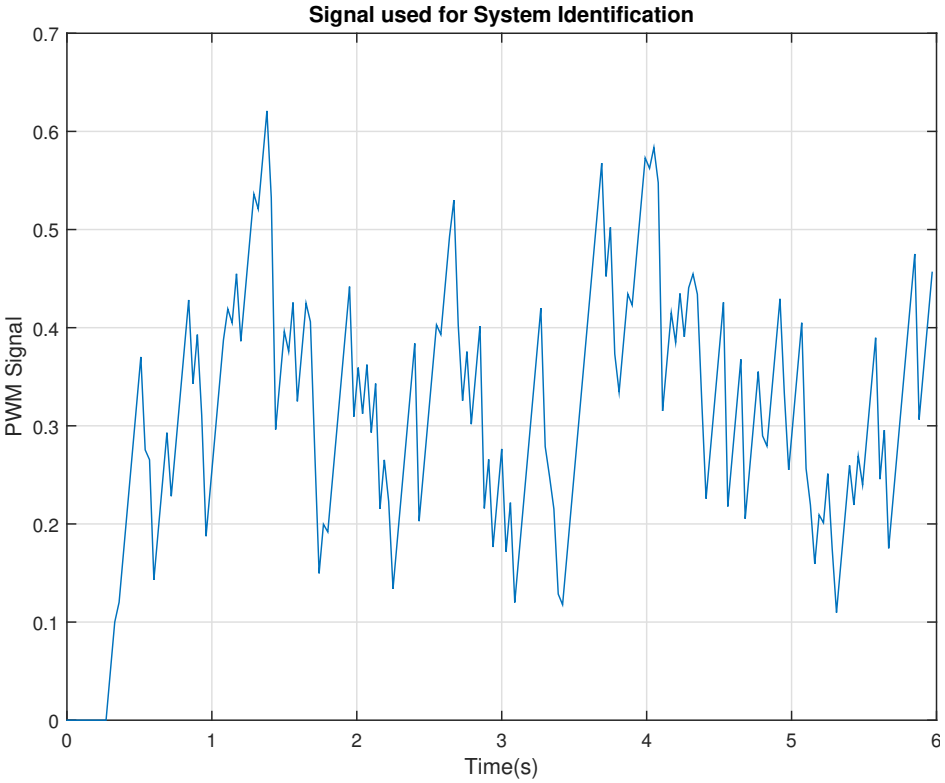


Figure 4-3: Input PWM signal used for identification of linear motor

The values obtained for damping coefficient b was found out to be 0.398 and 8.5026 with Variance accounted for (VAF) value of 99.9% and 97.5% for linear and rotary motor respectively. Figure 4-4 shows the plot of true process output and the modeled data for identification.

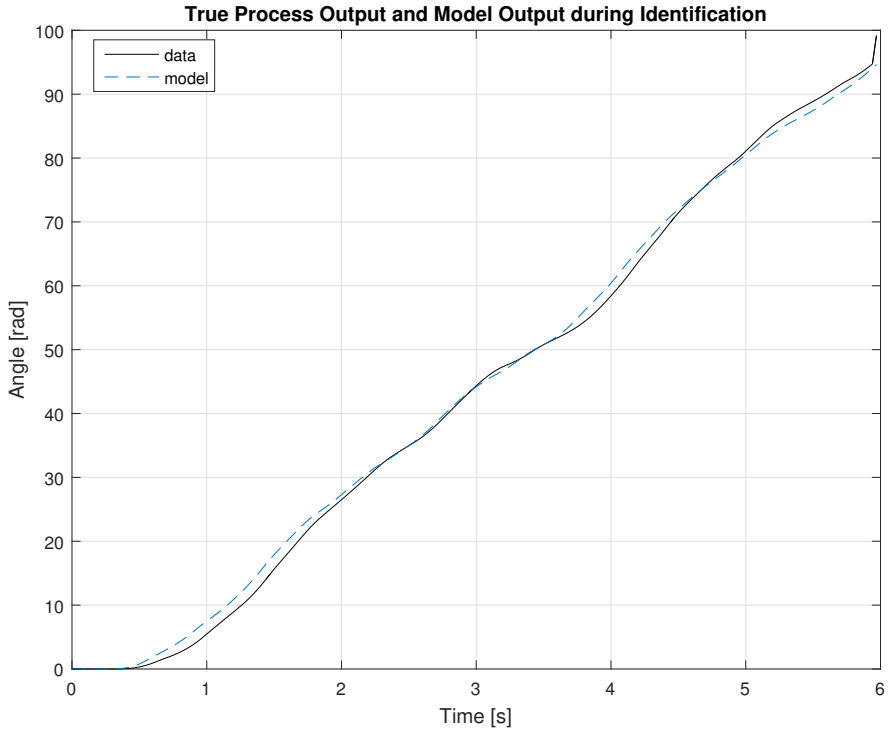


Figure 4-4: Comparison of true process output with a simulation of the estimated state-space model during identification.

The estimation resulted in the following state-space model for the linear motion

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -3052 & 0 & -152.5 \\ 0 & 0 & 1 \\ 162.3 & 0 & -123 \end{bmatrix} x(t) + \begin{bmatrix} 5236 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \\ y &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x(t) \end{aligned} \tag{4-12}$$

and for the rotary motion, the resulted state space model is:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -5134 & 0 & -217.5 \\ 0 & 0 & 1 \\ 5221 & 0 & -139.1 \end{bmatrix} x(t) + \begin{bmatrix} 8403 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \\ y &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x(t) \end{aligned} \tag{4-13}$$

The Matlab code for the estimation is given in Appendix B.2.

4-6-3 Model Validation

The validation of the identified state-space model of the HFR was done by simulating the obtained model and then comparing it against the output of the actual setup. A different set

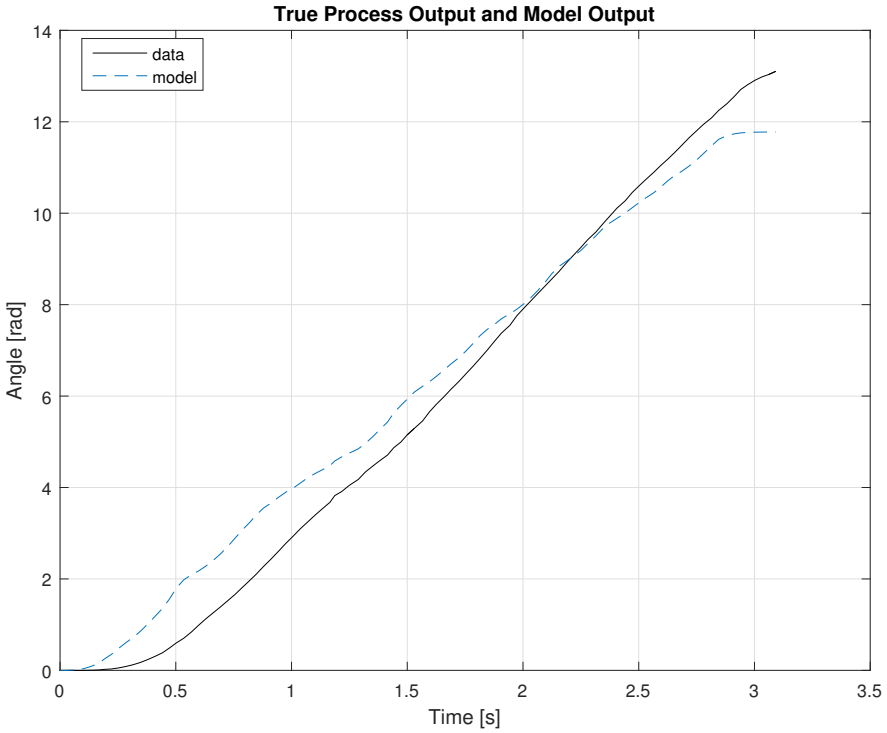
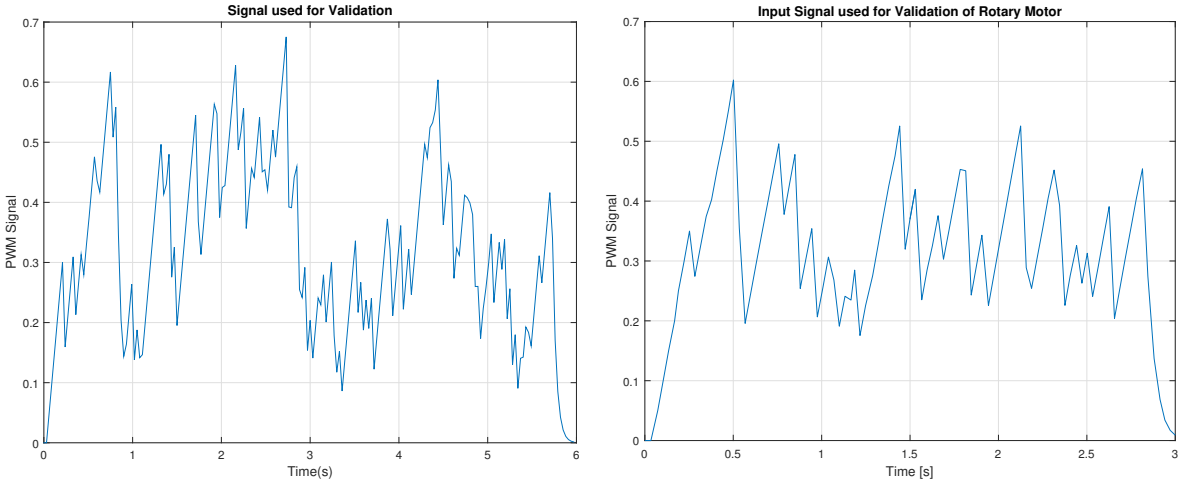


Figure 4-5: Comparison of true process output with a simulation of the estimated state-space model during identification of rotary motor.

of input data (Figure 4-6) was used and the output of the state-space model was compared with the calculated friction parameter against the actual setup output. The VAF value for the data validation was found to be 99.6% and 98.8% for linear and rotary motion respectively. Figure 4-7 and 4-8 shows the comparison of the true process output and a simulation of the estimated state-space model.

4-6-4 Converting the Continuous Time Model to Discrete Time

Continuous time signal can be converted to discrete time signal by selecting appropriate sampling frequency. The selection of the sampling frequency is important to avoid the aliasing problems. For discrete-time systems, the Shannons sampling theorem, based on the Nyquist frequency, is used to avoid aliasing. A continuous time signal that does not contain frequency components greater than crossover frequency ω_c is uniquely determined if the sampling frequency is higher than $2\omega_c$. One way to measure the sampling time is the closed loop bandwidth. A good rule of thumb is to use sampling frequency to be in between 10 to 30 times the bandwidth, which can be determined by closed loop bode plot. The time constant of a system determines the bandwidth of the system. Hence the time constant of the system can be used to select the rate at which the system must be sampled, to adequately represent in the digital form. For DC motors the system can be decomposed into component first order subsystems, by decomposing it into its mechanical and electrical part. The time constant for each of this system is given by:



(a) Input PWM signal used for Validation of linear motor.

(b) Input PWM signal used for Validation of rotary motor.

Figure 4-6: Input PWM signal used for identification of 2-DoF robot

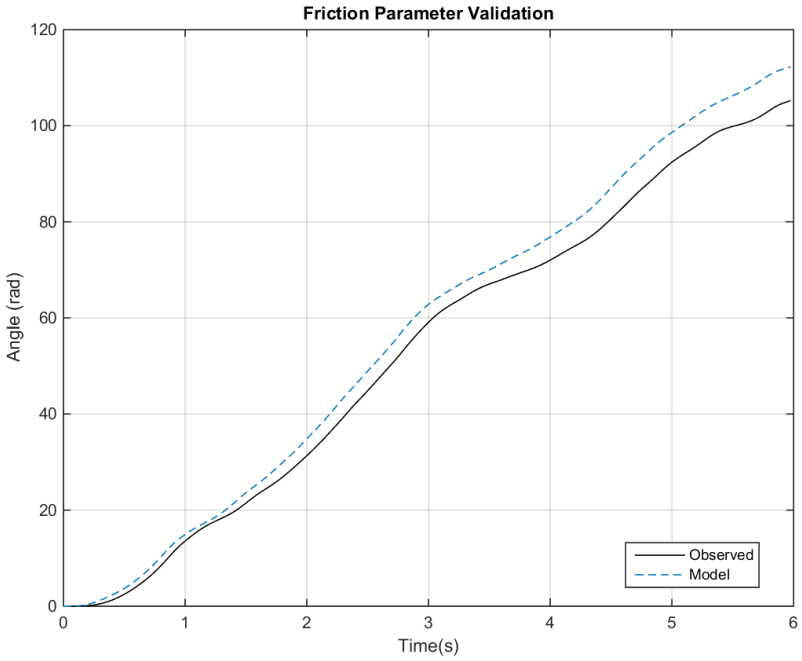


Figure 4-7: Comparison of true process output with a simulation of the estimated state-space model during validation of Linear motor.

$$\begin{aligned} \tau_e &= L/R \\ \tau_m &= J/B \end{aligned} \tag{4-14}$$

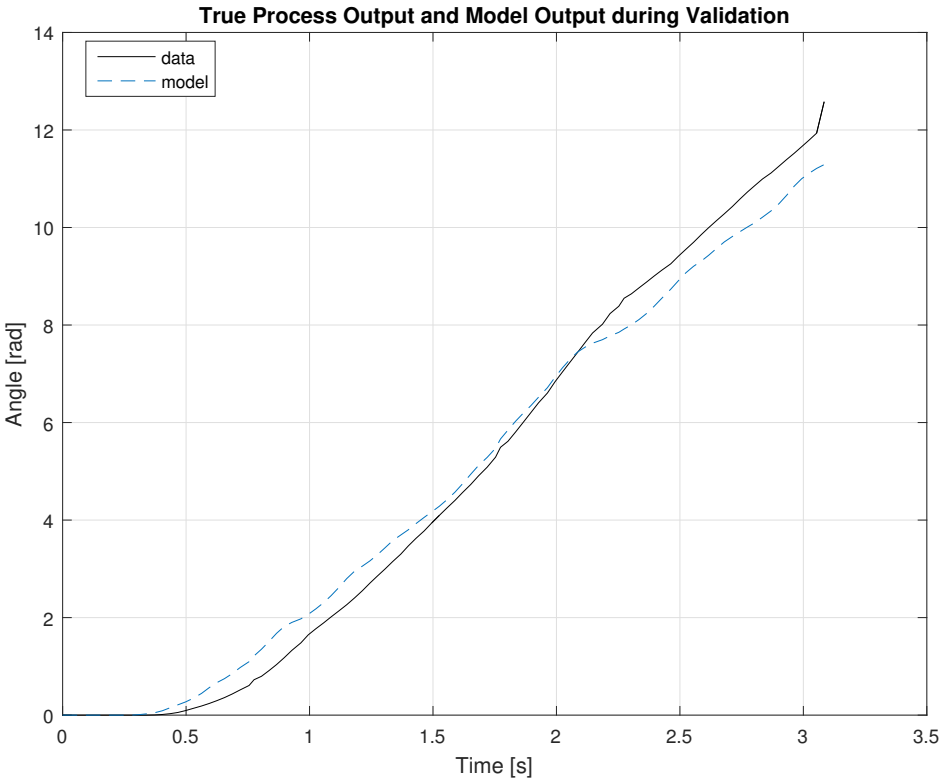


Figure 4-8: Comparison of true process output with a simulation of the estimated state-space model during validation of rotary motor.

where L is the inductance, R is the resistance; J_e is the inertia and b is the damping friction. For the DC motor used for linear motion, the electrical and mechanical time constants are 0.0003 and 0.2 respectively. For rotary motion, the electrical and mechanical time constants are 0.0002 and 0.44 respectively. It can be seen that mechanical subsystem is much slower than the electrical system. On the shaft speed time scales, current appear to change instantaneously, and the response of the motor is dominated by the mechanical subsystem. We can thus choose sampling time based on the time constant of the mechanical subsystem. Ideally, the sampling time of half the time constant is sufficient to avoid aliasing. However, in practice, the sampling time is chosen to be about ten times smaller than the time constant of the system. For the linear and rotary motion, the sampling time of 0.02 and 0.03 was selected respectively. Matlab command *c2d* with *tustin* method was used to convert the state matrix in continuous time to discrete time. The discrete time state space model thus obtained for the linear motion is

$$\begin{aligned} \dot{x}[n+1] &= \begin{bmatrix} -0.939 & 0 & -0.042 \\ 0 & 0 & 0.009 \\ 0.045 & 0 & -0.134 \end{bmatrix} x[n] + \begin{bmatrix} 3.209 \\ 0.023 \\ 2.335 \end{bmatrix} u[n] \\ y[n] &= \begin{bmatrix} 0 & 1 & 0.004 \end{bmatrix} x[n] + \begin{bmatrix} 0.012 \end{bmatrix} u[n] \end{aligned} \tag{4-15}$$

discrete time state space model the rotary motion is

$$\begin{aligned} \dot{x}[n+1] &= \begin{bmatrix} -0.988 & 0 & -0.013 \\ 0.005 & 0 & 0.005 \\ 0.316 & 0 & -0.686 \end{bmatrix} x[n] + \begin{bmatrix} 1.568 \\ 0.597 \\ 39.78 \end{bmatrix} u[n] \\ y[n] &= \begin{bmatrix} 0.002 & 1 & 0.002 \end{bmatrix} x[n] + \begin{bmatrix} 0.298 \end{bmatrix} u[n] \end{aligned} \tag{4-16}$$

These obtained discrete time state-space models will be used while doing the controller designs.

4-7 Discussion

To obtain the model of the HFR, system parameters were needed to be determined. Most of the parameters such as motor constants have been taken from the datasheets of the motor. Measurements were made to calculate the inertia of the setup. However, after calculating the overall inertia, it was realized that the inertia contribution from the gearbox and motor are much higher compared to the inertia of the setup e.g. inertia of gear box and the motor was $2.6 \times 10^{-4}\,\mathrm{kgm}^2$ while that of rest of the setup was $0.6 \times 10^{-4}\,\mathrm{kgm}^2$. In hindsight, the inertia of the structure could have been neglected. However, it was kept to maintain the calculation as accurate as possible. The 5° backlash is quite significant and in many cases cannot be ignored. For the purpose of fracture reduction, however, the direction of motion remains constant, and hence it is possible to avoid the backlash effect. The obtained viscous damper values were 0.398 and 8.5026 for linear and rotary motion. The value of damper for the rotary motion is almost 28 times more than the damper value of linear motion. Such a high damper value for rotation motion was expected because of two reasons. First of all due to high gear reduction in rotary motion, the efficiency of the gearbox is less compared to its linear counterpart. Secondly, there is high friction present in the rotary motion setup itself. For the validation of the model, a different set of input data was generated based on random number generation using Matlab command *rand*. The obtained VAF during validation was found out to be 99.6% and 98.8% for linear and rotary motion respectively. The VAF values obtained were expected, as the plots indicate good correspondence between the simulated output and the actual setup output.

4-8 Conclusion

This chapter presents the modeling of the HFR. Since the 2-DoF are completely decoupled, modeling was relatively straight forward. The state-space model for each degree is obtained by making use of the first principle, and then by parameter estimation. Most of the parameters of the setup were obtained from the data sheet of the manufacturer, while the rest is produced by measurements. Since friction cannot be measured, we modeled it as a viscous damper and obtained the value experimentally. In both the degrees, only friction parameters was needed to be modeled making it a one-dimensional optimization problem. Matlab function *lsqnonlin* was used for the estimation. It was found that the correspondence between the true process output and simulation output is excellent, as the VAF values obtained for both the degrees were above 98%. Finally, the obtained continuous time state-space models were converted to

discrete time by selecting appropriate sampling time. Conclusion is that the derived model can be used in testing various compliant control schemes.

Chapter 5

Controller Design

5-1 Introduction

Robots used in medical procedures have various level of autonomy. Even advanced surgical robots such as Da Vinci and Zeus use telemanipulation based force feedback control and are not completely autonomous. Robotic systems which can perform completely autonomous procedures, such as, hip replacement robot ROBODOC are called autonomous robots. The idea is that the Hip Fracture Reduction Robot (HFR) will eventually, autonomously carry out the reduction, by applying the traction force and rotation torque in a way similar to the procedure currently executed by the surgeon. In the open literature, both position and force control have been used to control the surgical robots. Two types of control schemes that are used in surgical robotics are as follows:

- Control of the robot position along the direction of task space, the environment imposes natural force constraints.
- Control of the robot force along the direction of task space, the environment imposes natural position constraints.

Unfortunately, the requirement of control for the automation of fracture reduction procedure is not understood. Efforts have been made by Maeda et al. (2005) [5] and Gösling et al. (2006) [16] to measure the force during the fracture reduction. However, external factor such as muscle activation force and additional force measured due to the excess force applied by the surgeon during the surgery influences the measurements. Modeling and simulation of muscular force, provide an alternative solution to this problem. Even the results obtained from theoretical modeling cannot be used directly. Customization on the theoretical Force/Torque (F/T) needs to be carried out for an individual patient. Since the hip fracture reduction procedure involves *pull* and *rotate* movement of the patient's leg, there will be direct interaction between the patient's leg and the robot, in the case of automation of

this procedure. Therefore, there is a fundamental requirement to control the robot force in as well as position in a compliant manner. Hence to understand the control needs of the HFR various compliant controller needs to be evaluated and compared. This chapter gives a detailed overview and simulation results of different types of compliant control schemes that can be used for the control of HFR. These simulation results will then be compared against the measurement data obtained during hip fracture surgery.

5-2 Simulation

Simulations were made by using the identified state-space model of HFR, and by modeling the patient's leg as a double spring and a damper system (see Figure 5-1). The model for the patient's lower limb was obtained from Ravishankar [32]. This model of the leg was achieved by performing measurement during hip fracture reduction procedure. The measurements values of a female patient having an age of 94, height of 165 cm and weight of 43 kg were used. The reported Kse, Kpe and b values used are 3670 N/m, 3920 N/m and 3920 Ns/m respectively. Similarly, for the rotation of the leg, patient's leg was modeled as a double torsional spring, and damper with Kse, Kpe and b values as 24.09 Nm/rad, 15.28 Nm/rad and 15.28 Nms/rad respectively. During the rotation of the leg, surgeon adopts a technique of trial and error as there is no clarity on the direction of rotation which would reduce the fracture. Hence the rotation data contain lots of unnecessary rotational movements. However, for HFR it is expected that the robot will get the information about the torque and angle of rotation from the higher level control and will only rotate the leg in one direction as described by the higher level control. Due to this reason, only a part of the measurement during rotation was used. The performance of each controller is then compared against the measurement made by Ravishankar [32].

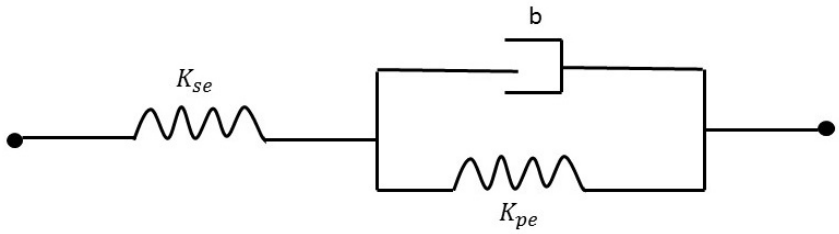


Figure 5-1: Model of patient's leg as a double spring and damper system. Source: Ravishankar [32]

5-3 Stiffness (Compliance) Control

Compliance control or stiffness control can be either passive or active. In passive compliance control, the robot's end effector is equipped with physical spring and damper to guarantee a particular compliant behavior. On the other hand, active compliance control is achieved by tuning the stiffness of the controller for the stiffness of the environment. This tuning is obtained by applying Position-plus-Derivative (PD) control on the positional error followed

by adjusting the controller gain. Smaller the gain, the more compliant the robot will be for the environment. In this type of control, the contact force is treated as a load disturbance, and no force measurements are required.

5-4 Impedance Control

Impedance (Z) is a measure of opposition to time-varying electric current in an electric circuit. Mechanical impedance is a measure of opposition to the motion of a structure subjected to force. In the frequency domain, mechanical impedance is represented by

$$\frac{F(s)}{\dot{X}(s)} = Z_m(s) \tag{5-1}$$

In terms of position $X(s)$, we may write

$$\frac{F(s)}{X(s)} = sZ_m(s) \tag{5-2}$$

In impedance control, the control law should impose a force to move the object; this force is generated by a virtual mass, spring, and damper as seen in Figure 5-2. The purpose of impedance control is to obtain a dynamic relation between position and force, rather than to control either of these variables alone.

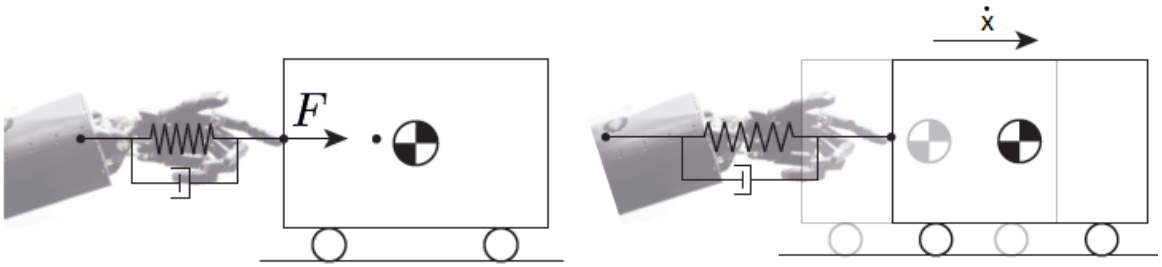


Figure 5-2: Impedence control as Spring and damper system

The simplest form of impedance is the static constant which is the ratio between force and displacement.

$$F = K(x_r - x) \tag{5-3}$$

where F is the force at the robot's end effector, x is the position, and x_r is the reference position. K is a constant which represents stiffness of the system.

Another important term in the impedance relation is the relationship between force and velocity. A desired relation can be written as

$$F = D.\frac{d}{dt}(x_r - x) \tag{5-4}$$

with the constant D representing damping.
The third term that is used to get the required impedance is the acceleration. It is interpreted as inertia.

$$F = M.\frac{d^2}{dt^2}(x_r - x) \tag{5-5}$$

The impedance relation is obtained by combining Equation 5-3, Equation 5-4 and Equation 5-5 i.e.

$$F = K(x_r - x) + D.\frac{d}{dt}(x_r - x) + M.\frac{d^2}{dt^2}(x_r - x) \tag{5-6}$$

Hence we can use an output variable obtained from Equation 5-6 defined as

$$y_i = K(x_r - x) + D.\frac{d}{dt}(x_r - x) + M.\frac{d^2}{dt^2}(x_r - x) - F \tag{5-7}$$

Control which keeps $y_i = 0$ will accomplish the desired impedance behavior of

$$sZ_m(s) = Ms^2 + Ds + K \tag{5-8}$$

where the constant M, D and K represent the desired Inertia, damping, and stiffness values, respectively. Equation 5-7 is used to implement the impedance controller. The error output of this structure can be fed to an either a proportional (P) or a Position-plus-Integral (PI) controller. In this type of control, there is a trade-off between trajectory error and force error. If the motion is unconstrained, F will be zero and the robot will move to the reference position.

Impedance control was tested by performing the simulations on the model of the HFR. A PI controller with $K_p = 0.03$ and $K_i = 0.0006$ was used to regulate the impedance variable $y_i$to zero. The patient's leg is modeled as a double spring and damper system [32]. When the robot starts from the initial state, the external forces are zero and begins to increase as the HFR pulls the patient's leg. At steady-state, the velocity is zero and the force applied by the HFR depends on only the stiffness K.

$$F_{ss} = K(x_r - x_{ss}) \tag{5-9}$$

where F_{ss} and x_{ss} are steady-state force and position of the HFR. Similarly, the force applied by the patient's leg will only depend on the stiffness of the leg. Since, there is no effect of b, the Kse and Kpe will be in series, and their effective stiffness will be:

$$k_{ef} = \frac{K_{se}K_{pe}}{K_{se} + K_{pe}} = 1895 \text{ N/m} \tag{5-10}$$

i.e.

$$F_{ss} = k_{ef}x_{ss} \tag{5-11}$$

Combining equation 5-9 and 5-11 , we get

$$\begin{aligned} F_{ss} &= K(x_r - x_{ss}) = k_{ef}x_{ss} \\ x_{ss} &= \frac{Kx_r}{K + k_{ef}} \end{aligned} \tag{5-12}$$

we observer that if $K >> k_{ef}$ then $x_{ss} \approx x_r$ and if $k_{ef} >> K$ then $x_{ss} \approx 0$.

For the simulation K was selected to be 25000 N/m. The steady state force value applied by the surgeon during the measurement was 169 N and the steady state displacement of the

leg was 80 mm. Using these values in Equation 5-9 the desired position $x_r = 86.76$ mm is obtained. This calculated desired position is used to simulate the impedance controller. The performance of the controller was compared to the force applied by the surgeon during actual surgery. Figure 5-3 shows the comparison of simulated force applied by HFR and by the surgeon on the patient's leg. It can be seen that the performance of the robot matches very well with that of the surgeon. In fact, the peak force applied by the robot is lesser than that applied by the surgeon. This will save the patient from soft tissue trauma.

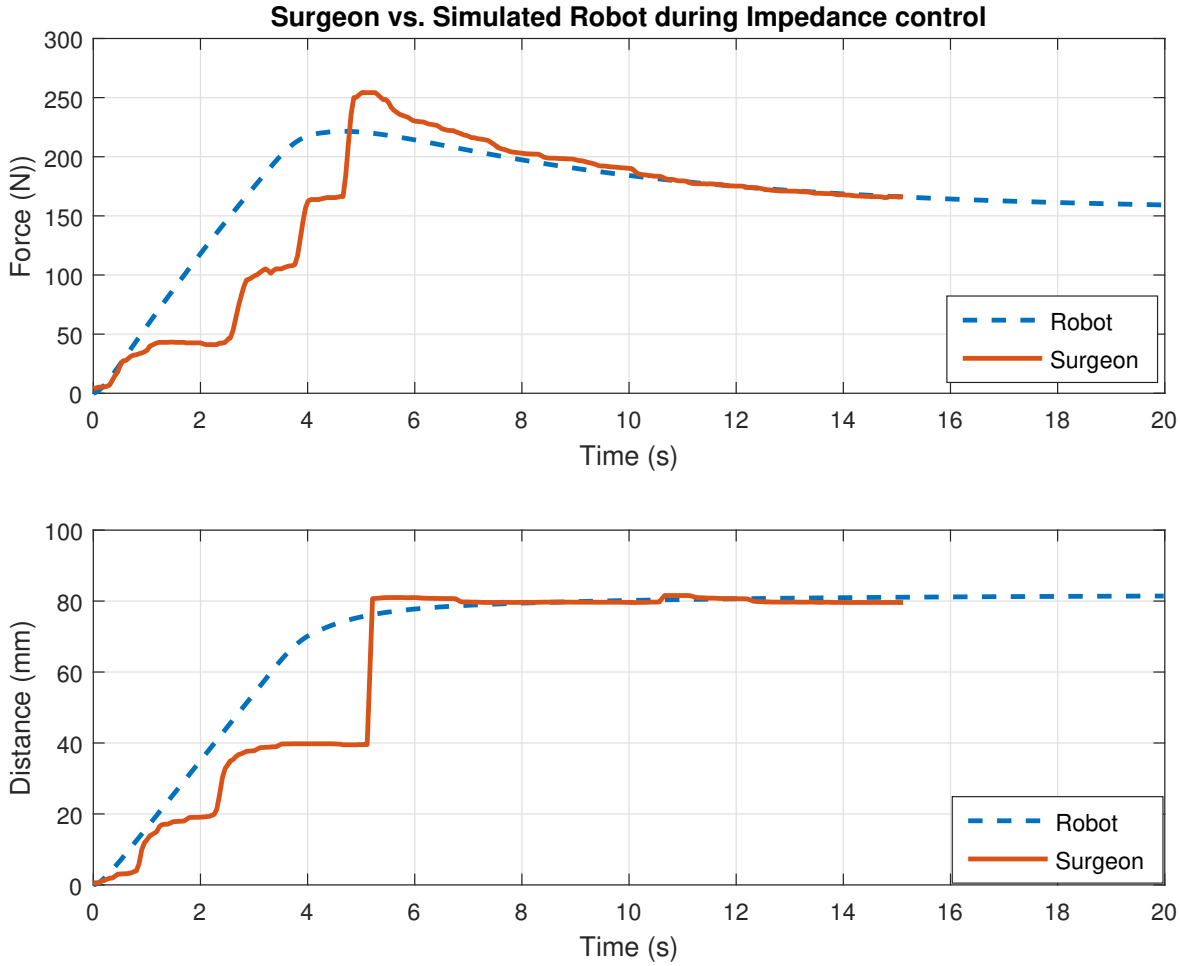


Figure 5-3: Force exerted by surgeon during actual surgery vs. simulation of force exerted by HFR during linear motion, using impedance control

Impedance control was also implemented for the control of the HFR when it rotates the patient's leg. The value of steady state torque and angle applied to the patient's leg by the surgeon was obtained from Ravishankar [32]. The stiffness value of 20 Nm/rad was selected for the simulation, and the damper value of 10 Nms/rad was chosen to avoid any overshoot. The desired angle has been chosen so as to apply a torque of 7.7 Nm. Using similar calculation, as done for the linear case the desired angle was obtained to be 1.15 rad. Figure 5-4 shows a comparison of the torque applied by the surgeon and the simulated HFR.

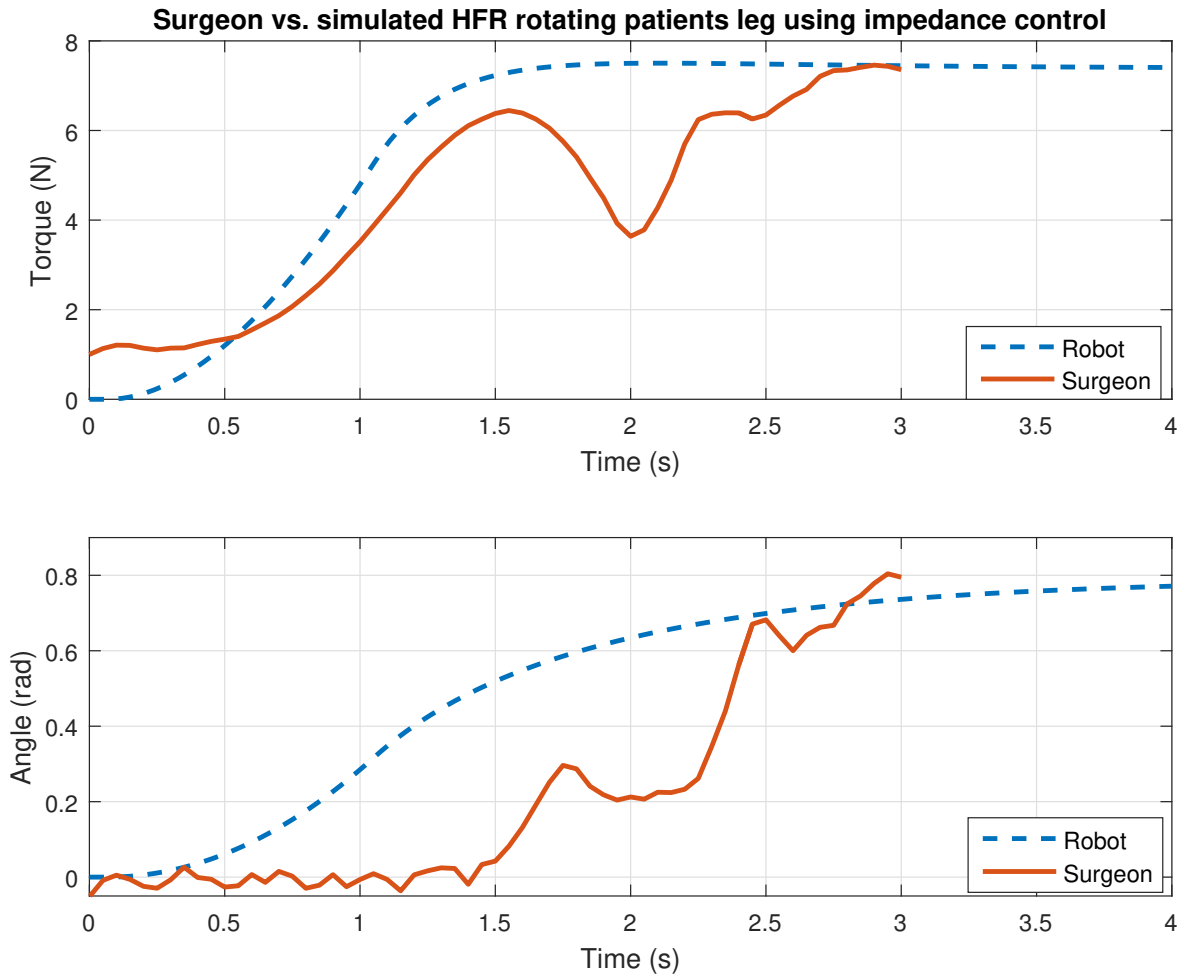


Figure 5-4: Force exerted by surgeon during actual surgery vs. simulation of force exerted by HFR during rotary motion, using impedance control

5-5 Admittance Control

Mechanical admittance is defined as

$$\frac{\dot{X}}{F} = A \tag{5-13}$$

It is the inverse of the impedance definition in Equation. 5-8. The underlying concept of compliant motion control, using admittance control, is to take a position-controlled robot as a baseline system, and to make the necessary modifications of the admittance to this system, to enable the execution of constrained tasks.

In general admittance control measures a force and impose a velocity. Since we want the manipulator to have admittance as described by Equation 5-13, the contact force which acts on the manipulator can be defined as:

$$F_e = K(x_r - x) + D(\dot{x_r} - \dot{x}) + M(\ddot{x_r} - \ddot{x}) \tag{5-14}$$

By transforming Equation 5-14 we get:

$$\ddot{x} = \ddot{x_r} - {M_d}^{-1}[F_e - K(x_r - x) - D(\dot{x_r} - \dot{x})] \tag{5-15}$$

Equation 5-15 is a base to design the admittance controller. The analysis for the admittance control was similar to that in Section 5-4. A PI controller with $K_p = 0.001$ and $K_i = 0.0001$ was used as the motion controller. The stiffness value of controller was selected to be 2400 Nm. Using Equation 5-9 with $F_{ss} = 169\,\text{N}$ and $x_{ss} = 80\,\text{mm}$, x_r was calculated to be 150 mm. For the rotation of the leg, the stiffness value was selected to be 1 Nm/rad. The desired angle was chosen so as to apply a torque of 7.7 Nm which was calculated, based on the steady state torque and angle of rotation and it was calculated to be 8 rad. The value of D was selected so as to get minimum oscillations. However tuning of D turned out to be tough as for small values D caused oscillation and high valued of D created velocity ripples. The optimal value of D for which least oscillation was observed is 100 Ns/m and 0 Nms/rad for linear and rotary motion respectively. The performance of the controller was compared to the force applied by the surgeon during the actual surgery. Figure 5-5 and 5-6 shows the comparison of simulated force and torque applied by HFR and by the surgeon on the patient's leg.

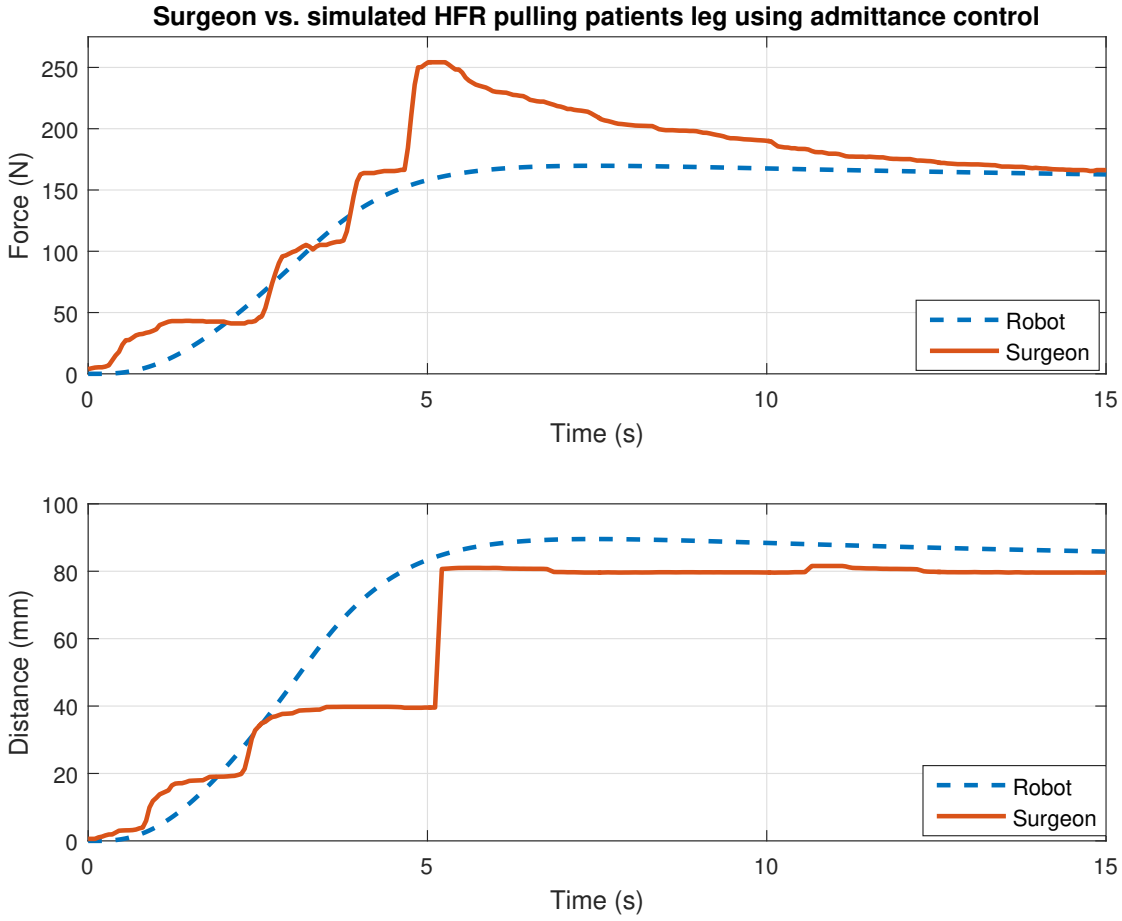


Figure 5-5: Force exerted by surgeon during actual surgery vs. simulation of force exerted by HFR during linear motion, using admittance control

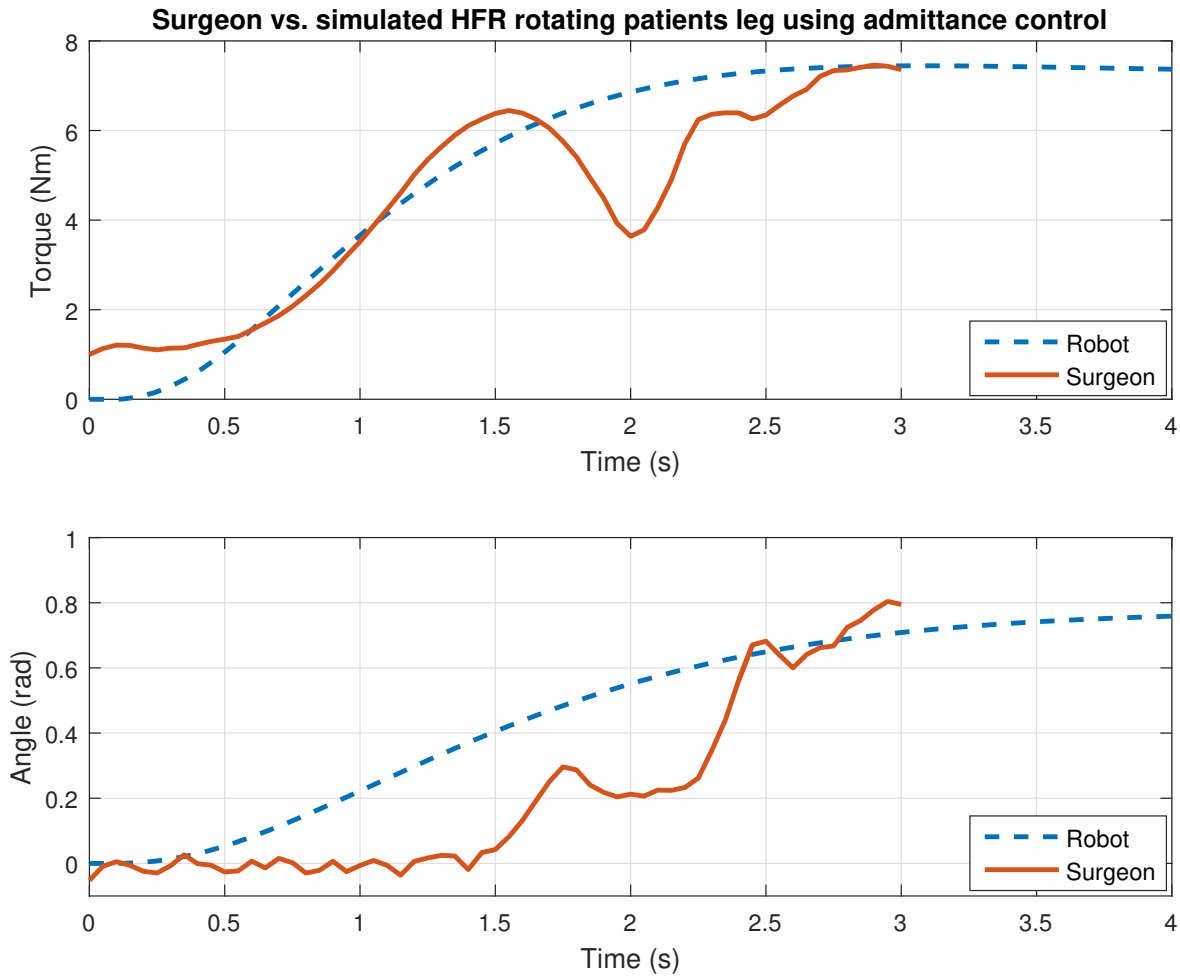


Figure 5-6: Torque applied by surgeon during actual surgery vs. simulation of torque applied by HFR during rotary motion, using admittance control

5-6 Explicit Force Control

Explicit force control is aimed at obtaining a specified force in the contact between the robot and the environment. In theory, this kind of control should be the best way to control both the transient and steady state forces, exerted by the manipulator on the environment. Explicit force control includes two categories [24]. One is force-based, and the other is position-based explicit force control. In the second category, the reference force is transformed to a reference position through an admittance which is the inverse of impedance Z defined in Equation 5-8. Since this control has the same structure as the admittance control, only the first category of force based control will be discussed and implemented.

In the force-based control, the control law is typically chosen as one of the subsets of Proportional-Integral-Derivative (PID). The measured force is directly used as a feedback to get the force error. If the desired force trajectory F_d is known, then a feed-forward control scheme similar to position control can be used, i.e. if we choose control input as

$$u = K_p(F_d - F) + K_d(\dot{F}_d - \dot{F}) + \phi \tag{5-16}$$

where F is the measured force, and ϕ is the feed forward signal. However, this type of force control is not very robust. Since the force measurement in practice is noisy, it is tricky to obtain $\dot{F}$. Hence PI controllers have shown greater stability and are preferred over PD control.

Since PI controllers have shown greater stability, it is used here for the simulation. The PI control for linear motion was selected to be $K_p = 0.02$ and $K_i = 0.0082$, for the rotary motion $K_p = 0.1$ and $K_i = 0.002$ were used. The steady-state value of the force and torque was selected from the measurement [32] as 169 N and 7.7 Nm. Figure 5-7 and 5-8 shows the comparison of force and torque applied by the surgeon and the HFR using the direct force controller.

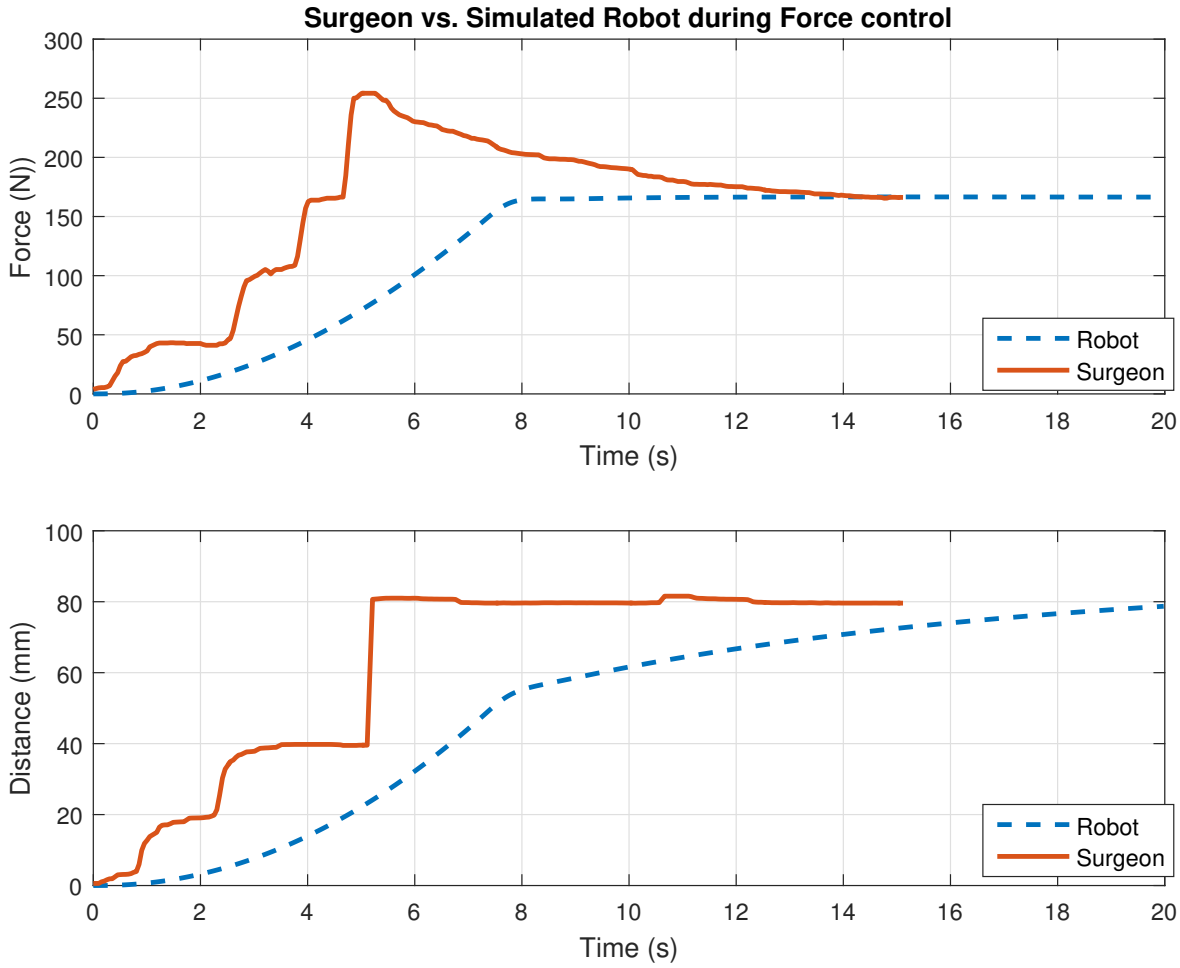


Figure 5-7: Force exerted by surgeon during actual surgery vs. simulation of force exerted by HFR during explicit force control

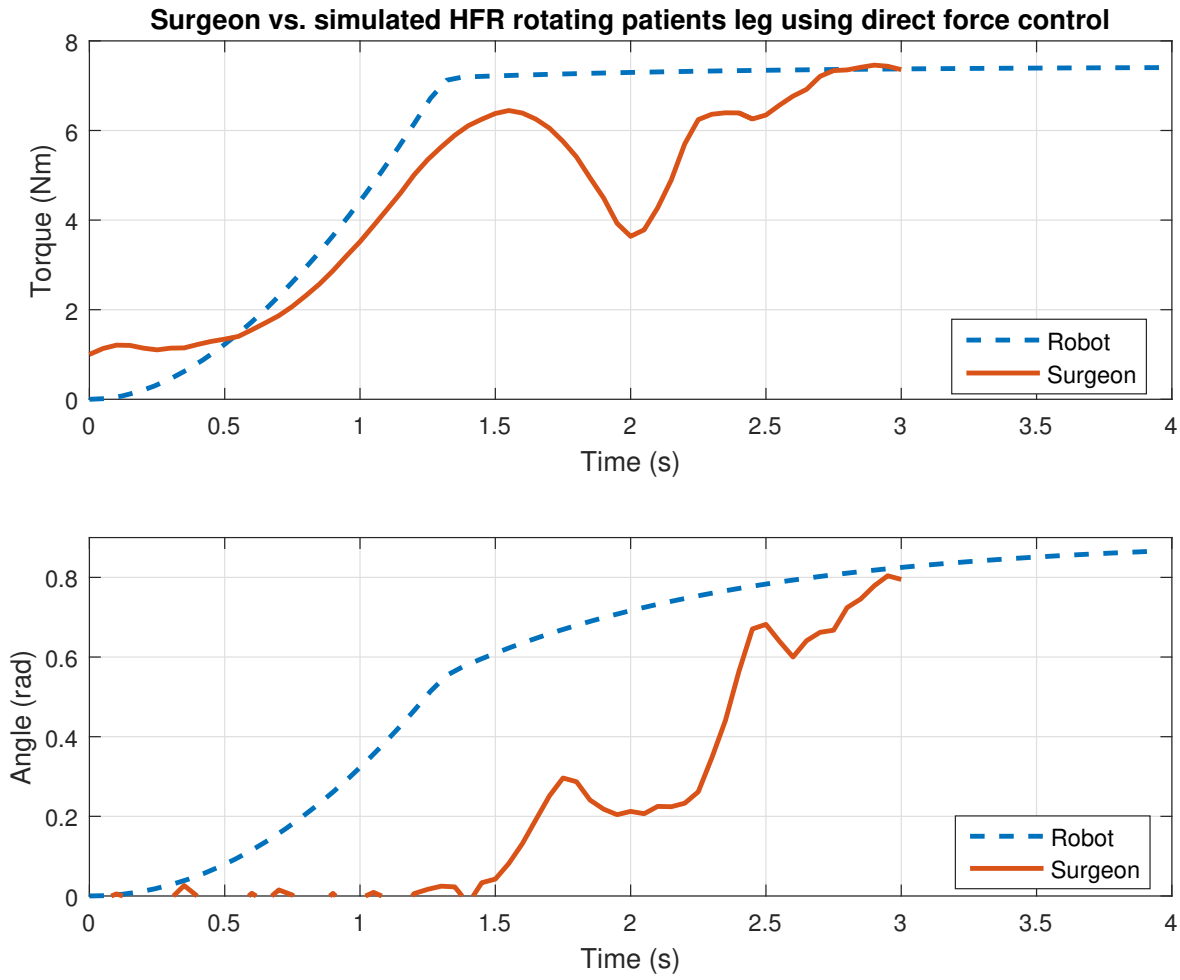


Figure 5-8: Force exerted by surgeon during actual surgery vs. simulation of torque exerted by HFR during explicit force control

5-7 Discussion

In the simulation, it was observed that the HFR was able to apply the desired amount of force and torque, without any overshoots in the case of Admittance and direct force control. This behavior is expected as the direct force control keeps the specified force between HFR and patient's leg. In the case of admittance control, for a known environment it possible to construct an admittance such that the system can track the force trajectory with small or zero error [40]. With no overshoot, patient's can avoid soft tissue trauma. Impedance control, on the other hand, had an overshoot of 33%. However, this overshoot is still well below the force overshoot of 48% measured during the surgery. In the impedance control for the rotation case the damper value used was 0 Nms/rad, this is because in the case of rotation there is already a damper of 8.5 Nms/rad present due to friction, as identified in Chapter 4. Since the stiffness value used in this case was 1 N/m, which is relatively small, no damping was required. In explicit force control, the HFR keeps moving even though the force remains constant. This is because of the muscle relaxation of the leg. As the muscular forces decrease, the HFR has

to go further to apply the desired amount of force and hence this type of behavior is not very appealing. Instead, it is better to have a little bit of overshoot so as when the muscle relaxes the robot does not have to move as the forces will die down to its steady state value. From the simulation, it can be seen that the HFR is capable of mimicking surgeon's action of *pull* and *rotate*, as the correspondence between the measured force, torque, displacement and angle during the actual surgery and simulated results are good.

5-8 Conclusion

During the study, it was found that the control of position and force for the fracture reduction robot is not well understood. Surgical robot, demands interaction between the robot and humans. Therefore, various control schemes where robot interacts with the environment has been simulated.
As seen in the previous simulations, all the control schemes presented here were able to apply the required amount of force to achieve fracture reduction. However, admittance control stands out as the most suitable control scheme as it is designed to track the force trajectory by regulating the mechanical admittance of the manipulator. Additionally, this approach is more appropriate for the control of HFR because the forces in a conventional surgery vary based on the patient parameters such as gender, age, height, body mass index (BMI), etc. These conditions are not suitable for position controller due to the deviations caused by the reaction force of the patient's leg. Admittance control also has an advantage over Impedance controller as the patient's leg is expected to comply with the movement of HFR, unlike in impedance control, where the robot complies to the target. For example, a window washing robot must comply with its target to prevent breaking the uncompliant glass. Hence, the window washing problem is better solved by using an impedance controller. When the HFR pulls the patient's leg, opposite situation occurs, since the patient's leg must comply to the motion of the HFR. This inverse compliance relationship makes admittance control better suited for the control of HFR. There was no overshoot in admittance and direct force control, while impedance control had 33% overshoot. These control will be beneficial to control HFR as it decreases the soft tissue trauma for the patient. However, the direct force control method is not very appealing when the muscle relaxes, because when the patient's muscle relaxes the reaction force of the leg decreases. To maintain the required amount of force, the position of the robots keep changing, and it takes much time before reaching a steady state value. Since these are only the simulation results, it does not reflect the reality in an actual sense. Chapter 6 compares the theoretical result of simulations with the experimental results from the real HFR setup.

Chapter 6

Practical Experiment and Results

6-1 Introduction

Chapter 3 started off by designing a cost-effective 2-Degree(s) of Freedom (DoF) Hip Fracture Reduction Robot (HFR) taking into account of the safety, hygiene and compatibility constraints of the operation room. This was followed by Chapter 4, where the development of the state-space model for the HFR was carried out by modeling the friction in the design. Finally, in Chapter 5, simulations on the obtained model were performed, to test different controller schemes which can be used in the control of HFR. These simulations are then compared to the force and torque applied by the surgeon during actual surgery. However, these are only the simulation models and the results. It is a moot point as to whether the real setup will always behave like the simulation. In this chapter, various control schemes previously discussed are implemented on the hardware setup, followed by a comparison between two test environments and the simulation results. Even though the forces and torque during the actual surgery can go as high as 600 N and 18 Nm [10] respectively, the experiments conducted on the real setup will be scaled down versions, due to unavailability of appropriate test setup.

6-2 Problem Formulation

The HFR will be in direct contact with the patient. Therefore, it is crucial for the HFR to be compliant, stable and robust enough to prevent damages and hazardous situations for the patient as well as the operating room (OR) staff. In this chapter, various experiments are performed on the HFR connected to an emulated model of the lower limb. The aim of these experiments is to study the influence of different types of interactive controls on the HFR and the lower limb phantom. A comparison between the various control algorithms is carried out to determine the optimum control solution for the system comprising of the HFR and the dynamic lower limb environment.
For the experiments, the velocity signal is obtained by applying a differential operator on the position signal and then using a low-pass filter to remove disturbances from the signal.

6-3 Experimental Setup

Experiments have been conducted on a computer and with the real setup. During the tests, the performance of different force control methods has been compared against each other as well against their simulation results (see Figure 6-1). The experimental setup is shown in Figure 6-2 and 6-3. In the first configuration, a KUKA arm is made to mimic like a lower limb of a patient by making it behave like a virtual spring and a damper. In the second configuration, an elastic element is attached to the end effector of the HFR. The experiments consisted of simulating the *pull* and *rotate* movement of the patient's leg.

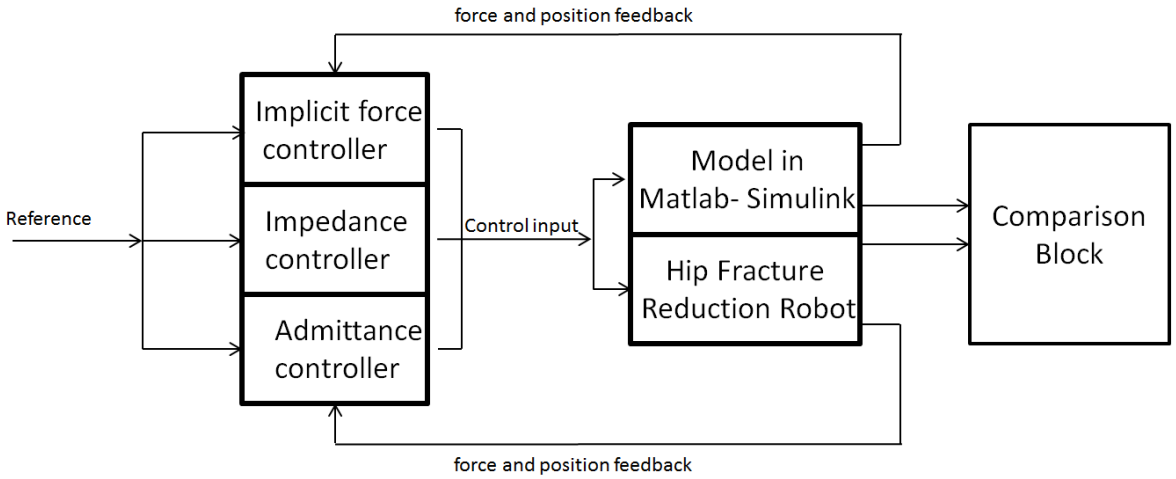


Figure 6-1: A graphical overview of the conducted experiments

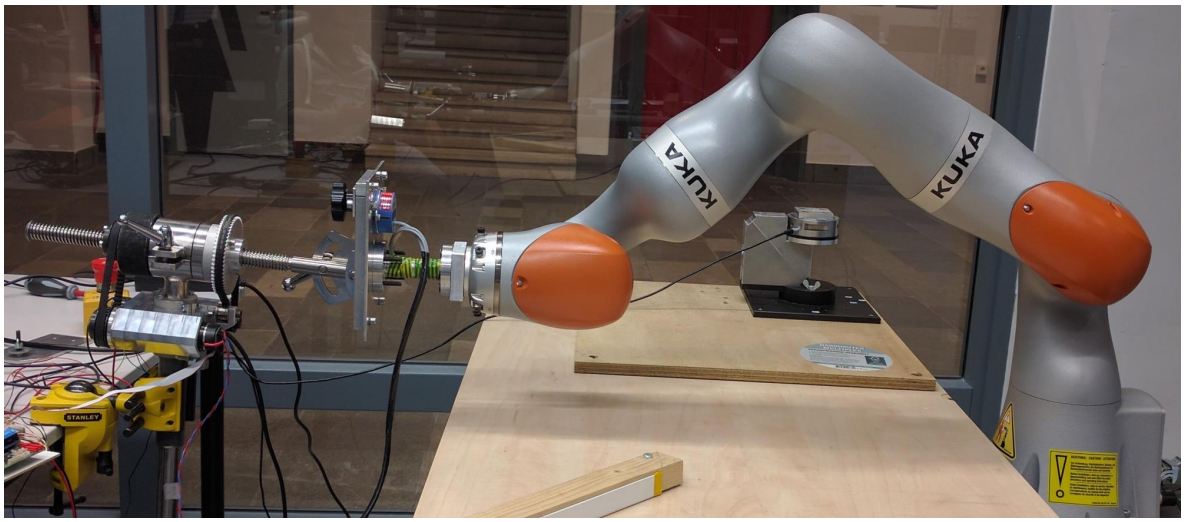


Figure 6-2: Experimental setup with KUKA arm as lower limb phantom

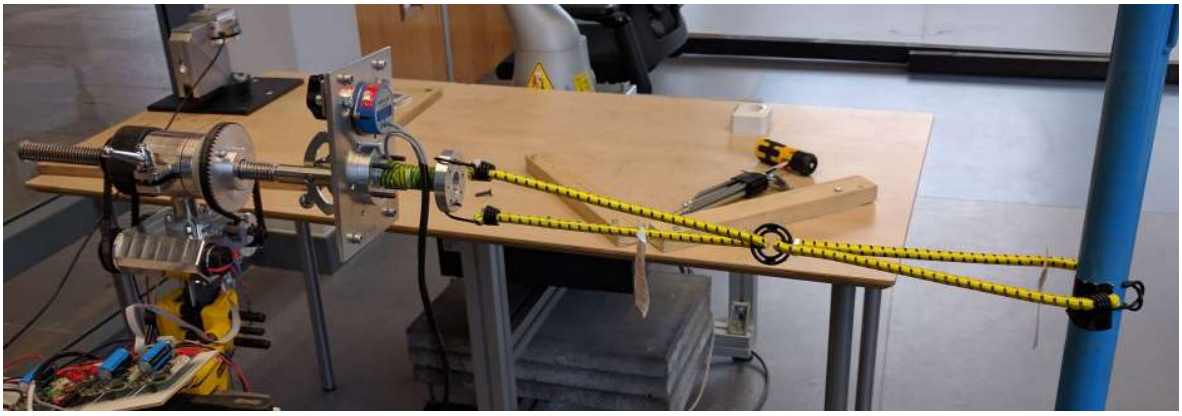


Figure 6-3: Experimental setup with elastic element as lower limb phantom

The average stiffness of the patient's leg during measurements was found to be 4000 N/m [32]. During the surgery, the force and the torque values can go up to 600 N and 18 Nm [10] respectively. However, a KUKA arm can only take force up to 70 N. Hence the setup is tested for scaled down values for the force and the torque. Keeping in mind the safety and the maximum force that can be applied on KUKA, the reference force and torque was selected to be 40 N and 3 Nm respectively. The Cartesian stiffness and wrench of KUKA was set to be 2400 N/m and 3 Nm/rad to see any perceivable movement. However, the damping of the KUKA could not be set, as the KUKA allows setting the damper value only in terms of *Lehr's ratio* and not as linear damping in N/m units. Hence, the lowest possible damping ratio of 0.1 was used for the experiments. The damping value of 500 Ns/m was assumed and used for the simulations. The stiffness of the elastic element was experimentally determined to be approximately around 1200 N/m. To calculate the stiffness of the spring element, steady-state force and position data were used, and an average was taken. The damping value of the spring could not be modeled, as its transient response varied a lot during the experiments. Hence it can be said that the spring element is non-linear. A two- axes Force/Torque (F/T) sensor is used to get the force and torque data along the longitudinal axis. Moving average filter with a window size of 20 is used to filter the F/T data coming from the Ni-DAQ box.

6-4 Force Control

6-4-1 Experiment

Simulations and experiments were performed on both, the KUKA arm and the elastic element. In Section 5-6, it was discussed that Position-plus-Integral (PI) controller shows greater stability and robustness compared to Position-plus-Derivative (PD) controller as the sensor inputs are noisy. Hence direct force control based on a PI controller with anti-windup is implemented here.

For the experiment, a reference signal of 40 N and 3 Nm is given to the HFR. The K_p and K_i value of 0.02 and 0.001 for the linear motion and 0.16 and 0.002 for the rotary motion was selected. Figure 6-4 and 6-5 show the simulation and the experimental result which were performed on the KUKA arm.

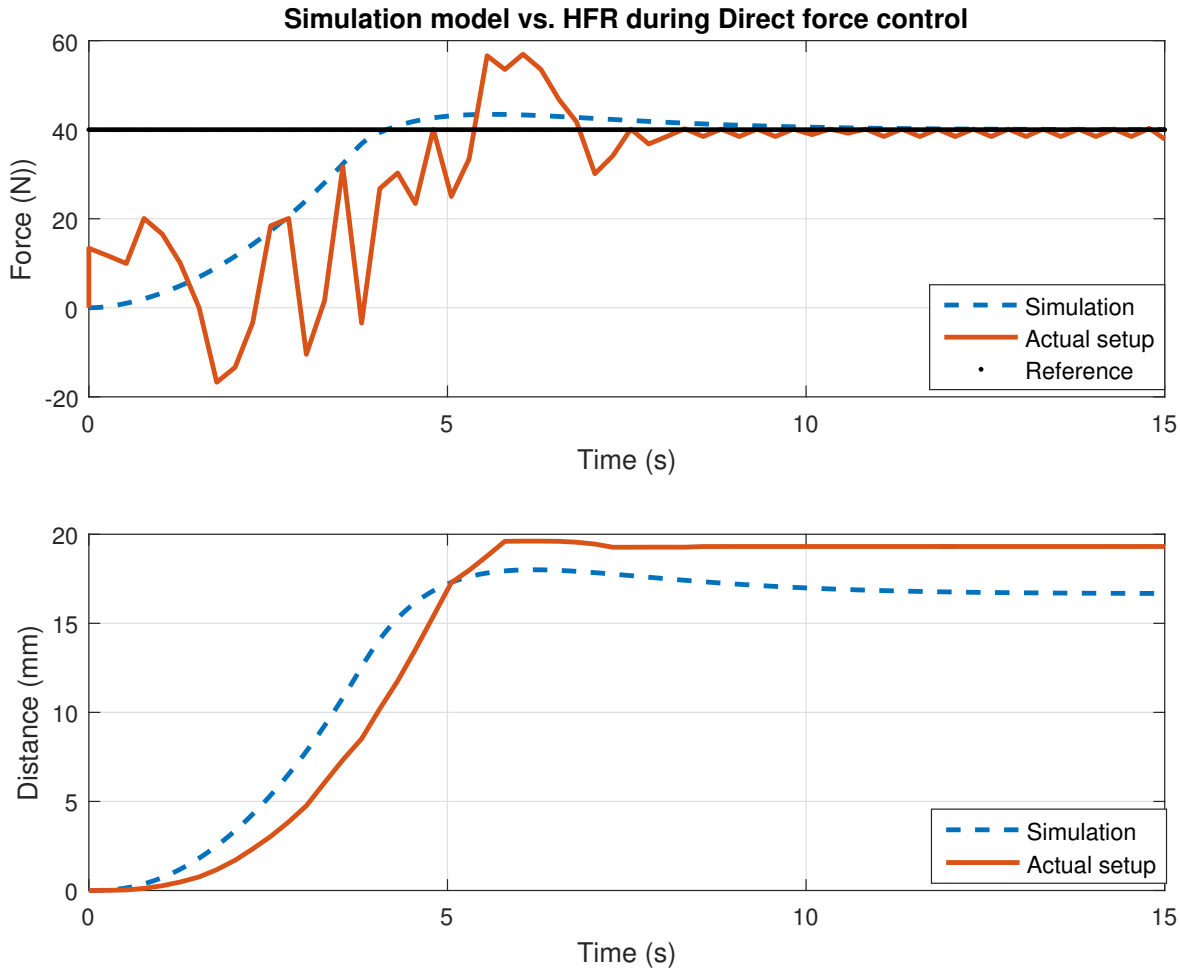


Figure 6-4: Simulation model vs. HFR during pulling of KUKA arm using direct control

6-4-2 Discussion

In the Figure 6-4, difference can be seen in the oscillation of the force signal and the overshoot in the position. The overshoot in the position is due to an error in the reaction force generated by the KUKA arm. Since the stiffness of the KUKA arm was set to be 2400 N/m, to provide a reaction force of 40 N it should be moved by 16.66 mm. However, from the plot, it can be seen that the end effector of KUKA moved by almost 19 mm to provide a reaction force of 40 N.

There is also a lot of oscillation present in the force plot. It is a known fact that signals from force sensors are noisy. However, these signal spikes are huge, especially considering that there is a median filter used for the force signal. The cause for these nasty spikes in force graph was found out to be because of the coupling between the two robots as well as that fact that KUKA is unable to provide constant force during the motion. This kind of abnormal behavior of KUKA was verified by making the HFR stationery and measuring the force at sensor attached to the KUKA arm end effector. It was found out to be noisy and oscillating. This abnormal behavior was further recognized to be originating from KUKA when the same control scheme was used on the elastic spring. Figure 6-6 show the output of experimental

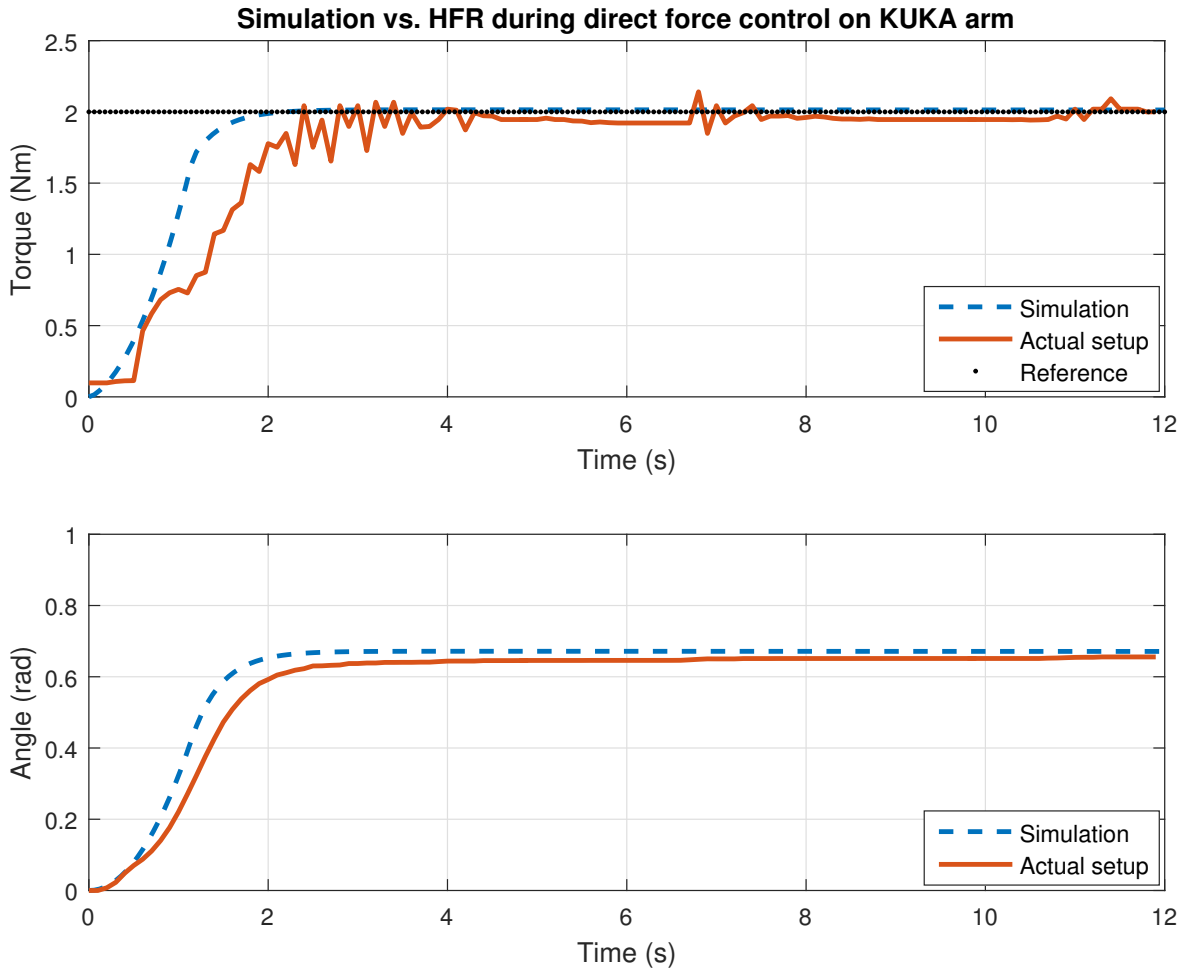


Figure 6-5: Simulation model vs. HFR during rotation of KUKA arm using direct control

result which was performed on the spring based lower limb phantom. It can be clearly seen that the nasty oscillation which was present with the KUKA arm are gone. However, the simulation results and the experimental result differ a lot. The difference in output can be attributed to the nonlinearity of the spring as well as the unknown damper value of the spring. From the above experiments, it can be seen that even though the stiffness of KUKA arm and elastic element were different direct force control was able to apply the desired amount of force on the phantom without changing any controller parameters. A similar experiment was conducted for the rotation motion. For the rotation case, the oscillation in KUKA was relatively less compared to when HFR had pulled it. However, there are still spikes in the force graph. These spikes are due to the noisy force sensor, and due to this reason itself PI controller was used instead of PD. From the above experiments, it can be seen that the correlation between the simulated output and the true output is good.

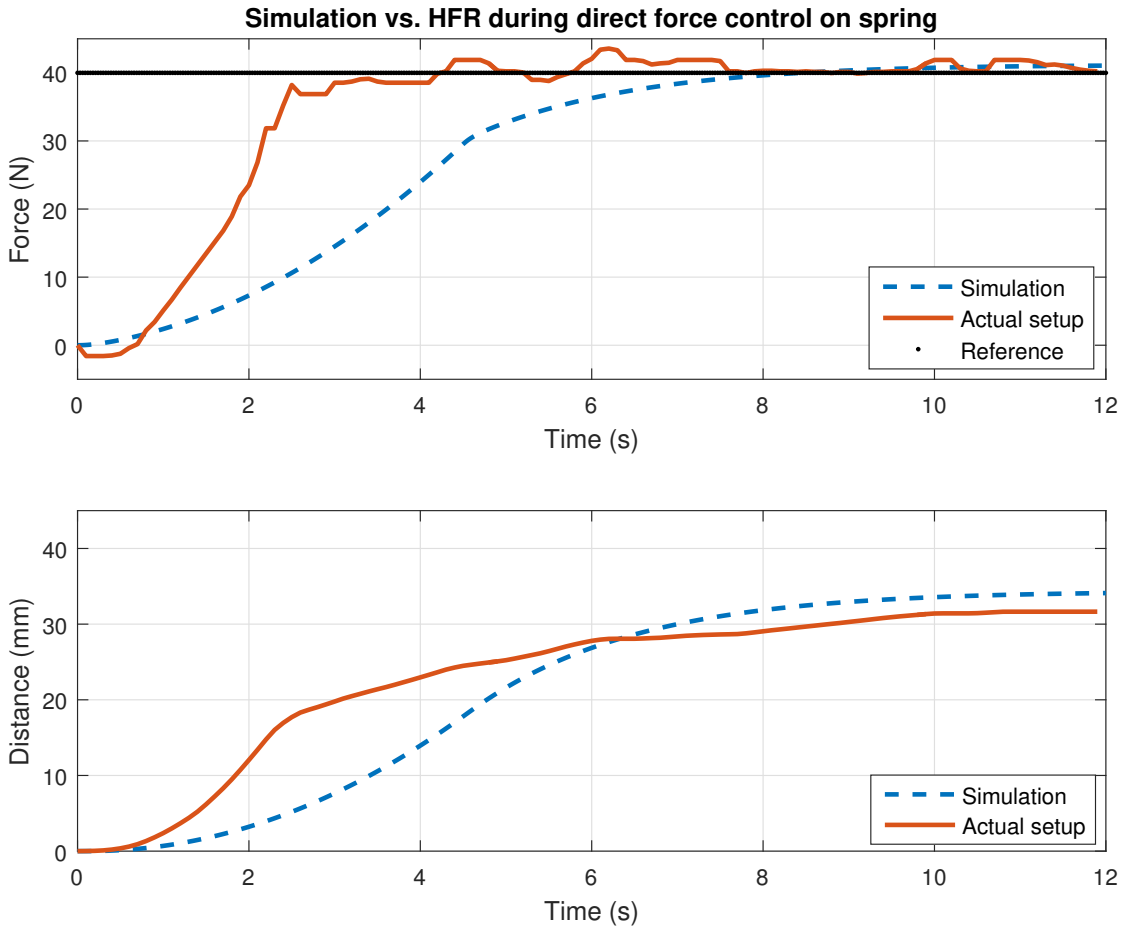


Figure 6-6: Simulation model vs. HFR during linear motion using direct force control on spring element

6-5 Impedance Control

6-5-1 Experiment

The impedance control which was described and simulated in the Section 5-4, has been tested here in simulations and on the test setup. Impedance output variable in Equation 5-7 is used to implement the controller. To achieve this a PI controller is used.

At steady-state, the velocity is zero and the force applied by the HFR depends only on the stiffness K.

$$F_{ss} = K(x_r - x_{ss}) \tag{6-1}$$

where F_{ss} is the steady-state force applied by the HFR, x_{ss} is steady-state end effector position of the HFR and x_r is the desired position of the HFR. Similarly, the force applied by the KUKA arm will only depend on its stiffness. i.e.

$$F_{ss} = k_{ef} x_{ss} \tag{6-2}$$

where k_{ef} is the stiffness of KUKA arm. Combining equation 6-1 and 6-2 , we get

$$\begin{aligned} F_{ss} &= K(x_r - x_{ss}) = k_{ef} x_{ss} \\ x_{ss} &= \frac{K x_r}{K + k_{ef}} \end{aligned} \tag{6-3}$$

It can be observed that if $K >> k_{ef}$ then $x_{ss} \approx x_r$ and if $k_{ef} >> K$ then $x_{ss} \approx 0$.
For the experiment, the target reference force was selected as 40 N and stiffness of the HFR was selected to be 4000 N/m. Since the stiffness of KUKA was set to 2400 N/m, the steady-state displacement of the KUKA arm to apply a 40 N of force can be calculated using Equation 6-2 and was found out to be 16.66 mm. Using these values in Equation 6-1, gives the desired position $x_r = 26.66$ mm, for which HFR will apply a force of 40 N.

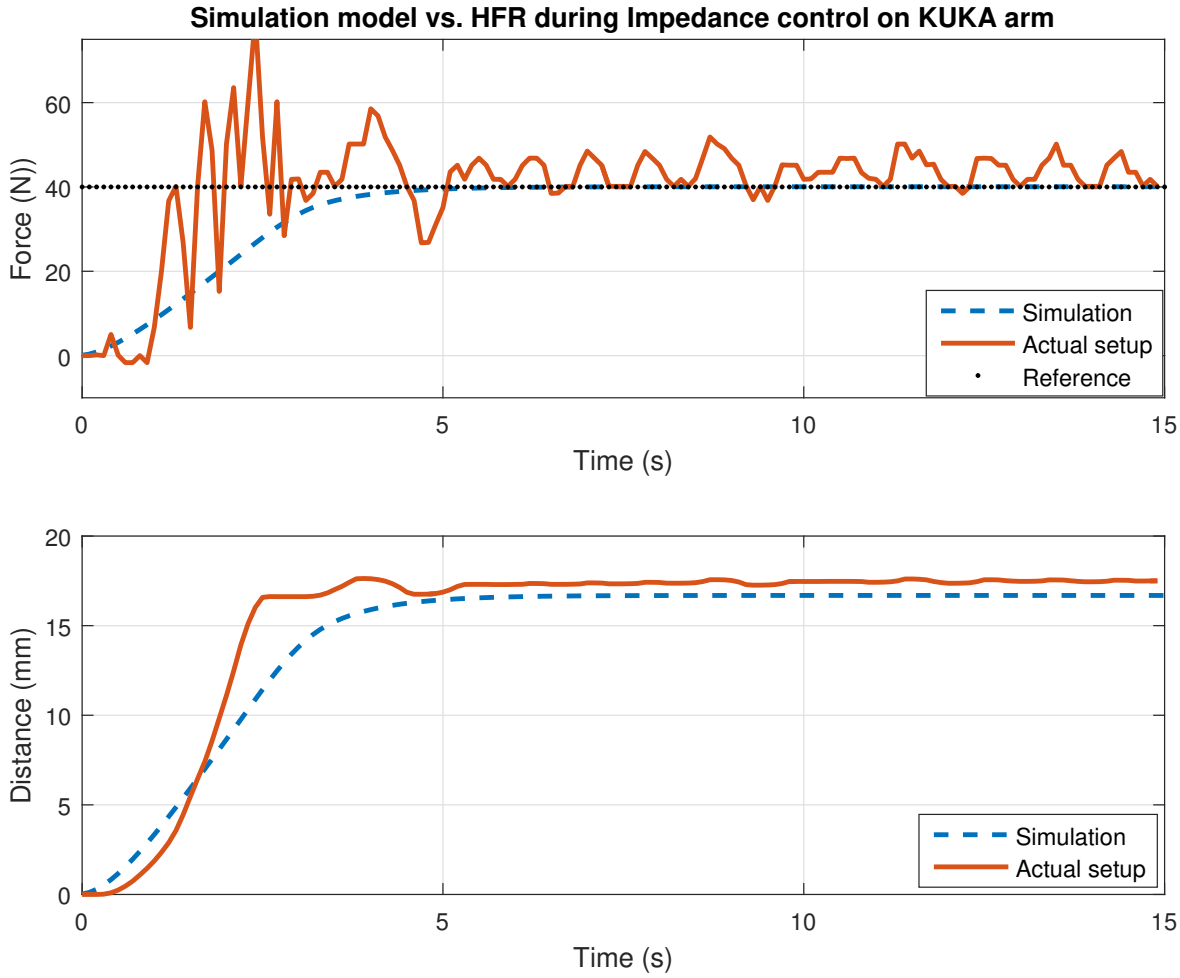


Figure 6-7: Simulation model vs HFR during linear motion using impedance control on KUKA arm

These theoretical results obtained are tested in the simulation and on the experimental setup. A PI controller with K_p and K_i value of 0.03 and 0.0006 is used to regulate the impedance

variable to zero. Figure 6-7 shows the comparison of simulation result and the actual output of the HFR setup. Similar experiments were done with the rotation of the HFR. Figure 6-8 shows the plot of HFR applying torque to rotate the KUKA with a stiffness and damper value of 10 Nm/rad and 0 Nms/rad.

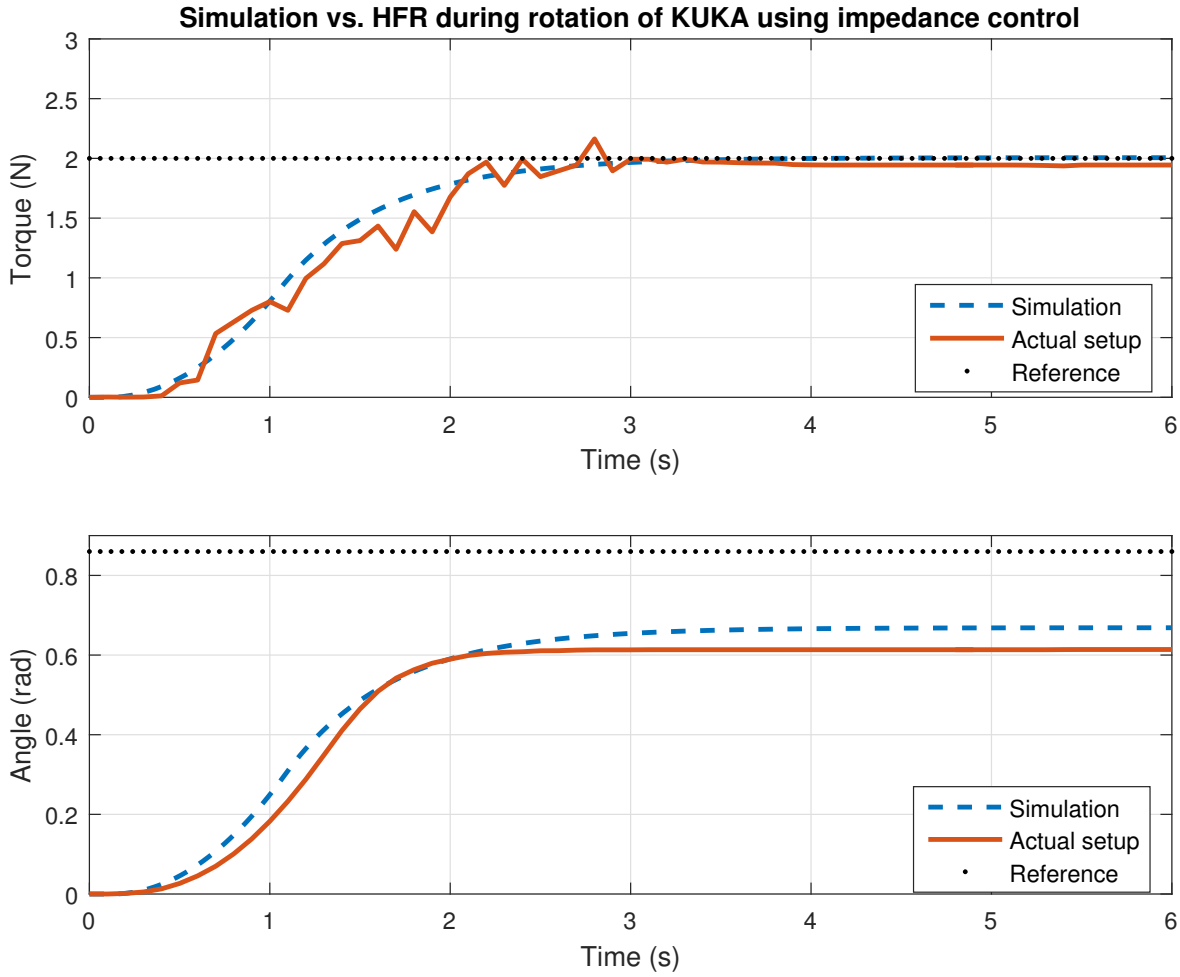


Figure 6-8: Simulation model vs HFR during rotation of KUKA arm using impedance

The tests were further performed on the spring element, since the spring element is more compliant the desired position was increased so the HFR could apply 40 N of force.

The elastic element is attached to the end effector of HFR as a lower limb phantom. The stiffness of the controller is kept the same. Figure 6-9 shows the comparison of simulation result and the actual output of the HFR setup with spring element attached. Experiments were performed to check the influence of the change in impedance parameters. Figure 6-9 shows the effect of changing the stiffness value of the HFR.

6-5-2 Discussion

Figure 6-7 shows that the oscillations that were encountered during direct force control are still present in impedance control. The steady-state values of force and displacement, of the

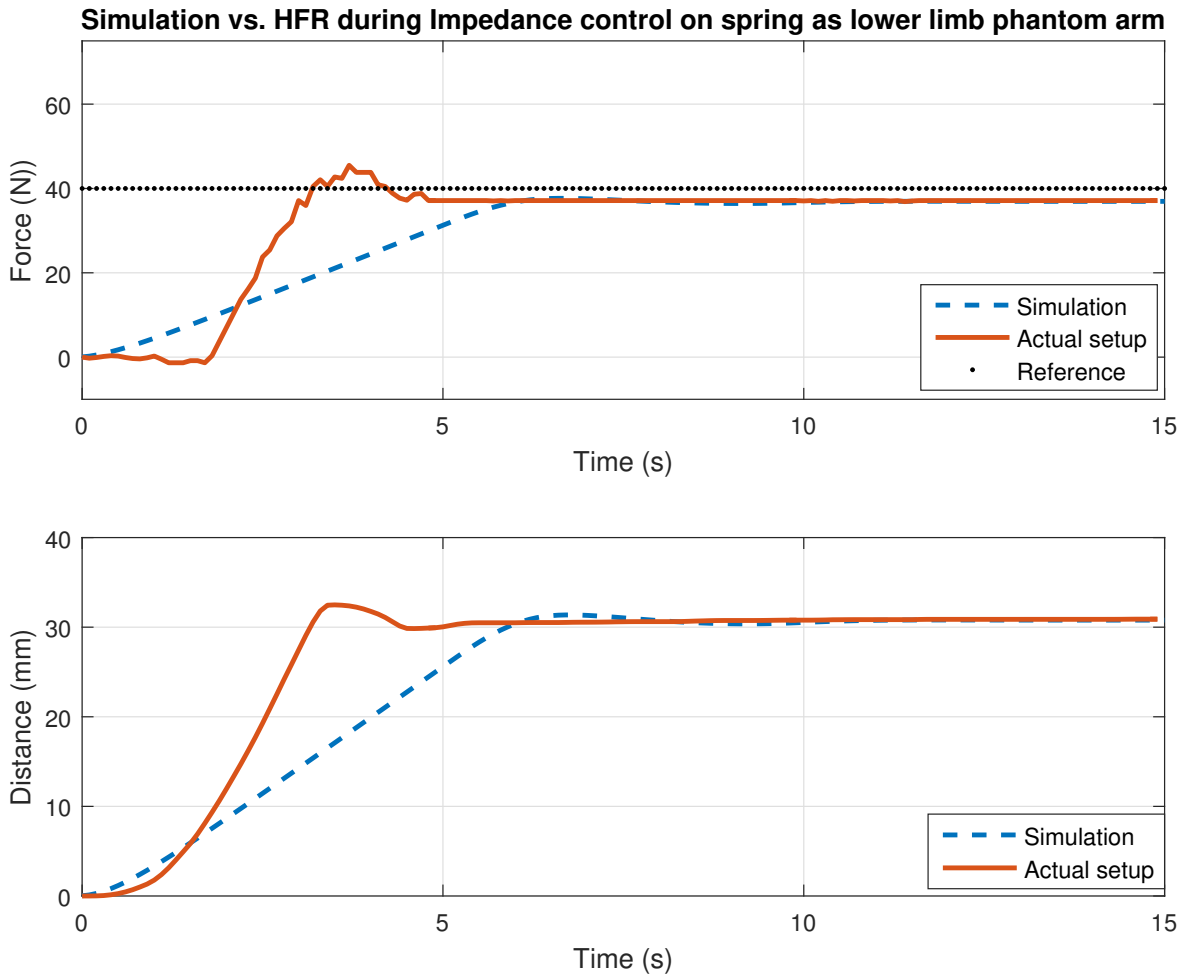


Figure 6-9: Simulation model vs HFR during linear motion using impedance control on elastic element used as lower limb phantom

original setup, matches with that of the simulated data. However, the transient behavior is quite different. This is because of two reasons. First, the setup has unmodeled nonlinear friction which especially becomes prominent at low speed. Secondly, damping value used to model the KUKA arm is not accurate, as only the *Lehr's ratio* is known, as explained in Section 6-3. In Figure 6-8, it can be seen that correspondence between the simulation and real setup is excellent. The small steady-state error of about 0.05 rad could be accounted to the initialization error of the KUKA arm and the backlash present in the HFR. It should be noted that the damping value of zero was used for impedance control, for the rotation of KUKA arm. This is because the rotation motion already has high damping due to the presence of friction and it did not require any additional damping.

Similar experiments were conducted on the spring element. However, there were no nasty oscillations present in this case even with the same impedance parameters of the HFR. Hence the nasty spikes in KUKA can be attributed to the coupling between the robots. Finally, Figure 6-10 shows the effect of changing the impedance of the HFR. From the graph, it can be incurred that as the stiffness K decreases, the HFR becomes more compliant to the

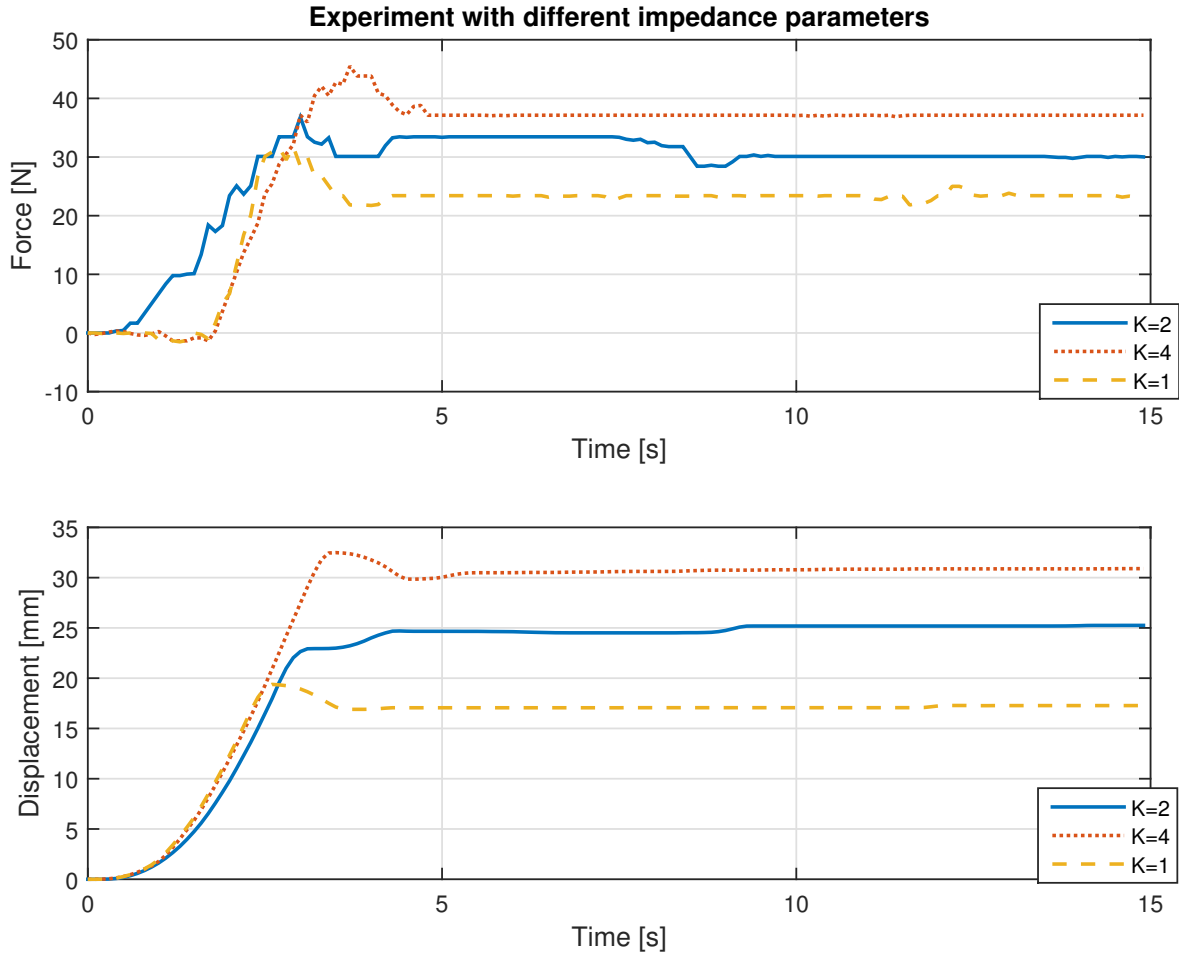


Figure 6-10: Experiment with different impedance parameters. The steady-state force and position are determined by K.

environment. The theoretical calculation which was performed is proved in the simulation as well in the experiments as the HFR was able to apply the desired force of 40 N. From the above experiments, it can be said that the correspondence between the simulated and experimental results are good when the controller is applied to KUKA arm, except for the oscillation. However, these oscillations are not present in the spring element, but since the spring element is nonlinear, its simulated and experimental results are very different.

6-6 Admittance Control

Theoretically, admittance control is an inverse of impedance control. In admittance control, the motion control action and impedance control actions are separated. The motion control is made stiff, but instead of following desired reference trajectory it follows a reference trajectory generated by impedance control action. That is, the desired position and the external force are the input to the controller which generated motion trajectory for the motion control.

6-6-1 Experiment

Simulation and experiments have been performed to check the performance of HFR while applying *pull* and *rotate* action of the KUKA arm using admittance control. A Proportional-Integral-Derivative (PID) controller with $K_p = 0.18$, $K_d = 0.003$ and $K_i = 0.002$ is used here for the motion control. Figure 6-11 shows the performance comparison of the simulated and experimental result. The impedance parameters selected were 12000 N/m and 8000 Ns/m for K and D respectively. The desired position was selected to be 20 mm, such that the HFR will apply a force of 40 N while pulling the KUKA arm.

Similarly for the rotation motion with stiffness and damper values of 10 Nm/rad and 5 Nms/rad respectively. Figure 6-12 shows the comparison between the simulated results and the actual setup.

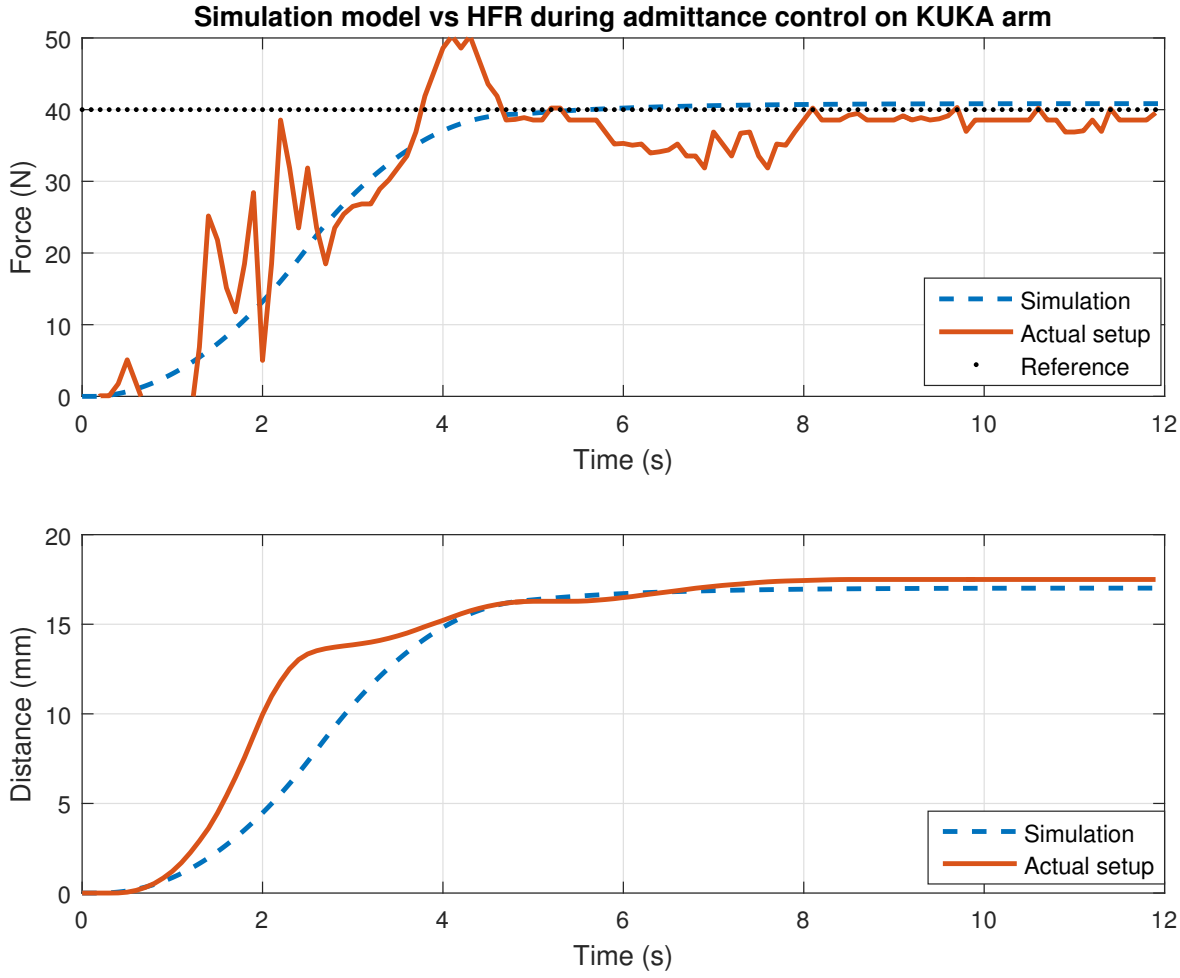


Figure 6-11: Simulation model vs HFR during linear motion using admittance control on KUKA arm

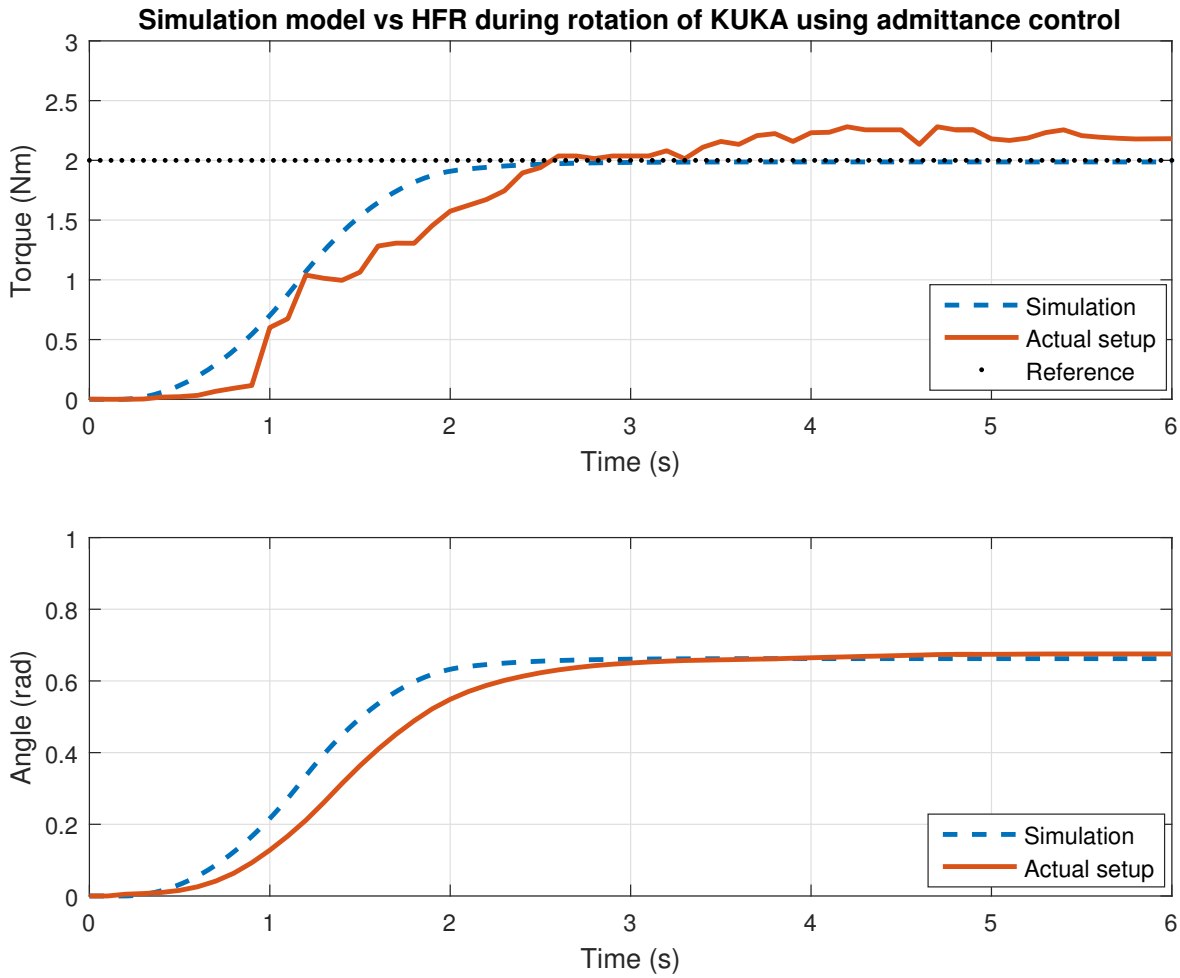


Figure 6-12: Simulation model vs HFR during rotary motion using admittance control on KUKA arm

6-6-2 Discussion

Figure 6-11 and 6-12 shows that the agreement between the simulation result and the experimental results are good. The steady-state value for both the linear and the rotary case matched with the theoretical calculations. However, for the linear motion, the transient behavior of the simulation and the setup differs a bit. This can be attributed to the unknown damping used for the KUKA arm as only *Lehr's ratio* was known. In Figure 6-11 at about 2.5 s, when the force reaches 40 N the controller decreases the speed of HFR. However, this decrease in speed is much more severe compared to the simulation, the reason being the presence of unmodeled nonlinear friction, particularly present at low speed. For the rotary case, the simulation and experimental data mismatch in terms of final steady-state force value. Ideally, if the stiffness of KUKA is 3 Nm/rad, for rotation of 0.66 rad, it should be providing a reaction force of 2 Nm. However, from the Figure 6-12 it can be seen that this is not the case. This error can attribute to the initialization error of the KUKA arm.

6-7 Conclusion

This chapter has been devoted to implementation and comparison of various compliant control schemes. The KUKA arm was used as a lower limb phantom. It was found that the correspondence between the simulation and the experimental data were good. Furthermore, it can be said that the performance obtained in Chapter 5 can be replicated in the real setup given a proper testing condition. For the impedance and admittance control, it has been shown theoretically and experimentally, how the stiffness parameter will determine the steady-state force. The impedance parameter D influences the transient behavior of the system. The drawback of impedance and admittance control is that they can only control the force indirectly by changing the impedance of the robot. Direct force control, on the other hand, will keep the contact force on a specified value, as described in Section 5-6. Since the hip fracture reduction procedure involves *pull* and *rotate* movement on patient's leg, there is fundamental requirement to control force as well as position in a compliant manner. Hence, direct force control is not very appealing, and impedance or admittance control is preferred as these controllers, control the dynamic relationship between force and position. Furthermore, admittance control also has an advantage over Impedance controller as the patient's leg is expected to comply with the movement of HFR, unlike in impedance control, where the robot complies to the target.

To achieve high accuracy trajectory tracking, position controlled robots are designed with high bandwidth. This results in making the robot stiff. This high stiffness is required for good disturbance rejection characteristics, in the presence of unexpected external forces. On the other hand, HFR needs to perform the fracture reduction procedure on a patient's leg in a compliant manner. The performance of HFR will depend on the stiffness of the patient's leg. Conventional performance indices like bandwidth and accuracy cannot be defined for compliant controllers. The idea is to control the stiffness of the robot, and there is an inherent trade-off between the position and force error. For the simulation and experiments conducted with HFR, it was assumed that the stiffness of the patient's leg would be provided by the higher level control. It was possible to theoretically calculate the desired position required for the required stiffness behavior of the robot; such that, it applies the necessary amount of force to the patient's leg. As such, modeling and identification of a patient's lower limb is a difficult task and may not be accurate always, as the patient parameters such as gender, age, height, body mass index (BMI), length of the leg, the thickness of the thigh and muscle strength influence the model. Hence, advanced controllers, that are more robust to modeling errors needs to be explored.

Chapter 7

Conclusion, Future Work and Improvement

7-1 Conclusion

The objective of this thesis was to design a two Degree(s) of Freedom (DoF) Hip Fracture Reduction Robot (HFR) that can apply traction force and rotational torque in a way similar to the procedure currently performed by the surgeon. To meet this objective, a two DoF low-cost, compact and lightweight robot has been designed. The design of the robot draws inspiration from the existing fracture reduction table. One of the advantages of this design is that minimum changes are required to the existing operating room (OR) setup, and the robot can easily be mounted and unmounted from the fracture table as and when required. The design is simple enough that the OR staff would feel encouraged in using it. It also incorporates most of the safety requirements such as the emergency stop button, the current limiter, and the watchdog timer, essential for the utilization of a surgical robot. Integrated with a two axis Force/Torque (F/T) sensor, the designed robot can accurately measure the force and torque that are being applied on the patient's leg. For controlling the HFR, a model of the robot was obtained, which was based on the friction model, and actuator model of the robot. Both simulations and experiments resulted in VAF values above 98%, thus validating that the model captures accurately the system dynamics. It was possible to achieve an accuracy of 1 mm and 2° in the translational and the rotational directions, respectively, using the position control. The implementation of the position controller is provided in Appendix B-7. Table 7-1 shows the specification of the designed HFR.

Compliant control methods such as direct force, impedance, and admittance controls were used to control the HFR. Since there is an inherent requirement to simultaneously control the force and position of the patient's leg, compliant controllers such as impedance or admittance are suitable. These controllers do not explicitly control the contact forces. However, given a perfect model of the lower limb, the desired contact force can be generated by appropriately setting the goal position, using the stiffness of the leg. The performance

Table 7-1: Specification of the designed HFR.

Parameters	Designed-Linear Motion	Designed-Rotary Motion
Peak Force/Torque	1284 N	24 Nm
Range of Motion	150 mm	±360°
Speed of Motion	28.67 mm/s	108°/s
Accuracy	1 mm	1°

of force-controlled robots depends on the stiffness of the environment. To perform fracture reduction autonomously, an accurate model of the lower limb is required. Modeling and identification of a patient's lower limb is a difficult task. Hence, the standard performance indices used to evaluate a control system, i. e., stability, bandwidth, accuracy, and robustness, cannot be defined only for the HFR, but should also refer to the stiffness of the lower limb. During experiments, it was assumed that the model of the lower limb would be provided by high-level control and modeling data from Ravishankar [32] were used to model the KUKA arm. Intuitively, admittance controller suits the requirement of the hip fracture reduction the best, as it is expected that the patient's leg will comply to the motion of the robot. It was also observed, both during the simulation and experiments that the admittance controller was able to apply the required amount of force without any force overshoot. This is beneficial to the patients as it will minimize the soft tissue trauma that happens during the manual procedure. It has been demonstrated that an automatic fracture reduction is possible by using the HFR. The simulation results presented in this thesis has been compared against the performance of the surgeon to perform a feasibility study. From this study, it can be concluded that the fracture reduction can be successfully performed by using the HFR. It was also demonstrated that the performance of simulated robot is similar to that of the real robot.

To finalize the discussion about the design and the control of the HFR, a few technical notes should be mentioned. The current implementation, HFR has following disadvantages:

- The linear motion is limited due to the size of the lead screw available at the fracture table. This limits the workspace of the robot to 150 mm in the linear direction.
- As mentioned earlier, HFR is equipped with various types of safety sensors and devices. However, these safety measures are not sufficient to be used for a clinical trial. Many improvements can be made on the design and the safety of the robot.
- A scaled down test setup was used for performing the experiments. The results obtained shows the potential of the HFR in carrying out the fracture reduction procedure. A complete scaled testing has to be done to demonstrate that, it is feasible to perform the fracture reduction using the HFR.
- After the fracture reduction procedure, the configuration of the legs needs to be maintained to perform next phase of intramedullary fixation. To achieve this mechanical breaks are required. The current HFR does not have the mechanical break for the rotary motion.

7-2 Future Work and Improvements

Future work should be undertaken to continue the development of the presented technology. In particular, improving the current design of the HFR. Safety is the first thing that comes to mind when surgical robotics are talked about. Providing the robot with multiple layers of security will go a long way in the acceptance of the robot. The current design has safety features such as the emergency stop button, the current limiter, the software force and the torque limiter, and the watchdog timer. As a future work, additional safety features such as the hardware force and torque limiter, in the form of mechanical overload absorption socket, should be attached to the foot fixator device. According to the measurements performed by Ravishanker [32], the maximum displacement of the patient's leg can go up to 200 mm. The current iteration of HFR could only provide a linear range of 150 mm. A bigger lead screw needs to be used to cater for the workspace requirement of the HFR.

The current HFR design has an inherent backlash due to wear and tear of the fracture table. This backlash needs to be removed by upgrading the old parts. As mentioned in Chapter 1, after the application of traction and rotation of the leg, the configuration of the leg is maintained, to perform the next phase of surgery i.e. the intramedullary fixation. To maintain this configuration of the leg, mechanical breaks needs to be installed in the HFR. Such a break is only required in case of rotary motion as the linear motion has static loading property due to the presence of lead screw.

Another improvement that can be made in the design of HFR is by making use of *Series Elastic Actuation* (SEA) instead of the present rigid mechanism. In SEA an elastic element with known stiffness, such as a spring, is added to the actuation. By measuring the change in length of the spring, the force and the torque can be computed. The end effector of HFR can be provided with passive compliance, by making use of SEA. By using SEA the force control problem can be converted to a position controlled problem. Therefore, SEA is particularly beneficial for force control problem, as is the case in the HFR. However, the feasibility for such an actuation needs to be tested, as the amount of force and torque required during fracture reduction are high. Additional improvements in the design include installation of the absolute encoders to perform a successful homing maneuver.

Surgical robots require rigorous testing and clinical trials before getting accepted. The testing methodology used for the HFR is very rudimentary in comparison to the prescribed level. The future experiment should be done on a one on one scale. A realistic lower limb phantom should be used to test the performance of the HFR. It would be beneficial to install a mock OR with a complete model of the fracture table and a mock Fluoroscopic C-arm. Since it is not possible to use an actual Fluoroscopic C-arm, cameras can be used to provide the vision.

The low-level control is currently implemented using the conventional force control methods. These control methods are not immune to the modeling errors, and there is room for improvement. It would be beneficial to include advanced force control methods, which are more robust to modeling errors such as the adaptive force control, the robust force control, and the force control based on learning.

The developed prototype is not without its flaws, and even with future improvements, it will be challenging for the HFR to finish the fracture reduction in an entirely independent mannerly. An intuitive surgeon-machine interface, based on the collaborative control will be helpful in this regard. In this type of control, the surgeon acts like a limited source of

information and the robot does most of the tasks, autonomously. However, based on the feedback from various surgeons, a conventional human machine interface will not be much appreciated, as the surgeon is not actively involved in the surgical procedure. To gain the acceptance of the surgical community, innovative technologies should be used. One such technology is the haptics shared control. The haptics shared control can put together the inputs of an experienced surgeon with the robot's dexterity. Implementing a haptic shared control strategy by which the control authority shifts continuously between the surgeon and the HFR would be beneficial. The surgeon may express his control intentions in a way that either conforms or overrides the action of HFR. A haptic steering wheel could be attached to the end effector of the HFR through which the surgeon can aid HFR in completing the surgery.

Appendix A

M-files

A-1 Listing for Impedance Control

```
% Impedance controller for Linear motion used with KUKA
%% Constants and definitions
Ts = 0.1;                                                % sampling period [s]
Tf = 15;                                                 % experiment duration [s]

%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                               % final number of
    samples
Ni=1;
umax=0.5;

%% History variables
t = Ts*(0:Nf-1)';                                        % time vector
force = zeros(length(t),1);                                  % initialize
    variables to log data
td = force; e=force; SumError=force; de=force;I=force;w=force;angle=force;
             % monitor elapsed time
time = force;u = force;force_virtual= force;

try
    spine('CloseAll');
    [O, S] = spine('Open', 'SPINE1');
catch
    disp('Warning: the SPINE is not connected to the computer.');
    %MOPSconnected = 0;
end;
```

```
%distance_d=50;
% x1=[0:0.335:40]';
% x2=40*ones(30,1);
% distance_d=[x1;x2]';
% fd = 40;
distance_d=25*ones(length(t),1);
r=40*ones(length(t),1);
%xd=(distance_d*2*pi)/4; % Reference Angle at Gear shaft
%% Define Controller Constants
Kp=0.03; % Proportional Constant
Kd= 0.0082;  % Differential Constant
Ki=0.0006; % Integral Constant
%% Impedence values
% K=3.3;
% b=0.4;
K=2;
b=3;

%% Initialize plots
lw = 2;
subplot(2,1,2);
h1 = animatedline;
axis([0 Tf -10 80]); grid on;
xlabel('Time [s]'); ylabel('Distance [mm]')

subplot(2,1,1);
h2 = animatedline;
if exist('r','var')
    hold on
    stairs(t,r,'r-','linewidth',lw);
    hold off
end;
%h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
axis([0 Tf 0 80]); grid on;
xlabel('Time [s]'); ylabel('Force [N]')

%% Each HIP2 gets a 4 element array: [PWM CurrentLimit 164EncoderReset Options]
%       PWM: -1 <= value <= 1
%       CurrentLimit: FPGA based fast absolute current limit, restarts
%                     PWM cycle
%       EncoderReset: 1 = reset immediately
%                     2 = reset on next index pulse
%                     3 = reset now, start counting on next index
%                     +4 = only use external index (ignore encoder)
%       Options:      1 = disable timeout
%                     -1 = tristate PWM bridge (timeout immediately)
%% Each HIP2 returns a 7 element array: [Status Current Temperature Encoder1
```

```
    Encoder2]
%       Status:       bit 0 = PWM okay
%                     bit 1 = temparature overload m1(1) and m2(1)
%       Current:      motor current in Amperes m1(2) and m2(2)
%       Analog:       analog input voltage m1(3) and m2(3)
%       Temperature1: amplifier temperature in Celcius m1(4) and m2(4)
%       Temperature2: motor temperature in Celcius m1(5) and m2(5)
%       Encoder1:     Position from encoder 1 in radians m1(6) and m2(6)
%       Encoder2:     Position from encoder 2 in radians m1(7) and m2(7)

try

    s = daq.createSession('ni');
    addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
    addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
    s.DurationInSeconds = 0.03;
    s.Rate = 5000;

catch
    disp('Warning: the DAQ box is not connected to the computer.');
    %MOPSconnected = 0;
end;
%% Start the DAQ session
s = daq.createSession('ni');
global globalT;
global globalF;
global globalTime;
%% open two analog input channel
addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
s.IsContinuous = true;
s.Rate = 5000;
s.NotifyWhenDataAvailableExceeds = 250;
lh = addlistener(s,'DataAvailable', @storeData1);
s.startBackground();
fcalibrate= globalF;
tic;
pause(0.06);
for k=Ni:Nf,
    force(k)=globalF-fcalibrate;
    % Read and write to motor driver
    [~, m2]=spine('setget',0,[0 10 0 0],[u(k) 10 0 0]);
    I(k)= m2(2);
    m2(6)=m2(6)*(500)/(512);
    angle(k) = m2(6);
    force_virtual(k)=K*(distance_d(k)-m2(6)*2/pi)+b*(0.335/Ts-w(k));
    e(k)=force_virtual(k)-force(k); % error
```

```
118     if k > Ni
119         addpoints(h2,Ts*k,force(k));
120         addpoints(h1,Ts*k,angle(k)*2/pi);
121         %set(h2,'xdata',Ts*[k-1 k],'ydata',[angle(k-1) angle(k)]);  % set angle
                in the plot
122                 drawnow;
123         w(k)=(angle(k)-angle(k-1))/0.3;
124         %set(h1,'xdata',Ts*[k-1 k],'ydata',[y(k-1) y(k)]);   % set force in the
                plot
125         % Find the error
126         SumError(k)=e(k)+SumError(k-1);
127         de(k)=(e(k)-e(k-1))/0.3;
128     end
129     drawnow;
130     u(k+1)=Kp*e(k)+Ki*SumError(k);%+Kd*de(i+1); % Controller
131     if(u(k+1)>umax)
132         u(k+1)=umax;
133         SumError(k)=0;
134     elseif(u(k+1)<-umax)
135         u(k+1)=-umax;
136         SumError(k)=0;
137     end
138     time(k)=toc;
139     %while toc < Ts; end;
140     %pause(0.06)
141     while toc < Ts; end;
142     tic;
143 end
144 pause(1)
145 [~, ~]=spine('setget',O,[0 10 1 0],[0 10 1 0]);
146 % m2(6)=m2(6)*(500)/(512);
147 % angle(i)=m2(6);
148 % amp(i)=m2(2);
149 subplot(2,1,1),plot(t,force)
150 subplot(2,1,1),hold on
151 subplot(2,1,1),plot(t,r,':','linewidth',lw);
152 subplot(2,1,1),title('Force experienced by the KUKA based phantom');
153 subplot(2,1,1),xlabel('Time (s)');
154 subplot(2,1,1),ylabel('Force (N))');
155 subplot(2,1,1),legend('Observed','Reference');
156 subplot(2,1,1),grid on;
157 subplot(2,1,2),plot(t,angle*2/pi);
158 subplot(2,1,2),title('Distance moved by the hip fracture robot');
159 subplot(2,1,2),xlabel('Time (s)');
160 subplot(2,1,2),ylabel('Distance (mm)');
161 subplot(2,1,2),grid on;
```

```
% Impedance controller for Rotary motion used with KUKA
%% Constants and definitions
Ts = 0.1;                                          % sampling period [s]
Tf = 12;                                           % experiment duration [s]

%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                          % final number of
    samples
Ni=1;
umax=0.3;

%% History variables
t = Ts*(0:Nf-1)';                                    % time vector
torque = zeros(length(t),1);                              % initialize
    variables to log data
td = torque; e=torque; SumError=torque; de=torque;I=torque;w=torque;angle=
    torque;          % monitor elapsed time
time = torque;u = torque;torque_virtual= torque;

try
    spine('CloseAll');
    [O, S] = spine('Open', 'SPINE1');
catch
    disp('Warning: the SPINE is not connected to the computer.');
    %MOPSconnected = 0;
end;
%angle_d=50;
% x1=[0:0.335:40]';
% x2=40*ones(30,1);
% angle_d=[x1;x2]';
% fd = 40;
angle_d=0.87*ones(length(t),1);
r=2*ones(length(t),1);
%xd=(angle_d*2*pi)/4; % Reference Angle at Gear shaft
%% Define Controller Constants
Kp=0.05; % Proportional Constant
Kd= 0.0001;  % Differential Constant
Ki=0.0001; % Integral Constant
%% Impedence values
% K=3.3;
% b=0.4;
K=10;
b=0;

%% Initialize plots
lw = 2;
```

```
45 subplot(2,1,2);
46 h1 = animatedline;
47 axis([0 Tf 0 1]); grid on;
48 xlabel('Time [s]'); ylabel('Distance [mm]')
49
50 subplot(2,1,1);
51 h2 = animatedline;
52 if exist('r','var')
53     hold on
54     stairs(t,r,'r-','linewidth',lw);
55     hold off
56 end;
57 %h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
58 axis([0 Tf 0 3]); grid on;
59 xlabel('Time [s]'); ylabel('torque [Nm]')
60
61 %% Each HIP2 gets a 4 element array: [PWM CurrentLimit 164EncoderReset Options]
62 %       PWM: -1 <= value <= 1
63 %       CurrentLimit: FPGA based fast absolute current limit, restarts
64 %                     PWM cycle
65 %       EncoderReset: 1 = reset immediately
66 %                     2 = reset on next index pulse
67 %                     3 = reset now, start counting on next index
68 %                     +4 = only use external index (ignore encoder)
69 %       Options:      1 = disable timeout
70 %                     -1 = tristate PWM bridge (timeout immediately)
71 %% Each HIP2 returns a 7 element array: [Status Current Temperature Encoder1
       Encoder2]
72 %       Status:       bit 0 = PWM okay
73 %                     bit 1 = temparature overload m1(1) and m2(1)
74 %       Current:      motor current in Amperes m1(2) and m2(2)
75 %       Analog:       analog input voltage m1(3) and m2(3)
76 %       Temperature1: amplifier temperature in Celcius m1(4) and m2(4)
77 %       Temperature2: motor temperature in Celcius m1(5) and m2(5)
78 %       Encoder1:     Position from encoder 1 in radians m1(6) and m2(6)
79 %       Encoder2:     Position from encoder 2 in radians m1(7) and m2(7)
80
81 try
82     %% Start the DAQ session
83     s = daq.createSession('ni');
84     global globalT;
85     global globalF;
86     global globalTime;
87     %% open two analog input channel
88     addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
89     addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
90     s.IsContinuous = true;
```

```
91      s.Rate = 5000;
92      s.NotifyWhenDataAvailableExceeds = 250;
93      lh = addlistener(s,'DataAvailable', @storeData1);
94      s.startBackground();
95
96  catch
97      disp('Warning: the DAQ box is not connected to the computer.');
98      %MOPSconnected = 0;
99  end;
100
101
102 fcalibrate=globalT;
103 tic;
104 pause(0.06);
105 for k=Ni:Nf,
106     torque(k)=globalT-fcalibrate;
107 %     % Read and write to motor driver
108     [m1,~]=spine('setget',0,[-u(k) 10 0 0],[0 10 0 0]);
109     I(k)= m1(2);
110     m1(6)=m1(6)*(624*125)/(35*13824);
111     angle(k) = m1(6);
112     torque_virtual(k)=K*(angle_d(k)-angle(k)/4)-b*w(k);
113     e(k)=torque_virtual(k)-torque(k); % error
114     if k > Ni
115         addpoints(h2,Ts*k,torque(k));
116         addpoints(h1,Ts*k,angle(k)/4);
117         %set(h2,'xdata',Ts*[k-1 k],'ydata',[angle(k-1) angle(k)]);  % set angle
                in the plot
118         drawnow;
119         w(k)=(angle(k)-angle(k-1))/(4*0.3);
120         %set(h1,'xdata',Ts*[k-1 k],'ydata',[y(k-1) y(k)]);   % set torque in
               the plot
121         % Find the error
122         SumError(k)=e(k)+SumError(k-1);
123         de(k)=(e(k)-e(k-1))/0.3;
124     end
125     drawnow;
126     u(k+1)=Kp*e(k)+Ki*SumError(k);%+Kd*de(i+1); % Controller
127     if(u(k+1)>umax)
128         u(k+1)=umax;
129         SumError(k)=0;
130     elseif(u(k+1)<-umax)
131         u(k+1)=-umax;
132         SumError(k)=0;
133     end
134     time(k)=toc;
135     %while toc < Ts; end;
```

```
    %pause(0.06)
    while toc < Ts; end;
    tic;
end
pause(1)
[~, ~]=spine('setget',O,[0 10 1 0],[0 10 1 0]);
% m2(6)=m2(6)*(500)/(512);
% angle(i)=m2(6);
% amp(i)=m2(2);
subplot(2,1,1),plot(t,torque)
subplot(2,1,1),hold on
subplot(2,1,1),plot(t,r,':','linewidth',lw);
subplot(2,1,1),title('torque experienced by the KUKA based phantom');
subplot(2,1,1),xlabel('Time (s)');
subplot(2,1,1),ylabel('torque (N))');
subplot(2,1,1),legend('Observed','Reference');
subplot(2,1,1),grid on;
subplot(2,1,2),plot(t,angle/4);
subplot(2,1,2),title('Distance moved by the hip fracture robot');
subplot(2,1,2),xlabel('Time (s)');
subplot(2,1,2),ylabel('Angle (mm)');
subplot(2,1,2),grid on;
```

A-2 Listing for Admittance Control

```
% Admittance controller for Linear motion used with KUKA
%% Constants and definitions
Ts = 0.1;                                          % sampling period [s]
Tf = 12;                                           % experiment duration [s]

%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                          % final number of
    samples
Ni=1;
umax=0.4;

%% History variables
t = Ts*(0:Nf-1)';
% time vector
force = zeros(length(t),1);                                % initialize
    variables to log data
td = force; e=force; SumError=force; de=force;I=force;w=force;angle=force;
            % monitor elapsed time
time = force;u = force;force_virtual= force;xd_d= force; xd= force; x= force;
acc= force; xdd_d= force;xdd= force;
```

```
19
20 try
21     spine('CloseAll');
22     [O, S] = spine('Open', 'SPINE1');
23 catch
24     disp('Warning: the SPINE is not connected to the computer.');
25     %MOPSconnected = 0;
26 end;
27 try
28     %% Start the DAQ session
29     s = daq.createSession('ni');
30     global globalF; %#ok<TLEV>
31     %% open two analog input channel
32     addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
33     addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
34     s.IsContinuous = true;
35     s.Rate = 5000;
36     s.NotifyWhenDataAvailableExceeds = 250;
37     lh = addlistener(s,'DataAvailable', @storeData1);
38     s.startBackground();
39
40 catch
41     disp('Warning: the DAQ box is not connected to the computer.');
42     %MOPSconnected = 0;
43 end;
44
45 %% Impedence values
46 K=12;
47 b=8;
48 M=2;
49 % d=F/K+pred_leg_movement;
50 % x1=[0:max_speed*0.2:d]';
51 % x2=d*ones(150,1);
52 % x_d=[x1;x2]';
53 x_d=36.6*ones(length(t),1);
54 % r=36*ones(length(t),1);
55 % x_d=40*ones(1,200);
56 %xd=(distance_d*2*pi)/4; % Reference Angle at Gear shaft
57 %% Define Controller Constants
58 Kp=0.018; % Proportional Constant
59 Kd= 0.0003;  % Differential Constant
60 Ki=0.0002;%0.0003; % Integral Constant
61
62 %% Initialize plots
63 lw = 2;
64 subplot(3,1,2);
65 h1 = animatedline;
```

```
66 axis([0 Tf 0 50]); grid on;
67 xlabel('Time [s]'); ylabel('Distance [mm]')
68
69 subplot(3,1,1);
70 h2 = animatedline;
71
72 % if exist('r','var')
73 %     hold on
74 %     stairs(t,r,'r-','linewidth',lw);
75 %     hold off
76 % end;
77 %h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
78 axis([0 Tf 0 50]); grid on;
79 xlabel('Time [s]'); ylabel('force [N]')
80 subplot(3,1,3);
81 h3 = animatedline;
82 axis([0 Tf 0 60]); grid on;
83
84
85 fcalibrate= globalF;
86
87 tic;
88 pause(0.06);
89 for k=Ni:Nf,
90     force(k)=globalF-fcalibrate;
91     % Read and write to motor driver
92     [~, m2]=spine('setget',0,[0 10 0 0],[u(k) 10 0 0]);
93
94     I(k)= m2(2);
95     m2(6)=m2(6)*(500)/(512);
96     angle(k) = m2(6);
97     xdd(k)=0 - (force(k)-K*(x_d(k)-x(k))-b*(0-xd(k)))/M;
98     xd(k+1)=xd(k)+xdd(k)*Ts;
99     x(k+1)=x(k)+xd(k)*Ts;
100    e(k)=x(k)-angle(k)*2/pi; % error
101    if k > Ni
102        %% Admittance Controller
103        w(k)=(angle(k)*2/pi-angle(k-1)*2/pi)/(Ts);
104        addpoints(h2,Ts*k,force(k));
105        addpoints(h1,Ts*k,angle(k)*2/pi);
106        addpoints(h3,Ts*k,x(k));
107        SumError(k)=e(k)+SumError(k-1);
108        de(k)=(e(k)-e(k-1))/Ts;
109        %xd_d(k+1)=(x_d(k+1)-x_d(k))/Ts;
110        acc(k)=(w(k)-w(k-1))/Ts;
111        %xdd_d(k+1)=(xd_d(k+1)-xd_d(k))/Ts;
112    end
```

```
    drawnow;
    u(k+1)=Kp*e(k)+Ki*SumError(k)+Kd*de(k); % Controller
    if(u(k+1)>umax)
        u(k+1)=umax;
        SumError(k)=0;
    elseif(u(k+1)<-umax)
        u(k+1)=-umax;
        SumError(k)=0;
    end
    time(k)=toc;
    while toc < Ts; end;
    tic;
end
pause(1)
[~, ~]=spine('setget',0,[0 10 1 0],[0 10 1 0]);
subplot(2,1,1),plot(t,force)
subplot(2,1,1),title('force experienced by the KUKA based phantom');
subplot(2,1,1),xlabel('Time (s)');
subplot(2,1,1),ylabel('force (N))');
subplot(2,1,1),grid on;
distance_moved=angle.*(4/(2*pi));
subplot(2,1,2),plot(t,distance_moved);
subplot(2,1,2),hold on
subplot(2,1,2),plot(t,x_d,':');
subplot(2,1,2),title('Distance moved by the hip fracture robot');
subplot(2,1,2),xlabel('Time (s)');
subplot(2,1,2),ylabel('Angle (rad)');
subplot(2,1,2),legend('Observed','Reference');
subplot(2,1,2),grid on;
```

```
% Admittance controller for Rotary motion used with KUKA
%% Constants and definitions
Ts = 0.1;                                          % sampling period [s]
Tf = 12;                                           % experiment duration [s]

%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                          % final number of
    samples
Ni=1;
umax=0.3;

%% History variables
t = Ts*(0:Nf-1)';
% time vector
torque = zeros(length(t),1);                                % initialize
    variables to log data
```

```
16 td = torque; e=torque; SumError=torque; de=torque;I=torque;w=torque;angle=
       torque;          % monitor elapsed time
17 time = torque;u = torque;torque_virtual= torque;xd_d= torque; xd= torque; x=
       torque;
18 acc= torque; xdd_d= torque;xdd= torque;
19
20 try
21     spine('CloseAll');
22     [O, S] = spine('Open', 'SPINE1');
23 catch
24     disp('Warning: the SPINE is not connected to the computer.');
25     %MOPSconnected = 0;
26 end;
27 try
28     %% Start the DAQ session
29     s = daq.createSession('ni');
30     global globalF; %#ok<TLEV>
31     global globalT;
32     %% open two analog input channel
33     addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
34     addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
35     s.IsContinuous = true;
36     s.Rate = 5000;
37     s.NotifyWhenDataAvailableExceeds = 250;
38     lh = addlistener(s,'DataAvailable', @storeData1);
39     s.startBackground();
40
41 catch
42     disp('Warning: the DAQ box is not connected to the computer.');
43     %MOPSconnected = 0;
44 end;
45
46 %% Impedence values
47 K=10;
48 b=5;
49 M=2;
50 % d=F/K+pred_leg_movement;
51 % x1=[0:max_speed*0.2:d]';
52 % x2=d*ones(150,1);
53 % x_d=[x1;x2]';
54 x_d=0.86*ones(length(t),1);
55 % r=36*ones(length(t),1);
56 % x_d=40*ones(1,200);
57 %xd=(distance_d*2*pi)/4; % Reference Angle at Gear shaft
58 %% Define Controller Constants
59 Kp=0.2; % Proportional Constant
60 Kd= 0.00;  % Differential Constant
```

```
61  Ki=0.0012;%0.0003; % Integral Constant
62
63  %% Initialize plots
64  lw = 2;
65  subplot(3,1,2);
66  h1 = animatedline;
67  axis([0 Tf 0 1]); grid on;
68  xlabel('Time [s]'); ylabel('Angle [rad]')
69
70  subplot(3,1,1);
71  h2 = animatedline;
72
73  % if exist('r','var')
74  %     hold on
75  %     stairs(t,r,'r-','linewidth',lw);
76  %     hold off
77  % end;
78  %h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
79  axis([0 Tf 0 3]); grid on;
80  xlabel('Time [s]'); ylabel('Torque [Nm]')
81  subplot(3,1,3);
82  h3 = animatedline;
83  axis([0 Tf 0 1]); grid on;
84
85  Tcalibrate=globalT;
86  tic;
87  pause(0.06);
88  for k=Ni:Nf,
89      torque(k)= globalT-Tcalibrate;
90      % Read and write to motor driver
91      [m1, ~]=spine('setget',0,[-u(k) 10 0 0],[0 10 0 0]);
92
93      I(k)= m1(2);
94      m1(6)=m1(6)*(624*125)/(35*13824);;
95      angle(k) = m1(6);
96      xdd(k)=0 - (torque(k)-K*(x_d(k)-x(k))-b*(0-xd(k)))/M;
97      xd(k+1)=xd(k)+xdd(k)*Ts;
98      x(k+1)=x(k)+xd(k)*Ts;
99      e(k)=x(k)-angle(k)/4; % error
100     if k > Ni
101         %% Admittance Controller
102         w(k)=(angle(k)-angle(k-1))/(4*Ts);
103         addpoints(h2,Ts*k,torque(k));
104         addpoints(h1,Ts*k,angle(k)/4);
105         addpoints(h3,Ts*k,x(k));
106         SumError(k)=e(k)+SumError(k-1);
107         de(k)=(e(k)-e(k-1))/Ts;
```

```
        %xd_d(k+1)=(x_d(k+1)-x_d(k))/Ts;
        acc(k)=(w(k)-w(k-1))/Ts;
        %xdd_d(k+1)=(xd_d(k+1)-xd_d(k))/Ts;
    end
    drawnow;
    u(k+1)=Kp*e(k)+Ki*SumError(k)+Kd*de(k)+0.03; % Controller
    if(u(k+1)>umax)
        u(k+1)=umax;
        SumError(k)=0;
    elseif(u(k+1)<-umax)
        u(k+1)=-umax;
        SumError(k)=0;
    end
    time(k)=toc;
    while toc < Ts; end;
    tic;
end
pause(1)
[~, ~]=spine('setget',O,[0 10 1 0],[0 10 1 0]);
subplot(2,1,1),plot(t,torque)
subplot(2,1,1),title('torque experienced by the KUKA based phantom');
subplot(2,1,1),xlabel('Time (s)');
subplot(2,1,1),ylabel('torque (Nm))');
subplot(2,1,1),grid on;

subplot(2,1,2),plot(t,angle/4);
subplot(2,1,2),hold on
subplot(2,1,2),plot(t,x_d,':');
subplot(2,1,2),title('Distance moved by the hip fracture robot');
subplot(2,1,2),xlabel('Time (s)');
subplot(2,1,2),ylabel('Angle (rad)');
subplot(2,1,2),legend('Observed','Reference');
subplot(2,1,2),grid on;
```

A-3 Listing for Direct Force

```
% Direct force control linear
%% Constants and definitions
Ts = 0.1;                                          % sampling period [s]
Tf = 12;                                             % experiment duration
    [s]

%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                          % final number of
    samples
```

```
Ni=1;
umax=0.5;

%% History variables
t = Ts*(0:Nf-1)';                                    % time vector
force = zeros(length(t),1);                              % initialize
    variables to log data
td = force; e=force; SumError=force; de=force;I=force;w=force;angle=force;
            % monitor elapsed time
time = force;u = force;

try
    spine('CloseAll');
    [O, S] = spine('Open', 'SPINE1');
catch
    disp('Warning: the SPINE is not connected to the computer.');
    %MOPSconnected = 0;
end;
try
    %% Start the DAQ session
    s = daq.createSession('ni');
    global globalF; %#ok<TLEV>
    %% open two analog input channel
    addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
    addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
    s.Rate = 5000;
    s.IsContinuous = true;
    s.NotifyWhenDataAvailableExceeds = 250;
    lh = addlistener(s,'DataAvailable', @storeData1);
    s.startBackground();
catch
    disp('Warning: the DAQ box is not connected to the computer.');
    %MOPSconnected = 0;
end;

% %% Initialize online plot and onscreen stop button
% USER_STOP = 0;                                       % a flag to stop RT
    control
% figure(1); clf;                                      % open figure for the
     button
% ssize = get(0,'screensize');                         % screen size
% set(gcf,'pos',[ssize(3)-575 ssize(4)-25-800 575 800]);  % set figure position
% set(1,'MenuBar','none','NumberTitle','off','Name',...
%     'Real-time experiment running ...');             % hide menus and set
    title
% stop_h = uicontrol( ...                              % define the
    pushbutton
```

```
50 %     'Interruptible','on', ...
51 %     'Style','pushbutton', ...
52 %     'BackgroundColor','c', ...
53 %     'ForegroundColor','k', ...
54 %     'FontSize',15, ...
55 %     'FontWeight','bold', ...
56 %     'Units','normalized', ...
57 %     'UserData',0, ...
58 %     'Position',[.37 .94 .3 .04], ...
59 %     'String','STOP', ...
60 %     'Callback','assignin(''base'',''USER_STOP'',1)');   % if pressed, set
       USER_STOP = 1
61
62 %% Initialize plots
63 lw = 2;
64 subplot(2,1,2);
65 h1 = animatedline;
66 axis([0 Tf -5 60]); grid on;
67 xlabel('Time [s]'); ylabel('Distance [mm]')
68
69 subplot(2,1,1);
70 h2 = animatedline;
71 if exist('r','var')
72     hold on
73     stairs(t,r,'r-','linewidth',lw);
74     hold off
75 end;
76 %h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
77 axis([0 Tf 0 60]); grid on;
78 xlabel('Time [s]'); ylabel('Force [N]')
79 %% Define Controller Constants
80 Kf=164.16;
81 Ktau=2.387;
82 %distance_d=50;
83 fd=40; % Reference Force at Gear shaft
84 r=fd*ones(Nf,1);
85 K=0.02; % Proportional Constant
86 Kd= 0.0082;  % Differential Constant
87 Ki=0.001; % Integral Constant
88 %% Each HIP2 gets a 4 element array: [PWM CurrentLimit EncoderReset Options]
89 %       PWM: -1 <= value <= 1
90 %       CurrentLimit: FPGA based fast absolute current limit, restarts
91 %                     PWM cycle
92 %       EncoderReset: 1 = reset immediately
93 %                     2 = reset on next index pulse
94 %                     3 = reset now, start counting on next index
95 %                     +4 = only use external index (ignore encoder)
```

```
%       Options:      1 = disable timeout
%                     -1 = tristate PWM bridge (timeout immediately)
%% Each HIP2 returns a 7 element array: [Status Current Temperature Encoder1 Encoder2]
%       Status:       bit 0 = PWM okay
%                     bit 1 = temparature overload m1(1) and m2(1)
%       Current:      motor current in Amperes m1(2) and m2(2)
%       Analog:       analog input voltage m1(3) and m2(3)
%       Temperature1: amplifier temperature in Celcius m1(4) and m2(4)
%       Temperature2: motor temperature in Celcius m1(5) and m2(5)
%       Encoder1:     Position from encoder 1 in radians m1(6) and m2(6)
%       Encoder2:     Position from encoder 2 in radians m1(7) and m2(7)
%% Initialize loop vatiables
u(1)=0.4; % speed (PWM value can varry from = -1 to 1)
% m2(6)=m2(6)*(500)/(512);% Gear ratio needs to be changed as spine board is hard coded for n= 624/35
% angle(i)=m2(6); % angle
%for i=3:500

fcalibrate= globalF;
tic;
pause(0.06);
for k = Ni : Nf,
    force(k)=globalF-fcalibrate;
    e(k)=fd-force(k);
    % Read and write to motor driver
    [m1, m2]=spine('setget',0,[0 10 0 0],[u(k) 10 0 0]);
    I(k)= m2(2);
    m2(6)=m2(6)*(500)/(512);
    angle(k) = m2(6);
    if k > Ni
        addpoints(h2,Ts*k,force(k));
        addpoints(h1,Ts*k,angle(k)*2/pi);
        %set(h2,'xdata',Ts*[k-1 k],'ydata',[angle(k-1) angle(k)]);  % set angle in the plot
        drawnow;
        w(k)=(angle(k)-angle(k-1))/0.3;
        %set(h1,'xdata',Ts*[k-1 k],'ydata',[y(k-1) y(k)]);   % set force in the plot
        % Find the error
        SumError(k)=e(k)+SumError(k-1);
        de(k)=(e(k)-e(k-1))/0.3;
    end
    drawnow;
    u(k+1)=K*e(k)+Ki*SumError(k);%+Kd*de(i+1)+Ki*SumError(i+1); % Controller
    if(u(k+1)>umax)
        u(k+1)=umax;
```

```
        SumError(k)=0;
    elseif(u(k+1)<-umax)
        u(k+1)=-umax;
        SumError(k)=0;
    end
    %     if USER_STOP,
    %         disp('Interrupted by user.');              % message
    %         close(1);                                  % close button figure
    %         break;                                     % exit control loop
    %     end;
    time(k)=toc;
    %while toc < Ts; end;
    %pause(0.06)
    while toc < Ts; end;
    tic;
end
pause(1)
figure(1); clf;
[~, ~]=spine('setget',O,[0 10 1 0],[u(k) 10 1 0]);
%[m1 m2]=spine('setget',O,[0 10 1 0],[0 10 1 0]);
% m2(6)=m2(6)*(500)/(512);
% angle(k)=m2(6);
% amp(k)=m2(2);
subplot(2,1,1),plot(t,force)
subplot(2,1,1),hold on
subplot(2,1,1),plot(t,r,':','linewidth',lw);
subplot(2,1,1),title('Force experienced by the four spring based phantom');
subplot(2,1,1),xlabel('Time (s)');
subplot(2,1,1),ylabel('Force (N)');
subplot(2,1,1),legend('Observed','Reference');
subplot(2,1,1),grid on;
subplot(2,1,2),plot(t,angle*2/pi);
subplot(2,1,2),title('Distance moved by the hip fracture robot');
subplot(2,1,2),xlabel('Time (s)');
subplot(2,1,2),ylabel('Distance (mm)');
subplot(2,1,2),grid on;
%print('Directforcecontrol4spring','-dpng');
% print('Directforcesimulationspring','-depsc');
```

```
%% Constants and definitions
Ts = 0.1;                                          % sampling period [s]
Tf = 12;                                           % experiment duration
    [s]
%% Computed parameters, etc.
% All kinds of parameters can be derived from the above, such as:
Nf = Tf/Ts;                                        % final number of
    samples
```

```
Ni=1;
umax=0.3;

%% History variables
t = Ts*(0:Nf-1)';                                          % time vector
% initialize variables to log data
e=zeros(length(t),1);  % error
SumError=e;% Summation of Error
de=e; % derivative of Error
w=e; % omega (speed)
acc=e; % acceleration
angle=e; % angle
amp=e; % current
u=e;
time=e;
%% Open new connection and create a spine object O
try
    spine('CloseAll');
    [O, S] = spine('Open', 'SPINE1');
catch
    disp('Warning: the SPINE is not connected to the computer.');
    %MOPSconnected = 0;
end

%% Define Controller Constants
Kf=164.16;
Ktau=2.387;
taud=0;                                    % Reference Angle at Gear shaft
r=taud*ones(Nf,1);
Kp=0.16;                                     % Proportional Constant
%Kd= 0.03;                                   % Differential Constant
Ki=0.002;                                   % Integral Constant

%% Initialize plots
lw = 2;
subplot(2,1,2);
h1 = animatedline;
if taud>0
axis([0 Tf 0 60]); grid on;
else
axis([0 Tf -60 0]); grid on;
end
xlabel('Time [s]'); ylabel('Angle [deg]')

subplot(2,1,1);
h2 = animatedline;
if exist('r','var')
```

```
54      hold on
55      stairs(t,r,'r-','linewidth',lw);
56      hold off
57  end;
58  %h1 = plot([0 0],[0 0],'b-','EraseMode','none','linewidth',lw);
59  if taud>0
60  axis([0 Tf 0 3]); grid on;
61  else
62  axis([0 Tf -3 0]); grid on;
63  end
64  xlabel('Time [s]'); ylabel('Torque [N]')
65
66  %% Start the DAQ session
67  s = daq.createSession('ni');
68  global globalT;
69  global globalF;
70  global globalTime;
71  %% open two analog input channel
72  addAnalogInputChannel(s,'Dev1', 0, 'Voltage');
73  addAnalogInputChannel(s,'Dev1', 1, 'Voltage');
74  s.IsContinuous = true;
75  s.Rate = 5000;
76  s.NotifyWhenDataAvailableExceeds = 250;
77  lh = addlistener(s,'DataAvailable', @storeData1);
78  s.startBackground();
79  fcalibrate= 0;
80  tic; % starts timer
81  pause(0.06);
82  %% Calculate force offset
83  % fcalibrate= globalT;
84
85  for i=Ni:Nf,
86      blahmasd(i)=globalT;
87      e(i)=taud-blahmasd(i);
88      [m1, ~]=spine('setget',0,[-u(i) 10 0 0],[0 10 0 0]);
89      m1(6)=m1(6)*(624*125)/(35*13824);
90      angle(i)=m1(6);
91      amp(i)=m1(2);
92      if i>1
93          %addpoints(h2,Ts*[i-1 i],[blahmasd(i-1) blahmasd(i)]);
94          addpoints(h2,Ts*i,blahmasd(i));
95          addpoints(h1,Ts*i,rad2deg(angle(i)/4));
96          %addpoints(h1,Ts*[i-1 i],[rad2deg(angle(i-1)) rad2deg(angle(i))]);
97          drawnow;
98          SumError(i)=e(i-1)+SumError(i-1);
99          %         de(i)=(e(i)-e(i-1))/Ts;
100         %         w(i)=(angle(i)-angle(i-1))/Ts;
```

```
101         %         acc(i)=(w(i)-w(i-1))/Ts;
102     end
103     u(i+1)=Kp*e(i)+Ki*SumError(i);%+Kd*de(i+1); % Controller
104     if(u(i)>umax)
105         u(i)=umax;
106         SumError(i)=0;
107     elseif(u(i)<-umax)
108         u(i)=-umax;
109         SumError(i)=0;
110     end
111 %     if USER_STOP,
112 %         disp('Interrupted by user.');              % message
113 %         close(1);                                  % close button figure
114 %         break;                                     % exit control loop
115 %     end;
116 
117     %pause(0.06);
118     while toc < Ts; end;
119     time(i)=toc;
120     tic;
121 end;
122 [~, ~]=spine('setget',O,[0 10 1 0],[0 10 1 0]);
123 subplot(2,1,1),plot(t,blahmasd,'linewidth',lw)
124 subplot(2,1,1),hold on
125 subplot(2,1,1),plot(t,r,':');
126 subplot(2,1,1),title('Torque experienced by KUKA as lowerlimb phantom');
127 subplot(2,1,1),xlabel('Time (s)');
128 subplot(2,1,1),ylabel('Torque (Nm))');
129 subplot(2,1,1),legend('Observed','Reference');
130 subplot(2,1,1),grid on;
131 
132 subplot(2,1,2),plot(t,(rad2deg(angle))/4);
133 subplot(2,1,2),title('Angle moved by the hip fracture robot');
134 subplot(2,1,2),xlabel('Time (s)');
135 subplot(2,1,2),ylabel('Angle (deg)');
136 subplot(2,1,2),grid on;
137 delete(lh);
```

Appendix B

Miscellaneous

B-1 Sensors

Safety is one of the primary requirement of Hip Fracture Reduction Robot (HFR). The HFR while performing the reduction should not apply any extra force and torque to the patient's leg. The measurement was performed to find the maximum force and Torque that can be applied during reduction and it was found out to be 548N and 17.4Nm. Hence, the force and torque that is applied by robot need to be measured. Sensors requirements are to measure the force and torque of about 600N and 20Nm respectively with an accuracy of about 1-2%. It was found out that the Force and torque sensor that is required for our requirements are expensive. To reduce the overall cost of the robot, it was decided that the sensors should be built in-house. It was decided to make use of Strain Gauge as Force Sensor. Since there is no moving part in it and they are small in size as well as relatively inexpensive, they are perfect for our requirement. The disadvantages they have are that they are non-linear and it needs to be calibrated.

B-2 Strain gauge

Strain gauge converts mechanical force to an electrical signal. A strain gauge is a long length of conductor arranged in a zigzag pattern on a membrane, and it works on the principle of change in electrical resistance. When a mechanical element subjects to tension or a compression the electric resistance of the material changes. This is used to measure the force acted upon the element. Figure B-1 shows one such strain gauge.

B-2-1 Basic Principle of Strain gauge load cell

When any material is subjected to force, it tends to change in dimension. Hence if the strain gauges are bonded to this material, the strain gauge also is stretched or compressed, causing

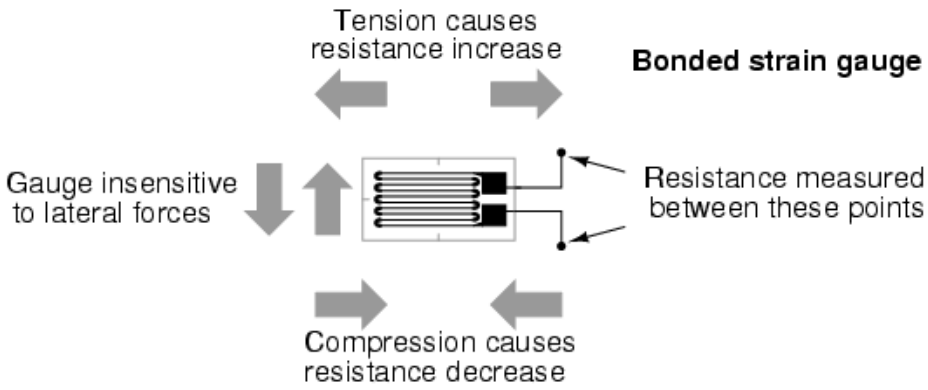


Figure B-1: Strain Gauges

a change in its length and diameter. This change in dimension of the strain gauge causes its resistance to change. This change in resistance or output voltage of the strain gauge becomes a measure of applied force.

Figure B-2 shows a typical load cell in which a cylinder is made up of known material, whose young's modulus is known. Four strain gauges are bonded to this cylinder such that two of them are mounted along the direction of force applied, and the other two strain gauges are mounted at right angle to the previous to gauges.

B-2-2 Operation of strain gauge Load cell:

The four strain gauges are connected in such a way that they form a Wheatstone bridge. The output voltage is measured as shown in Figure B-3.

Scenario 1: When there is no force on the cylinder, all the strain gauges have the same resistance, and the Wheatstone bridge will remain balanced. Hence the output voltage will be zero.

Scenario 2: When the axial force is applied for example a compressive force, the Strain gauges that are connected axially undergo compressive strain. Hence their resistance will decrease. The gauges that are connected transverse will go under tensile strain, and their resistance will increase. The Wheatstone bridge is not balanced anymore and hence an output voltage is generated. The change in output voltage due to the applied force becomes a measure of the applied load force when calibrated.

B-3 Torque sensing using strain gauge

The principle for measurement of torque using strain gauges is similar to that of force measurement. The only difference is the placement of the strain gauges. To measure Torque, the strain gauges are installed to the shaft at 45 degrees as shown in Figure B-5.

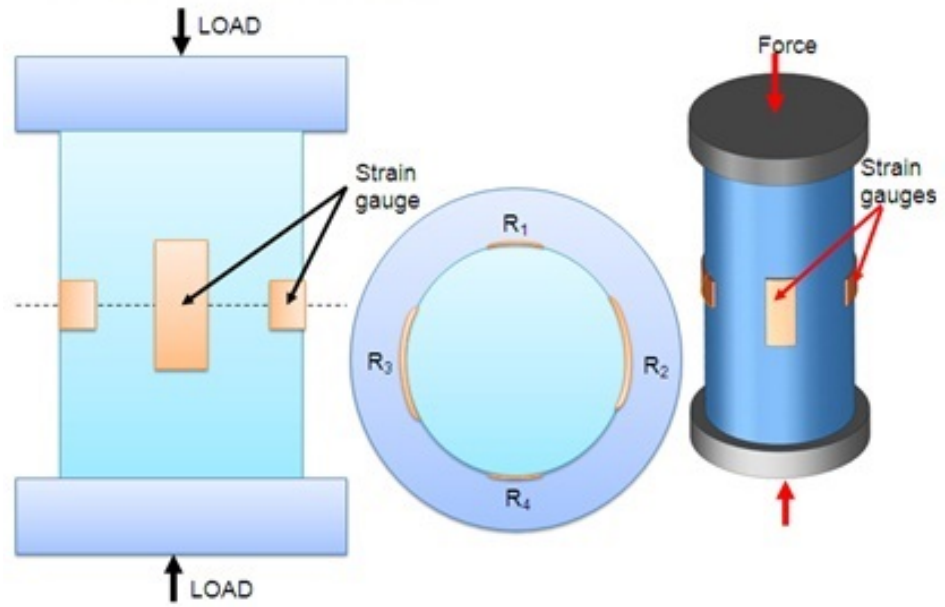


Figure B-2: Construction of load cell

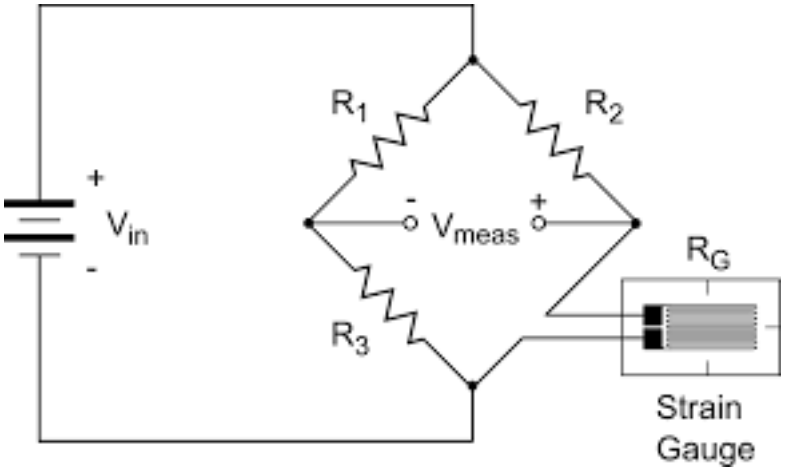


Figure B-3: Wheatstone bridge with strain gauges

Once strain gauges are installed, they are connected, such that they form Wheatstone bridge. The measurement is done similar to what we did with the force measurement setup. When a torque is applied to the cylinder, a torsion is created in the axis as shown in Figure B-6.

The strain gauges one and three will expand, and their resistances will increase. The gauges 2 and four will compress, and their resistance will decrease. The Wheatstone bridge will not be balanced anymore, and the output voltage can be measured. The change in output voltage due to the applied torque becomes a measure of the applied load torque when calibrated.

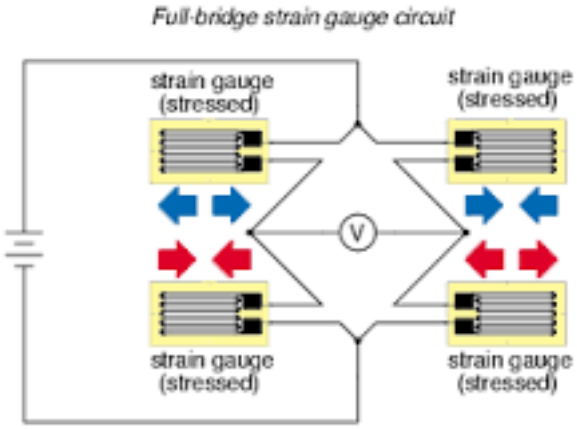


Figure B-4: Full bridge Strain gauge

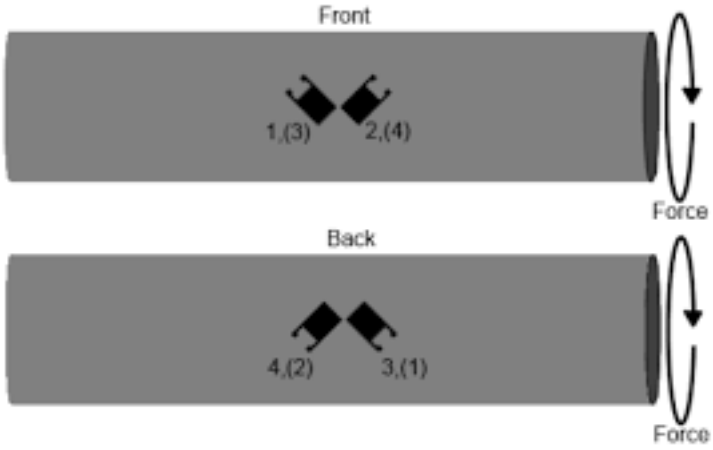


Figure B-5: Full bridge Strain gauge

B-4 Designing the spring element

The first step in designing the load is the selection of the spring material. Spring element is the material that takes the force being measured and convert it into a linear extension. It realizes on the elastic property of the material to measure the force indirectly. The strain gauges will be bonded to the spring element and as the spring elements expand so will, the strain gauge will. Despite the name, the spring element should be chosen for minimal compliance i.e. minimal displacement under rated load. The spring element should appear to be completely rigid under full load; the strain gauge is capable of measuring imperceptible deflection in the spring element, and anything more has a good chance of causing damage to the spring element which would permanently damage the load cell.
The material we will use is Aluminum. We need to design the spring element such that it maintains its elastic property when maximum load is applied.

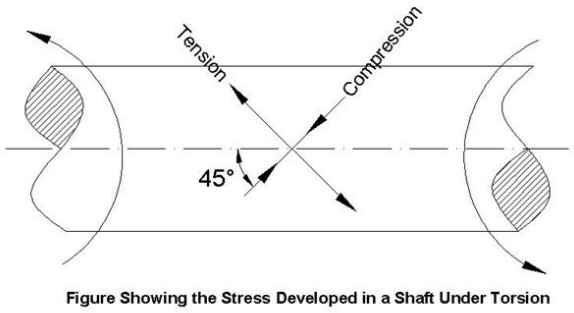


Figure B-6: Full bridge Strain gauge

B-5 Installing Strain Gauges

The first step is to use some sand paper or steel wool to rough up the surface of the spring element. Since we are going to be using glue to hold the strain gauge in place, we want to ensure the best adhesion possible. Not only does sanding the metal a bit remove any foreign material from the surface, but it also increases the surface area exposed to give the glue more to grab on to. Once we are done with this, we need to make sure that the area is clean. Any powder left behind needs to be removed before we move forward. We used alcohol ass the cleaning agent since it will evaporate quickly and not leave any residue behind that might interfere with the glue curing process.

Since we need to mount strain gauges for both force and the torque measurement, we will mount them at the opposite ends of the spring element. We will connect them such that they form Wheatstone bridge when connected. It is critical to mount the strain gauges for force and torque at 90° and 45° respectively as explained earlier. Hence marking was made before gluing the strain gauges, and we lined up the gauges as best as we could. Super glue was used to mount the strain gauges to the elastic element. Now that the strain gauge is mounted, we solder the wires onto the terminal tab to connect it to amplifier

Finally, we will want to protect the strain gauge. Hence, a protective coating is provided to the strain gauge. We used epoxy resin and made three layers if protective coating.

B-6 Calibration

Once the protective coating was dried we did sensor calibration. For force calibration, different weights were lifted using the sensor and the corresponding voltages were measured using multimeter. Figure B-7 and Figure B-8 shows the plot of force and torque calibration. The accuracy of force sensor is about 4 N and that of torque is about 0.2 Nm.

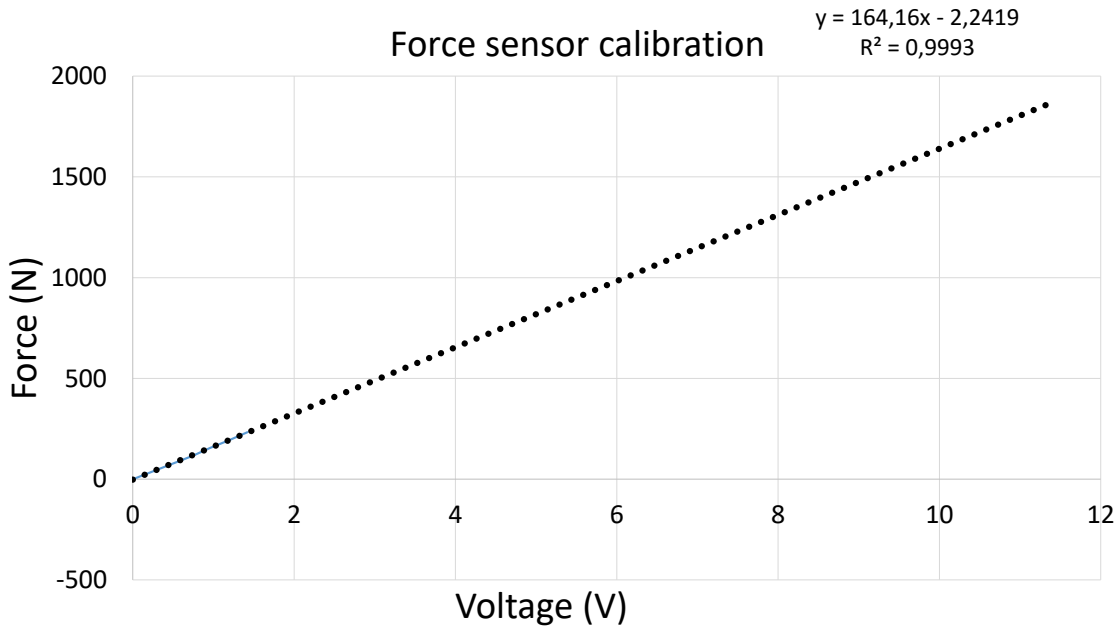


Figure B-7: Force sensor calibration

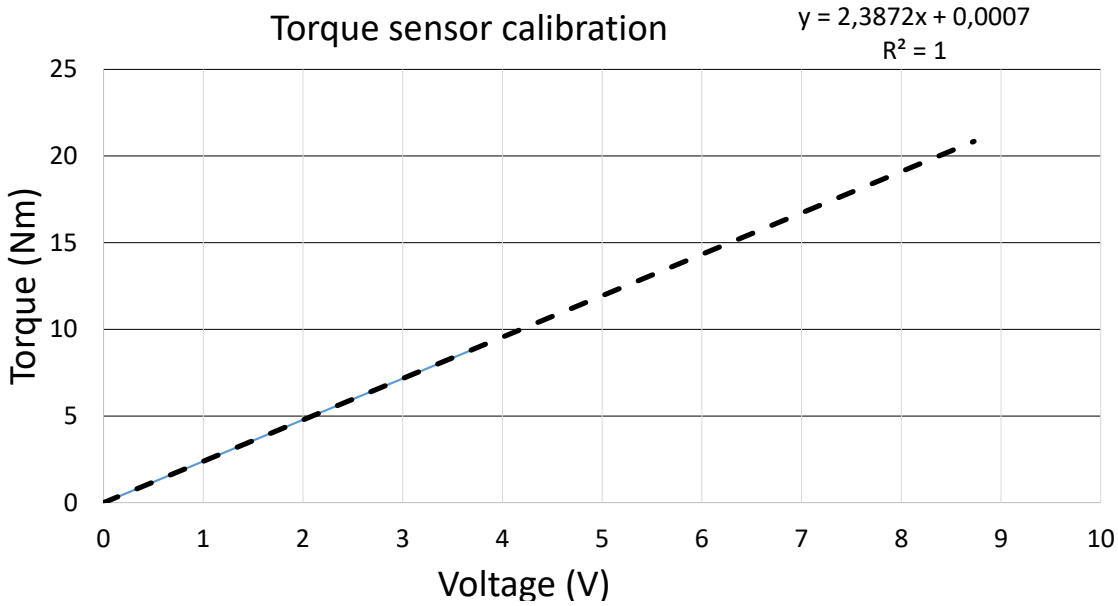


Figure B-8: Torque sensor calibration

B-7 Positional Controller

To test the performance of developed HFR positional controllers were implemented on the setup. Proportional-Integral-Derivative (PID) controller with anti-windup was used as the

positional controller. Matlab's siso tool was used to tune the PID controller by using the HFR model obtained in the Chapter 4. Figure and shows the plot of the graph of the HFR under positional controller. An accuracy of 1° and 1 mm was obtained for the rotational and translational motion respectively.

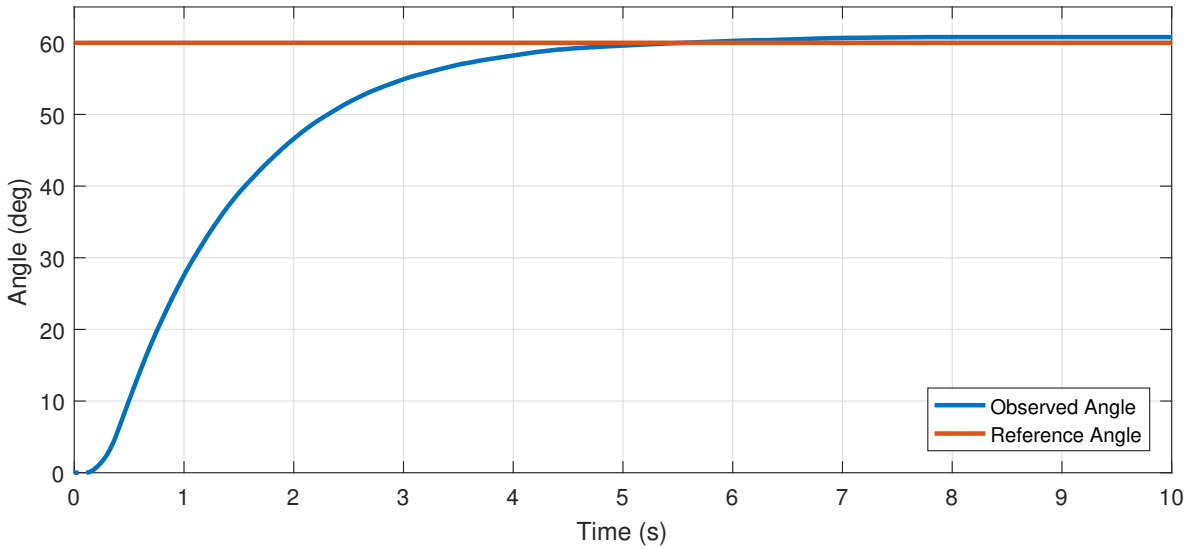


Figure B-9: Performance of HFR under PID control for linear motion.

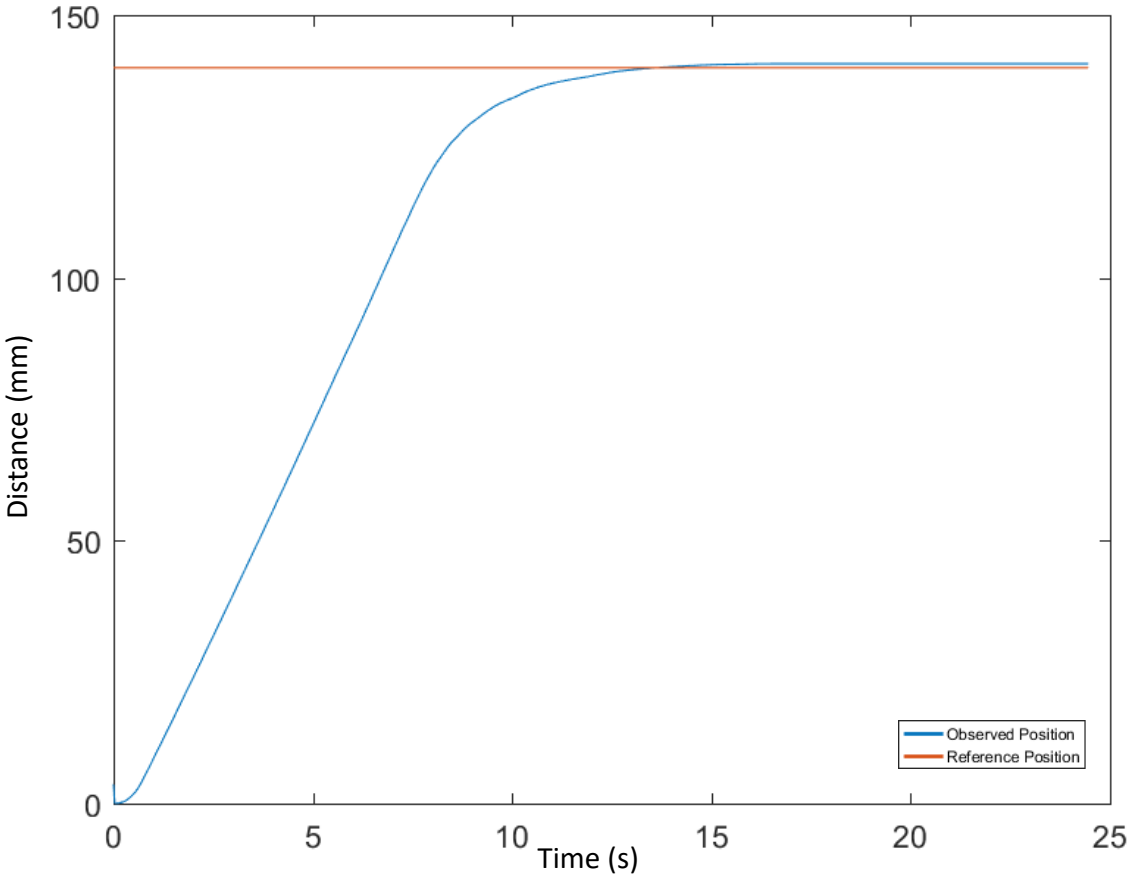


Figure B-10: Performance of HFR under PID control for linear motion.

Appendix C

Simulation

C-1 Impedance Control

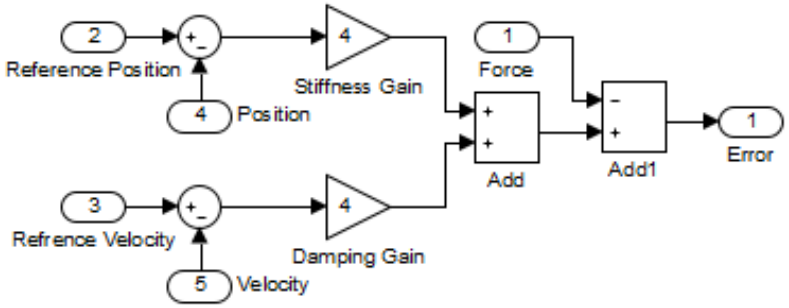


Figure C-1: Impedance controller implementation

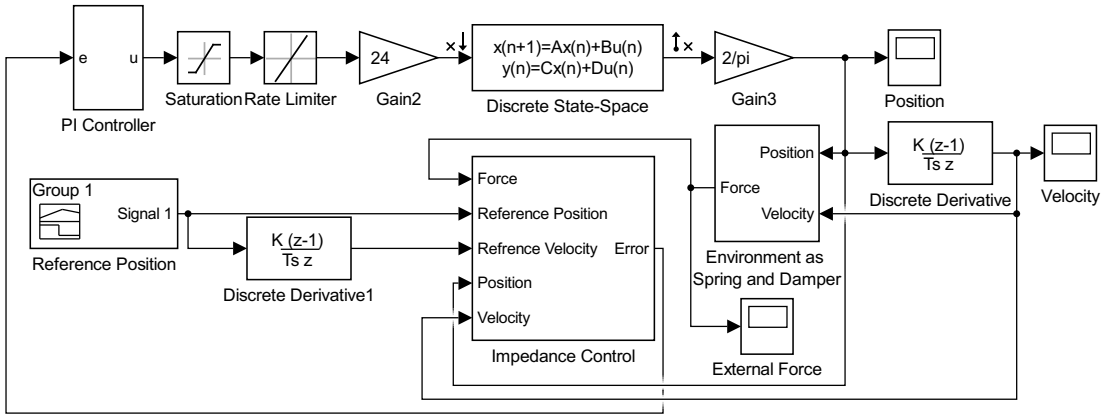


Figure C-2: Simulink model of Impedance Controller

C-2 Admittance Control

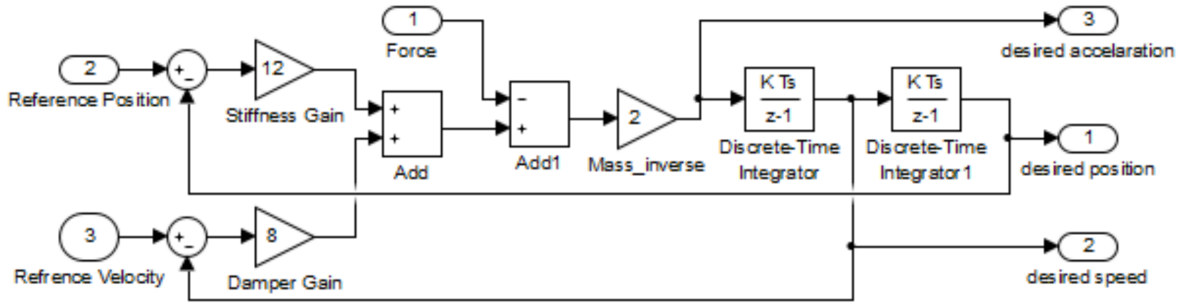


Figure C-3: Admittance controller implementation

C-3 Explicit Force Control

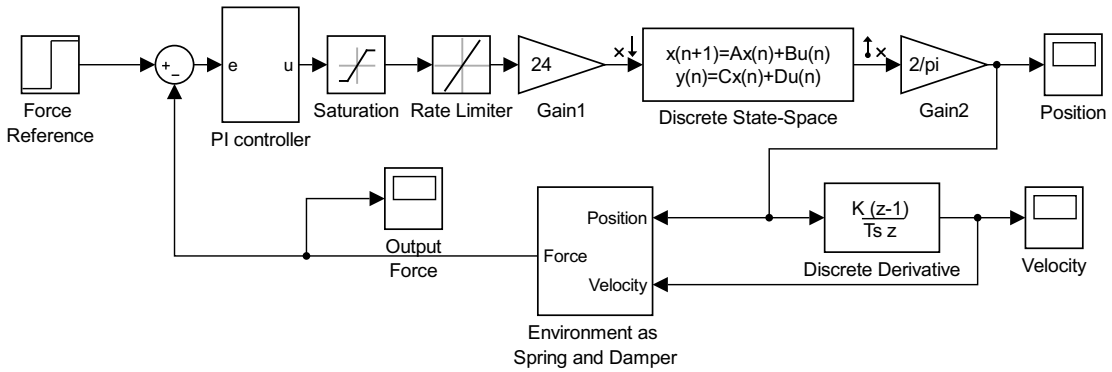


Figure C-4: Simulink model of direct force control

C-4 Environment model

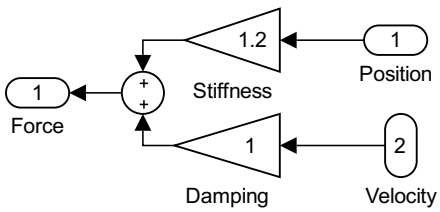


Figure C-5: Spring-damper model of patients leg and its Simulink representation

Bibliography

[1] Klaas A. Hartholt, Christian Oudshoorn, Stephanie M. Zielinski, et al. The Epidemic of Hip Fractures: Are We on the Right Track?

[2] KA. Hartholt, EF. van Beeck, S. Polinder, N. van der Velde, EMM. van Lieshout et al. Societal consequences of falls in the older population: injuries, healthcare costs and long term reduced quality of life. 10.1097/TA.0b013e3181f6f5e5, 2010.

[3] K. Bouazza-Marouf, I. Browbank, and J. Hewit. Robotic-assisted internal fixation of femoral fractures. In *Proceeding of the Institute of Mechanical Engineers* (1995), pp. 51-58

[4] S. Warisawa, T. Ishizuka, M. Mitsuishi, K. Yonenobu, N. Sugano, and T. Nakazawa. Development of a femur fracture reduction robot. In *ICRA 2004 - IEEE International Conference on Robotics and Automation*, 2004

[5] Y. Maeda, N. Sugano, M. Saito, K. Yonenobu, I. Sakuma, T. Ishizuka, S. Warisawa, T. Nakazawa, and M. Mitsuishi. Robot-assisted femoral fracture reduction: Preliminary study in patients and healthy volunteers. In *COEX Seoul, Korea "Imaging the Future Medicine"*, 2007

[6] S. Joung, H. Kamon, H. Liao, J. Iwaki, T. Nakazawa, M. Mitsuishi, Y. Nakajima, T. Koyama, N. Sugano, Y. Maeda, M. Bessho, S. Ohashi, T. Matsumoto, I. Ohnishi, and I. Sakuma. A robot assisted hip fracture reduction with a navigation system. In 11^{th} *International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part II*, 2008

[7] B. Füchtmeier, S. Egersdoerfer, R. Mai, R. Hente, D. Dragoi, G. Monkman, and N. Nerlich. Reduction of femoral shaft fractures in vitro by a new developed reduction robot system 'RepoRobo'. In *Elsevier Ltd*, 2004

[8] A E. Graham, S Q. Xie , K C. Aw, W L. Xu, and S. Mukherjee. Robotic Long Bone Fracture Reduction. In *Medical Robotics, Book edited by Vanja Bozovic, ISBN 978-3-902613-18-9, pp.526, I-Tech Education and Publishing, Vienna, Austria*, 2008.

[9] A E. Graham, S Q. Xie , K C. Aw, W L. Xu, and S. Mukherjee. Design of a Parallel Long Bone Fracture Reduction Robot with Planning Treatment Tool. In *International Conference on Intelligent Robots and Systems October 9 - 15, 2006, Beijing, China*

[10] T. Huiskens. Measurement of force and torque during fracture reduction and intramedullary fixation of pertrochanteric femoral fractures.

[11] R. Winquist, S. J. Hansen, and D. Clawson. Closed intramedullary nailing of femoral fractures. A report of five hundred and twenty cases. In *J Bone Joint Surg Am,* 1984.

[12] Peter Kazanzides. Safety Design for Medical Robots. In *31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, September 2-6,* 2009.

[13] R H. Taylor, A H. Paul, P. Kazanzides, B D. Mittelstadt, W. Hanson , J. Zuhars, B. Williamson, B. Musits, E. Glassman, and W L. Bargar. Taming the Bull: Safety in a Precise Surgical Robot. In *International Conference on Advanced Robotics (ICAR)* 1991.

[14] S S. Hung, and M Y. Lee. Functional assessment of a surgical robot for reduction of lower limb fractures. In *The International Journal of Medical Robotics and Computer Assisted Surgery* 6:413-421, 2010.

[15] H. Paul, W. Bargar, B. Mittlestadt, B. Musits, R. Taylor, P. Kazanzides, J. Zuhars, B. Williamson, and W. Hanson. Development of a surgical robot for cementless total hip arthroplasty. In *Clinical Orthopaedics and Related Research* 285:57-66, 1992.

[16] T. Gösling, R. Westphal, J. Faülstich, K. Sommer, F. Wahl, C. Krettek, and T. Hufner. Forces and Torques during Fracture Reduction: Intraoperative Measurements in the Femur. In *Wiley InterScience (www. interscience. wiley. com)*. DOI 10.1002/jor. 2004.

[17] R. Ye and Y. Chen. Development of A Six Degree of Freedom (DOF) Hybrid Robot for Femur Shaft Fracture Reduction. In *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, 2009.

[18] S. Joung, H. Liao, E . Kobayashi,M . Mitsuishi, Y .Nakajima, N. Sugano, M. Bessho, S. Ohashi, T. Matsumoto, I. Ohnishi and I. Sakuma. Hazard Analysis of Fracture-Reduction Robot and its Application to Safety Design of Fracture-Reduction Assisting Robotic System. In *2010 IEEE International Conference on Robotics and Automation*, 2010.

[19] Stryker Gamma3: Long Nail R1.5 and R2.0

[20] S. Chiavereni and L. Sciavicco. The Parallel Approach to Force/Position Control of Robotic Manipulators. In *IEEE Transaction on Robotics and Automation*, 1993.

[21] T. Douke. Control of Fracture Reduction Robot Using Force/Torque Measurement. In *30th Annual International IEEE EMBS Conference*

[22] Control Tutorials for MATLAB and Simulink (CTMS)*(http: // ctms. engin. umich. edu/ CTMS/ index. php? aux= Home)*

[23] H. Olsson, K.J. Åström, C. Canudas de Wit, M. Gäfvert and P. Lischinsky. Friction Models and Friction Compensation.

[24] R . Volpe and P . Khosla. A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators. In *IEEE Transaction on Automatic Control*, (1993) .

[25] AP. Schulz, K. Seide , Queitsch, A. von Haugwitz , J. Meiners.Results of total hip replacement using the Robodoc surgical assistant system: clinical outcome and evaluation of complications for 97 procedures. In.*The International Journal of Medical Robotics*

[26] Proximal femur, closed reduction with short intramedullary nail. https://www.aofoundation.org

[27] Matthew R. Bong, Kenneth J. Koval and and Kenneth A. Egol. The History of Intramedullary Nailing. In *Bulletin of the NYU Hospital for Joint Diseases*, 2006.

[28] Blomfeldt R, Tornkvist H, Ponzer S, Soderqvist A, Tidermark J. Internal fixation versus hemiarthroplasty for displaced fractures of the femoral neck in elderly patients with severe cognitive impairment. In *The Journal of Bone Joint Surgery British Volume*, 2005

[29] I. Sugarman, I. Adam, and T. Bunker. Radiation dosage during AO locking femoral nailing. In *Injury*, 1988

[30] da Vinci Robot Allegedly Marketed to Less-Skilled Doctors. LawyersandSettlements.com 23 April 2013. Retrieved 24 April 2013

[31] http://www.ancient.eu/Trephination/

[32] S.M. Ravishankar. Modeling of Lower Limb Muscles and Estimation of Parameters for Hip Fracture Reduction Robot.

[33] Staubli Robotics. TX200 series industrial robots. http://www.staubli.com/

[34] I. Virgala, P. Frankovsky and M. Kenderova. Friction Effect Analysis of a DC Motor. In *American Journal of Mechanical Engineering*, 2013.

[35] A. Weightman, N. Preston, R. Holt, M. Allsop, M. Levesley, B. Bhakta. Engaging children in healthcare technology design: developing rehabilitation technology for children with cerebral palsy. In *Journal of Engineering Design, 21(5):579-600*, 2010.

[36] Conti Synchrobelt HTD synchronoud drive belt. http://nl.rs-online.com/webdocs

[37] G.A. Pratt and M.M. Williamson. Series Elastic Actuation. *MIT Artificial Research Laboratory*

[38] Y. Ningbo, K. Wang, X. Chag ana L. Jingtai. A Haptic Shared Control Algorithm for Flexible Human Assistance to Semi-Autonomous Robots. In *International Conference on Intelligent Robots and Systems*, 2015

[39] K.H. Goodrich, P.C. Schutte, F.O. Flemisch and R. A. Williams . Application of the H-Mode, A Design and Interation Concept for Highly Automated Vehicles, to Aircraft. In *IEEE/AIAA 25th Digital Avionics Systems Conference*,2006.

[40] G. Zeng and A. Hemami. An overview of robot force control. *Robotica, Cambridge University Press*, 1997.

[41] Background and History of Surgical Robotics.http://allaboutroboticsurgery.com/roboticsurgeryhistory.html

[42] A.C. Grundstrom, C.E. Guse, and P.M. LaydeRisk. Factors for Falls and Fall-Related Injuries in Adults 85 Years of Age and Older. In*Archives of Gerontology and Geriatrics Volume 54, Issue 3*,2012.

Glossary

List of Acronyms

DCSC	Delft Center for Systems and Control
TU Delft	Delft University of Technology
DoF	Degree(s) of Freedom
PD	Position-plus-Derivative
PI	Position-plus-Integral
PID	Proportional-Integral-Derivative
DC	Direct Current
F/T	Force/Torque
OR	operating room
RPM	rotations per minute
RPS	rotations per second
F/T	Force/Torque
PWM	Pulse Width Modulation
VAF	Variance accounted for
HFR	Hip Fracture Reduction Robot
FDA	Food and Drug Administration
BMI	body mass index
SEA	*Series Elastic Actuation*
DAQ	Data acquisition
FEM	Finite Element Method