Force Control of a Unilateral Master-Slave System Using a SCARA Robot Arm

by

Mohammad Al Mashagbeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Industrial manipulators have several applications in a multitude of disciplines. The use of industrial manipulators has increased rapidly, and they are more refined in many applications due to advances such as fast response time, high precision, quick speed and a high level of performance. Most industrial manipulators are position-controlled; usually vision and force sensors are not integrated in most commercial industrial robots. Therefore, the addition of force and vision sensing mechanisms is required to successfully automate advanced tasks, and to enable robots to avoid high contact forces while working in applications that require contact with environments.

The objective of this thesis is to implement a unilateral master-slave system for medical applications. In this thesis, a Polaris Vicra optical tracking device is used to represent the master system, while a four degree of freedom (DOF) position-controlled SCARA manipulator from Epson is used to represent the slave system. The manipulator is equipped with a force-torque sensor to facilitate operation in unknown environments. In addition, MapleSim is used to find the dynamic model for the SCARA manipulator. Furthermore, MapleSim is also used to validate the control algorithm prior to implementation on the hardware.

Three force control techniques are used in this research and the robot's performance are evaluated. The control techniques are impedance control, admittance control and fuzzy logic control. The admittance and fuzzy logic controllers are applied to the proposed master-slave system while the impedance control is applied to the manipulator model, which was obtained from MapleSim.

In order to validate the presented control algorithms, several experiments and simulations were carried out. The experimental results show the ability of the presented controllers (admittance and fuzzy logic) to track the operator signal while keeping the force within the desired range. The simulation and animation of the impedance controller on the other hand, shows that the robot's performance can be evaluated through software.

Acknowledgements

I would like to take this opportunity to first and foremost thank God for being my strength and guidance in the completion of this work. I thank him for the gift of life and health. Without him I would not have had the physical ability to successfully complete this thesis. I would like to offer my special thanks to several kind people for their help and contribution in this research.

I would like to express my deep and sincere gratitude to my supervisor, Dr. Behrad Khamesee for giving me the opportunity to work on this project. His patience, guidance and effort were very necessary to keep me focused and motivated. I attribute the level of my Master's degree to his encouragement and effort and without him this thesis would not have been completed. I would like to thank my thesis readers, Dr. William Melek and Dr. James Tung, for taking their time to review and offer feedback on this thesis. I also would like to thank all the staff and faculty members in the department of Mechanical and Mechatronics engineering at University of Waterloo.

I would like to express my gratitude to my sponsor, the University of Jordan for their financial support. Without their financial support it would not have been possible for me to pursue and complete my Master's degree successfully.

I would like also to show my greatest appreciation to my colleagues and office mates at University of Waterloo for their encouragement and support throughout my studies. A special thanks to Roberto Ribeiro for reviewing my thesis and providing valuable feedback. To my friends scattered around the world, thank you very much for your thoughts, well-wishes, phone calls, messages and being there whenever I needed a friend. I am also deeply grateful to everyone helped me in this project.

Lastly, a special thanks to my parents for all the moral support and the amazing chances they have given me during my life. I would not be where I am today and who I am today without their love, care, help and support. You are the best parents in this world and I always owe my success to you. To my brothers, sisters and family, thank you so much for being so supportive during my study and pushing me towards achieving my goals. You are my inspiration, and I hope to make you all proud.

Dedication

I dedicate my dissertation work to my family and friends. A special feeling of gratitude to my loving parents, who have supported me in my life. I hope that this accomplishment makes you proud of me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Controlling a machine from a distance (Teleoperation) is a growing field of research in the last few decades. Teleoperation has several applications in a multitude of disciplines. In general, any teleoperated system consists of two sub-systems; the master and the slave. The slave system typically is an industrial manipulator that works to follow the master commands. The master-slave configuration is commonly used in a variety of applications, most of which requiring contact between the slave manipulator and the environment. Such contact necessitates a robust control of the interaction force, in light of unknown geometry and characteristics of the environment. In order to control the contact force between the slave system and the environment, the slave system (industrial manipulator) should be able to sense the contact force.

There are two kind of industrial manipulators; torque-controlled and position-controlled. The majority of the commercial industrial manipulators are programmed using a position control. In this control mode, the manipulator's end-effector follows a prescribed path which is programmed before run time. However, in many teleoperation applications such as chest surgery, the cutting path is undefined and it depends on many parameters such as the surgeon's hand movement and the shape of the patient's chest. In these cases, force and position control are required instead of controlling a predefined trajectory only. Moreover, in these applications the predefined trajectory which is dependent on the surgeon's hand movement (using the master device) is always tangential to the chest while the force is perpendicular. Thus, force and position control should be applied in unison. The main issue in using industrial manipulators in teleoperation is that force-torque measurements are not supported in the majority of the industrial manipulators. Nowadays, force-torque sensors are commercially available and they are effective at measuring forces and torques. Therefore to successfully implement the master-slave teleoperation configuration, force sensing should be integrated within industrial manipulators to interrupt the robot during high contact force conditions or events.

To successfully implement the teleoperation system, equipment such as vision sensors and optical tracking devices should also be integrated within industrial manipulators to represent the master system. After connecting the manipulator and the other systems together with a main computer workstation, the manipulator can work as unilateral master-slave teleoperation system.

Synchronization is the most important requirement in any master-slave system. Therefore, another important issue with respect to any master-slave teleoperation system is to synchronize all of the system components together and to avoid delays. Therefore, the applied force controller should modify the manipulator end-effector pose sequentially with change in force. To accomplish this objective correctly and without delay, a simple control approach should be applied to allow the robot to work in real-time.

The force control algorithms discussed in this thesis are impedance control, admittance control and fuzzy-logic control. The algorithms are applied to an EPSON SCARA (Selective Compliance Assembly Robot Arm) manipulator with four degree of freedoms (DOF's). The admittance and fuzzy-logic control are suitable for position-controlled industrial manipulators. On the other hand, the impedance control is useful for torque-controlled manipulators. Therefore the EPSON SCARA manipulator and the environment were modeled using MapleSim software and several simulations were conducted to test the impedance controller in the presence of contact force. In this thesis the position-force algorithms are applied to an EPSON manipulator to perform different tasks where the manipulator comes in contact with the environment.

1.2 Motivation

Robots are increasingly used in a multitude of applications. The motivation of this thesis is to study how to control the interaction force between industrial manipulators and working environments. The first objective is to develop a suitable control strategy for a slave Epson industrial manipulator working in contact with the environment. In addition, to successfully test the slave robot, the communication system of the master-slave teleoperated system should be implemented. Therefore, developing a robust communication system is the second goal of this project. As a result the overall system can be used in many applications such as medical and surgical robots or in different environment such as hazardous environments. Finally, the majority of force control algorithms in the field of robotics are dependent on the robot's dynamic model which is difficult to find analytically. Therefore, a part of this thesis focuses on finding the dynamic model of any robot (or any multi-body system).

1.3 Problem Statement

Most of industrial manipulators are position-controlled; usually vision and force sensors are not integrated in most commercial industrial manipulators. Therefore, adding force and vision sensing elements is required to successfully automate advanced tasks.

Master-slave teleoperation is a well-known example of the use of industrial manipulators. Several master-slave teleoperation systems have been developed for different applications (i.e. medical). Also, there are a few commercial systems available. These systems are expensive and some health centers are not able to afford them.

This thesis presents the development of a unilateral master-slave system using an industrial manipulator. The used industrial manipulator is position-controlled with no force sensing. Therefore, an optical tracking device and a force-torque sensor were integrated into a position-controlled industrial manipulator to work as a master-slave teleoperation system to address medical application needs. The operator will be able to manipulate the industrial manipulator which contacts with the environment from a distance. Since the slave manipulator will contact with the environment, force control between the slave manipulator and the environment is the main aspect in this thesis. Different force control algorithms are presented to control the interaction force. Furthermore, the communication system between the master and slave system are developed and improved.

1.4 Thesis Outline

This thesis is divided into seven chapters, structured as following: Chapter 1 presents a general overview of force control for industrial manipulators, and the applications of industrial manipulators in master-slave systems. Furthermore, this chapter also presents the motivation behind the work performed in this thesis. Chapter 2 provides a literature review for force control techniques that are used in industrial manipulators, and also for the master-slave systems that are being used in medical applications. Chapter 3 provides detailed information about the system and the different components of the integrated system. In addition, this chapter will outline the completed modeling for the manipulator and also the modeling of the contact force using MapleSim. Chapter 4 describes the communication module of the unilateral master-slave system. Chapter 5 will focus on the force control algorithms that are used. This chapter provides a brief introduction of the implemented force control techniques, also the mathematical derivation of each model. Chapter 6 will cover the experimental and simulation results of the force control algorithms. In addition, this chapter covers the experimental results of the improved unilateral master-slave system. Finally, the thesis summary and conclusions are presented in chapter 7.

Chapter 2

Literature Review

A robot is a machine that is programmed to execute several tasks quickly, and with high precision. The use of robots has been increased rapidly, and they preferable in many applications. Robots nowadays can be found in many applications. For instance, aerospace, medical, automotive, packaging, assembly lines, painting, welding and handling are a few examples of robotics applications. Robots are preferred due to several advantages, some of them are:

i. Robots are faster and more accurate than human operators.

ii. Robots can work in hazardous areas that humans can not.

iii. Robots can work at high speeds with limited down time.

The demands on robots are increasing rapidly and consequently the future of the robotics industry is promising. For example, Figure 2.1 demonstrates the world robot population between 2006 and 2008. It can be seen that the number of robots is growing steadily. Robots in general can be divided into three types:

1. Mobile robots: a mobile robot is a type of robot that moves in the environment for different tasks (usually used in space and hazardous environments). An example of this type of robot is NASA's mars curiosity rover (Figure 2.2).

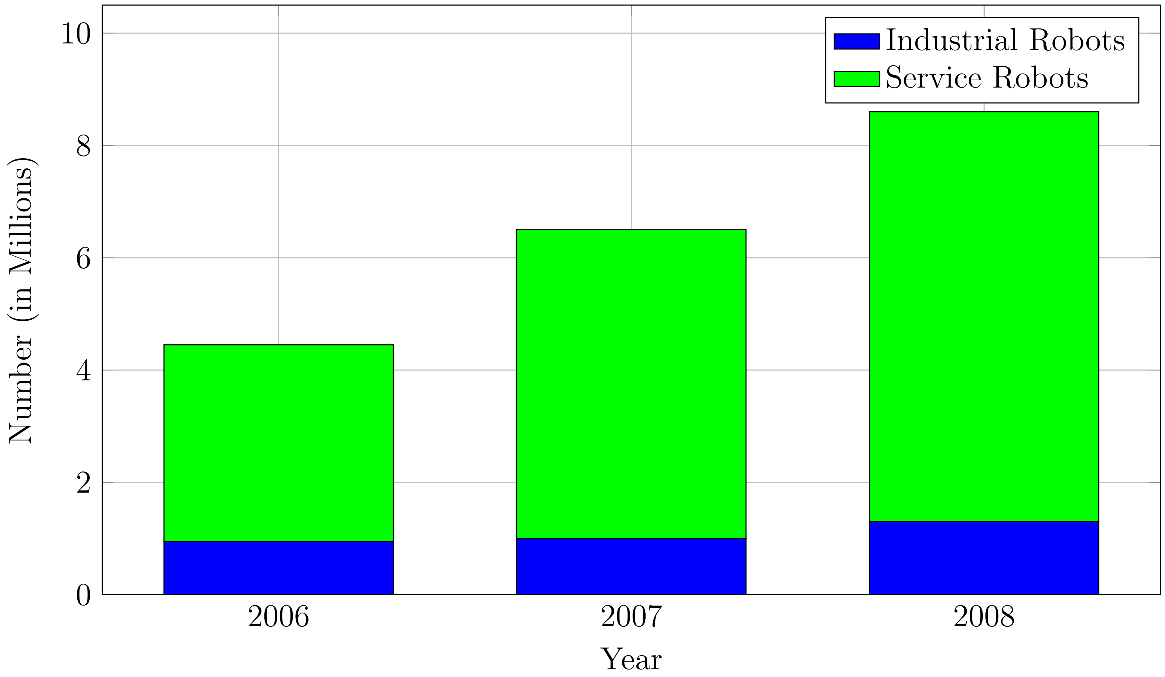


Figure 2.1: World robot population. Images reproduced from [1].

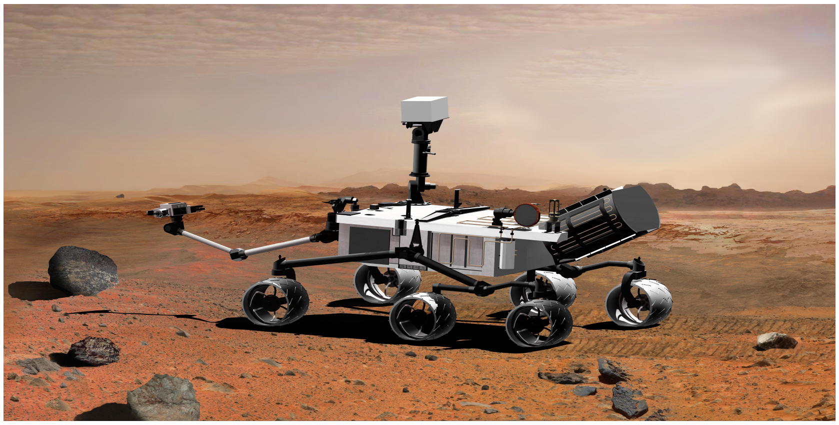


Figure 2.2: NASA's Mars Curiosity rover. Images reproduced from [2].

2. Humanoid robots: a human robot is a robot with a human body shape. It can be used for different applications from personal assistance to a space exploration. An example of this type of robot is ASIMO, which is a humanoid robot developed by Honda (Figure 2.3).

Figure 2.3: ASIMO humanoid robot. Images reproduced from [3].

3. Industrial robots (Manipulators): an industrial robot is a type of robot that contain multiple links connected by different types of joints. Industrial robots are most widely used in industry for different tasks such as: welding, material handling, painting, assembly and others. The first industrial robot was developed by Unimation in the late 1950s [4]. A famous example of an industrial robot that is widely used nowadays in many applications is the KUKA industrial manipulator (Figure 2.4).

Among the previously mentioned robots, industrial manipulators are the most common type of robots due to the variety of applications, high speeds and the ability to execute repetitive tasks with a high degree of precision. According to Figure 2.5 which represents a summary of the worldwide sales of industrial manipulators between 2004 and 2013; the demand for industrial manipulators is increasing rapidly (except for during the global Recession of 2009). When demand increases, new applications and tasks are found. Therefore, the industrial manipulator applications are getting more sophisticated. The sophisticated applications require robust control of the manipulator. Hence, controlling the manipulator is the most vital aspect of the manipulator in the execution of complex tasks successfully.

In general, the functionality of any robot is to perform some predefined tasks. Recently, due to the wide variety of robot tasks, robots should adapt to any working environment and that is why force control has been studied in the last few decades. The majority of

Figure 2.4: KUKA industrial manipulator. Images reproduced from [5].

industrial manipulators are not equipped with force sensing devices, but in the last few years, many robot manufacturers have already begun to integrate force sensors into their robots. The benefit of using force control rather than position control alone is that robots gain the ability to avoid high contact forces in applications that require contact within the environment. Furthermore, by knowing the force acting on the end effector, robot performance can be improved.

In case of the position control alone, the manipulator will follow a prescribed or programmed trajectory. Such applications include pick and place, thermal spray and assembly lines. In other words, it can be called offline programing, since the manipulator tries to follow the given trajectory without taking the force and the environment into consideration. Force-position control, on the other hand, is more beneficial. Many applications such as welding, painting and peg-in-hole assembly require the robot to come in contact with environment and keep the interaction force under a required limit to avoid damage to the robot or environment. This can be called online programming; as the manipulator attempts to not only follow the predefined input trajectory, but also adjust to maintain the contact force.

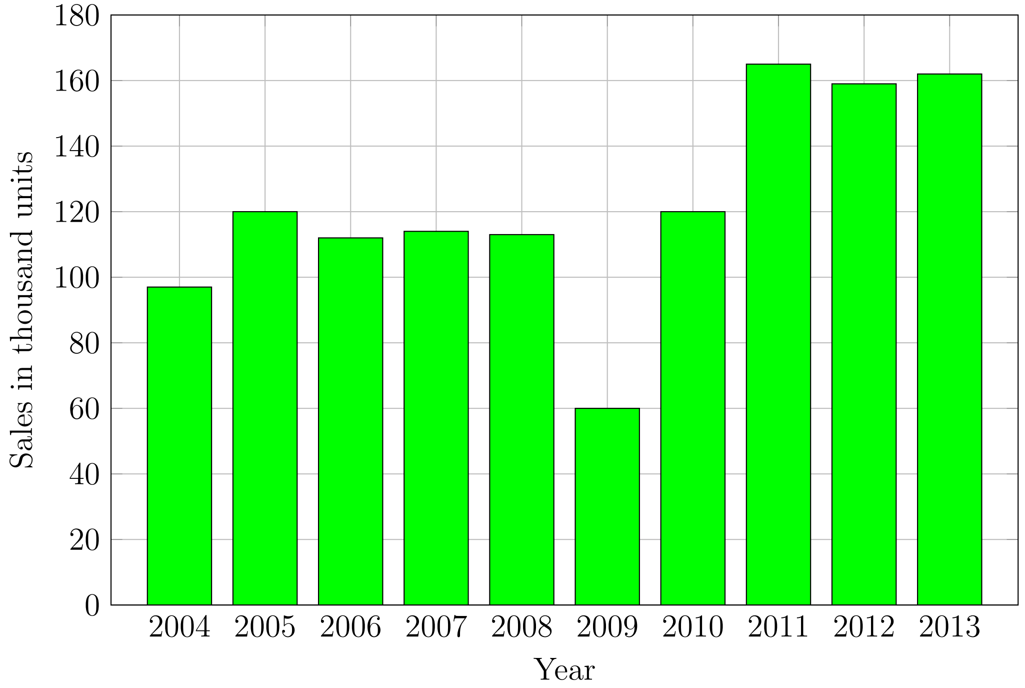


Figure 2.5: Summary of worldwide sales of industrial robots between 2004 and 2013. Images reproduced from [6].

Nowadays, force control is an essential part of successfully automating several manufacturing processes. Furthermore, many robot manufacturers are trying to integrate their products with force sensing devices. This will give robot operators the ability to easily program their robots in the presence of contact force. As mentioned before, force control is widely discussed in the last three decades in many books and journal publications. It is not a new science but it is still a growing field and many new methods are being developed and presented in literature. In this thesis, force control will be discussed specifically for position-controlled industrial manipulators.

2.1 Force Control for Industrial Manipulators

Solving real-time force control problems is still a growing field. Many of the force control techniques are complex and robot dynamic should be fully modeled and understood to apply these advanced approaches. The environment characteristics also need to be understood to move a robot with a constant contact force. Robot force control techniques are classified into two groups; fundamental and advanced force control techniques [7]. The fundamental algorithms are based on the relation between the position, velocity or acceleration and the applied force. Techniques such as impedance control, hybrid force-position control and stiffness control are categorized as fundamental algorithms. The advanced force control techniques such as adaptive control, fuzzy logic control and learning control are usually used in the case of unknown environmental conditions. Therefore, selection of the controller is depending on many variables such as environment, robot dynamics, the given task, overall accuracy and controller robustness.

Force control methods allow the robot to move and respond to any contact with the environment. This robot movement is divided into two categories; fine and gross motion. In gross motion, reaching the robots final destination is the main objective. On the other hand, the robot precisely positions itself to avoid obstacles and reach the final target in the fine motion case [8]. The relative position between the robot and the environment can be adjusted in the fine motion case to avoid non desired contact. Therefore, fine motion is usually slower than the gross motion.

As mentioned before, many force control techniques have been introduced. For example, in 1972 Groome [9] used continuous force feedback to adjust the contact force and maintain it under the required amount. Whitney [8], also used the force feedback to adjust the path while maintaining contact. In the passive device method that was introduced by Watson in 1976, a spring and a damper were connected to the robot end-effector to adjust the contact force [10]. Stiffness control on the other hand uses active force sensors such as strain gauges that is embedded in robot actuator or in the contact point to correct the force error [11]. In explicit force control, the input force is used to control instead of position, velocity or acceleration [12]. Hogan [13] modeled the environment as mechanical impedance that contains a virtual mass, spring and damper. The force feedback is used then with the impedance controller to adjust the robot end-effector position and contact force. Using this model, the end-effector position depends not only on a predefined input, but also on the dynamic characteristics of the contact force [14]. Another popular technique is called hybrid force-position control. In this control method that was firstly introduced by Raibert and Craig in 1981, some directions are position-controlled while the other are force controlled [15]. The selection matrix is used to select which directions are position or force controlled. This method is hard to implement in position-controlled robots due to the robots internal controller. The advantage of this control technique is that both force and position are controlled separately. However, the system stability should be studied since some works found that the hybrid force-position control might be unstable [16].

In recent years new force control techniques have been introduced. Most of them are based on the previously mentioned techniques such as impedance control and hybrid force-position. For example, a synthesis method of force/position control systems without using force-torque sensors [17]. This method however depends on the dynamic model of the robot. Therefore, an accurate model for the given robot is needed otherwise the method might not be applicable. Another example that uses impedance control is presented in [18]. In this method a position based impedance controller with a torque observer is used to control the interaction force.

Many of the presented methods are used successfully to control interaction force. However, many of them are not applicable for position-controlled industrial manipulators. The position-controlled industrial manipulators have closed controllers and accessing each joint variable is not possible. Due to this limitation of position control, many of the advanced force control techniques [15] are not applicable without any modification to the controller.

The majority of the previous discussed methods are difficult to apply to position-controlled industrial manipulators due to three core reasons:

1. In position-controlled industrial manipulators, it is difficult to access the joint torques. Therefore, torque feedback is not applicable. Referring to [19], it is possible to estimate joint torques of a robot by knowing the forces and torques at the end-effector. However, this method is dependent on many parameters such as the dynamics of the robot and the location of the contact force. In addition, due to added complexity of deriving joint torques, additional computation power is necessary and further latency may occur. In our system, even by estimating joint torques, it is not possible to apply a torque to each joint due to the limitations of a position-controlled robot.

2. The output parameter in the majority of the discussed force techniques is torque while the input to the position-controlled industrial manipulator is a position.

3. The majority of the discussed force techniques depend on robot dynamic model. Therefore, any inaccurate modeling might result in failure during the control process.

Based on these facts, only the position based force control techniques are practically applicable in these types of robots. The admittance and the fuzzy logic controllers are an example of the position based force control techniques.

2.2 Master-slave Configuration

Typical applications for force control are found in master-slave system configurations. These configurations are widely used especially in medical applications. Typically, surgical robots consist of two main systems: master and slave. The slave system typically is a manipulator that works on the patient side. The master system is located on the doctor or surgeon side. It can be another manipulator, haptic device or a vision system. The slave is controlled to follow the master's position, force or velocity commands. Another interesting aspect of this technology is that the surgeon and patient can interact remotely (this is called teleoperated surgery). The first master-slave teleoperated system was developed by Raymond Goertz in the late 1940s [20]. Over time, teleoperation has been improved significantly and become a vital part of many medical and surgical applications [21, 22]. Recently, many advanced teleoperation devices have been introduced such as the diVinciTM [23] (Figure 2.6) and ZEUS robotic surgical system [24] . However, there are two types of teleoperation: unilateral and bilateral. In the former, the slave follows the master without a force feedback. Therefore, the master's position is used only for feedback. In the latter, the force and position are fed back from the slave system to the master system. This thesis introduces a unilateral master-slave system implemented for use in medical applications. To successfully automate the master-slave system in real-time, position and force should be controlled to avoid undesired behavior of the system. The main focus of the current work is to introduce the communication and system integration. Practically, the time delay is a significant factor in the success of any master-slave system. Therefore, most of the current work is focused on reducing this parameter.

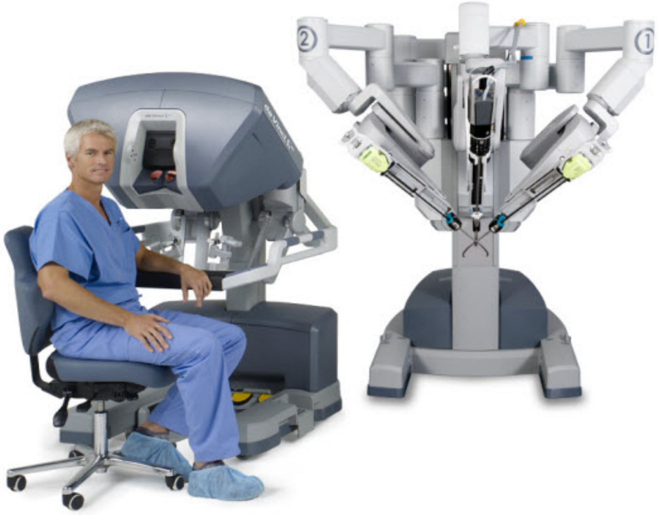


Figure 2.6: The daVinci surgical system. Image reproduced from [25].

As previously mentioned, a unilateral master-slave system using an industrial manipulator will be implemented. In the next chapter, the components of the master slave system are presented.

Chapter 3

System Overview and Modeling

In this chapter, an overview of the components and integration of the unilateral master-slave system are presented. Moreover, the modeling techniques employed to model the manipulator forward kinematics as well as dynamics. MapleSim modeling is also presented for the contact force and SCARA manipulator.

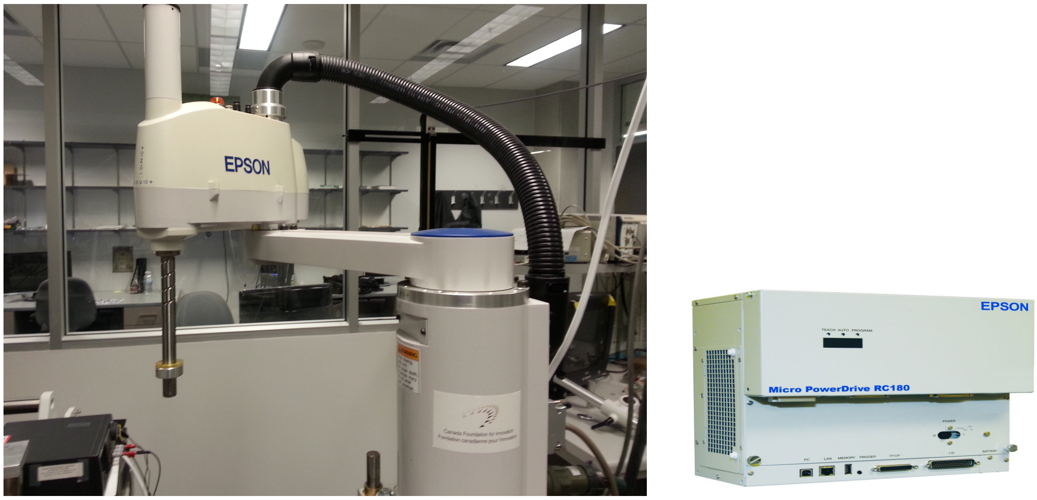
3.1 System Overview

In this section, the components of the unilateral master-slave system are explained.

3.1.1 Epson SCARA Manipulator

The manipulator is based on a four DOF SCARA that can rotate around and move along the z-axis. SCARA robots are widely used, especially in industrial and bio-mechanical applications. Figure 3.1a shows the actual SCARA manipulator (EPSON SCARA E2S651S-UL) which is manufactured by Epson. The manipulator consists of three links: L_1, L_2 and L_3 that are 415 mm, 235 mm and 170 mm long respectively. Links L_1, L_2 and the end-effector rotate around the z-axis, while the link L_3 moves along the z-axis.

The manipulator is equipped with an RC-180 controller shown in Figure 3.1b. The manipulator's controller communicates with the Epson RC+ 5 software installed on the main workstation via a USB port. The robot specifications are summarized in Table 3.1.



(a) The manipulator.

(b) Epson RC-180 controller. Images reproduced from [26].

Figure 3.1: The SCARA Epson manipulator.

Table 3.1: Summary of the Epson E2S651S-UL robot specifications.

Weight	$20\,kg$	
Driving method	All joints	AC servo motor
Max. operating speed	Joint #1, #2	$6300\,mm/s$
	Joint #3	$1100\,mm/s$
	Joint #4	$1870\,degrees/s$
Repeatability	Joint #1, #2	$0.015\,mm$
	Joint #3	$0.010\,mm$
	Joint #4	$0.02\,degrees$
Resolution	Joint #1	$0.0010986\,degrees/pulse$
	Joint #2	$0.0017578\,degrees/pulse$
	Joint #3	$0.0032552\,mm/pulse$
	Joint #4	$0.005493\,degrees/pulse$

3.1.2 Force Sensor

The design of a force controller is necessary to control the interaction force. However, without a device that measures the force in real time, the controller is useless. Thus, force measurement is the first step in controlling the whole process. There are two types of external force sensors that are used within the robotic field: contact and contact-less. An optical force sensor is an example of a non-contact case. On the other hand from its name, contact sensors need to be in contact at a point to measure force. Strain gauges are a well-known mechanism used in this category of contact sensors.

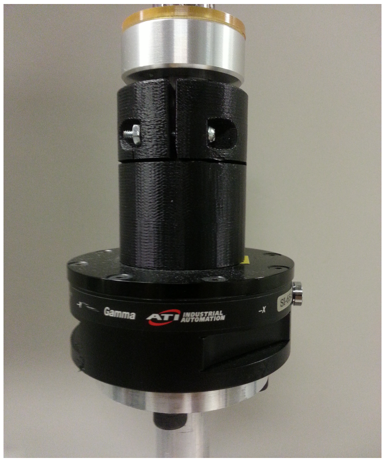


Figure 3.2: ATI force-torque sensor.

In our system, the ATI Multi-Axis Gamma force-torque sensor is used to measure forces and corresponding torques (Figure 3.2). The sensor consists of the following components:

1. Transducer
2. Transducer shielded cable
3. Intelligent data acquisition system
4. F/T controller box

This 6 DOF force-torque sensor measures all of the contact forces and torques. After that, it sends the measurements to the ATI F/T controller box to convert it to an analog output voltage. The data acquisition system then converts the output to a digital numeric value using a 9600 baud rate. This sensor is capable of measuring applied forces with a resolution of up to $1/40\,N$ in the F_x and F_y directions and a resolution up to $1/20\,N$ in the F_z direction. The specifications for the Gamma model are shown in Table 3.2.

Table 3.2: Gamma force-torque sensor specifications.

Single-Axis Overload	
F_{xy}	$1200\,N$
F_z	$4100\,N$
T_{xy}	$79\,Nm$
T_z	$82\,Nm$
Physical Specifications	
Weight	$0.255\,kg$
Diameter	$75.4\,mm$
Height	$33.3\,mm$

Forces in the x, y, and z directions as well as the corresponding torques are measured and then sent to a workstation computer using a RC232 port. Based on the force readings at each time step k, the controller calculates the corresponding increment of position and sends it to the manipulator. The sensor is mounted on the robot wrist as shown in Figure 3.3

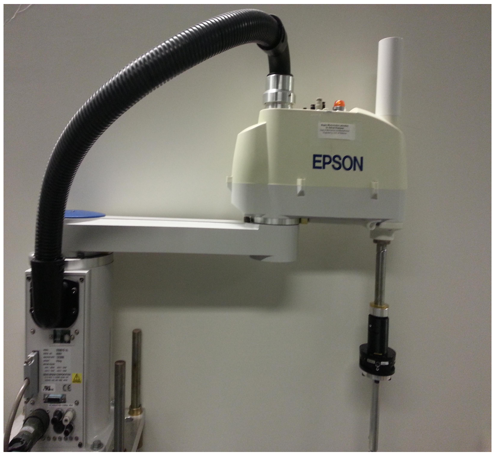
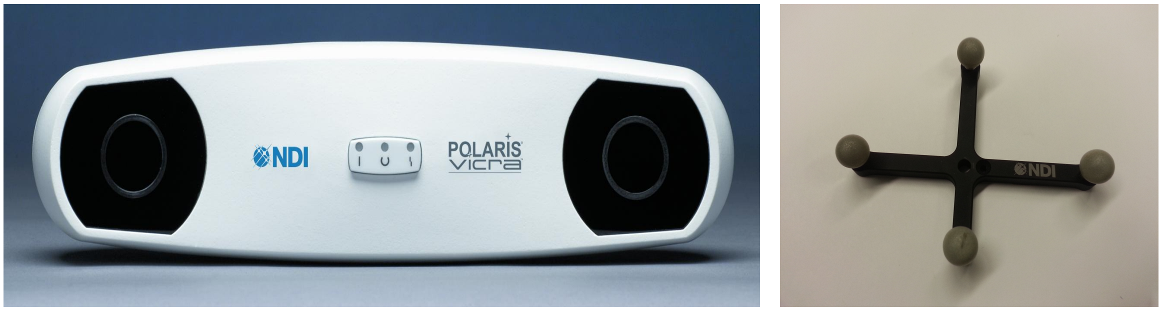


Figure 3.3: Force sensor mounted on the manipulator wrist.

3.1.3 Tracking Device

The tracking system is a Polaris Vicra optical tracking device from Northern Digital Inc. The Polaris Vicra optical tracking device in Figure 3.4 is used to measure the 3-D position of a rigid body consisting of four retro reflective passive markers. The Polaris Vicra optical tracking device consists of the following four components:

1. Position Sensor (Figure 3.4a)
2. USB Converter
3. Power adapter
4. Spherical passive markers (Figure 3.4b)



(a) The Polaris Vicra position sensor. Images reproduced from [27].

(b) A rigid body tool with four retro reective markers.

Figure 3.4: Polaris Vicra optical tracking device.

The maximum update rate of the optical tracking device is $20\,Hz$ using the Application Program Interface (API) and the measurement accuracy is up to $0.3\,mm\ RMS$. The Polaris Vicra communicates with the workstation PC using an RS232 port. The Polaris tracking device is mounted in an overhead position to ensure it maintains ideal conditions to keep the rigid body in its line of sight. This device works by emitting an infrared beam (IR) within its working volume. The emitted light then is received by a capacitive discharge device (CCD) within the unit which is reflected from the passive markers on the rigid body, refer to Figure 3.5.

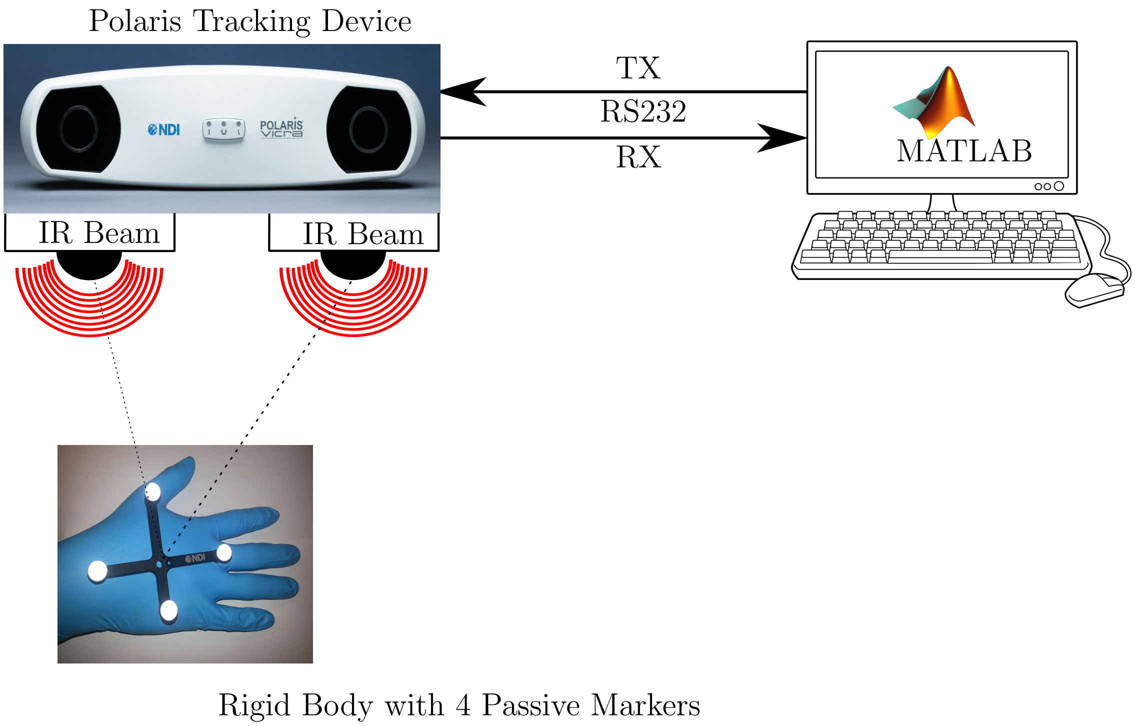


Figure 3.5: Polaris Vicra optical tracking device setup.

The Polaris Vicra System is able to track the rigid body tool within a specific measurement volume. A 3D representation of the measurement volume is shown in Figure 3.6.

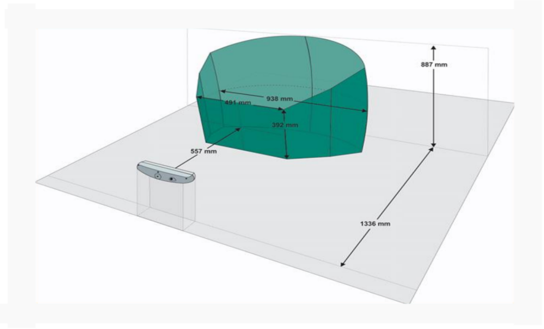


Figure 3.6: The measurement volume of the Polaris Vicra system is illustrated by the dark shaded area. Source: Polaris Vicra User Guide [28].

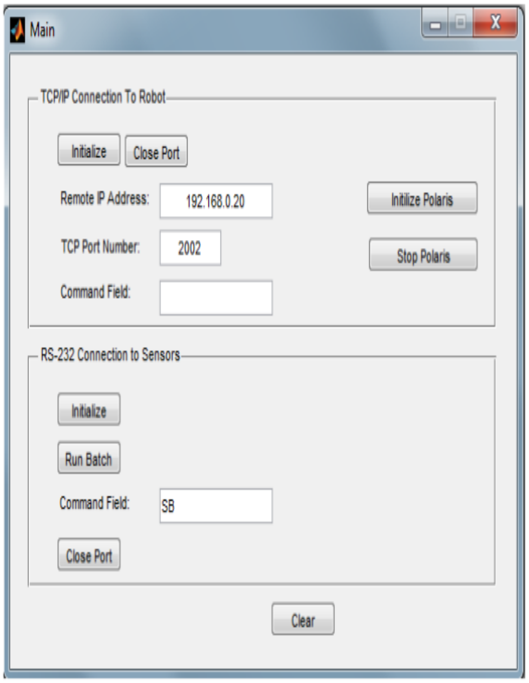
3.1.4 Graphical User Interface

The essential objective of the developed graphical user interface (GUI) is to:

1. Connect the force-torque sensor and the optical tracking device to the main workstation and continuously read and display values at the workstation.

2. Connect the main workstation to the Epson RC+ 5 software using a TCP/IP connection port.

As discussed before the force-torque sensor and the optical tracking device are connected to the main workstation using a RS232 port. The GUI allows the operator to send and receive information using the RS232 ports which are connected to the both systems. The GUI was developed using MATLAB (Figure 3.7).



Remote IP Address: 192.168.0.20

TCP Port Number: 2002

Command Field:

Command Field: SB

Figure 3.7: Graphical User Interface of the communication system.

3.1.5 Manipulator Interface

As previously mentioned, the manipulator's controller communicates with the Epson RC+ 5 software which is installed on the main workstation using a USB port. This USB port allows the manipulator's operator to send different kinds of commands to the manipulator. It also allows the operator to read the robot's parameters such as position and joint angles. However since the force-torque sensor and the tracking device are working together, there is a need to establish a TCP/IP connection to continuously feed the RC+ 5 software with the position from the tracking system and the measured contact force from the force sensor.

3.1.6 PC Workstation

MATLAB, a commercial product of The MathWorks, Inc., is installed on the main PC workstation to control the entire process. The main workstation connects with the different systems using a variety of connectors. After getting all of the required parameters from the different systems, the controller output is fed to the manipulator controller (Figure 3.8).

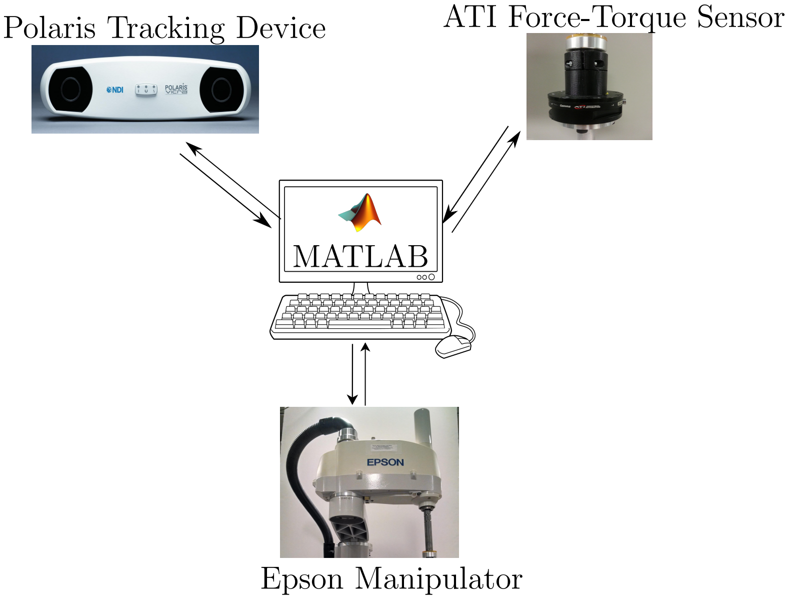


Figure 3.8: Master-slave system overview.

3.2 SCARA Robot Kinematic

Forward kinematics deals with finding the position and orientation of the robot end-effector as a function of its joint angles. In robots, forward kinematics can be found using several methods; such as D-H convention or the graphical method. In this section, the graphical method is used due to the shape of the SCARA robot. The position of the End-Effector is $q = [X_q, Y_q, Z_q, \theta_q]$, while the joint variables are $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]$. Figure 3.9 depicts the relation between the end effector position and joint angles.

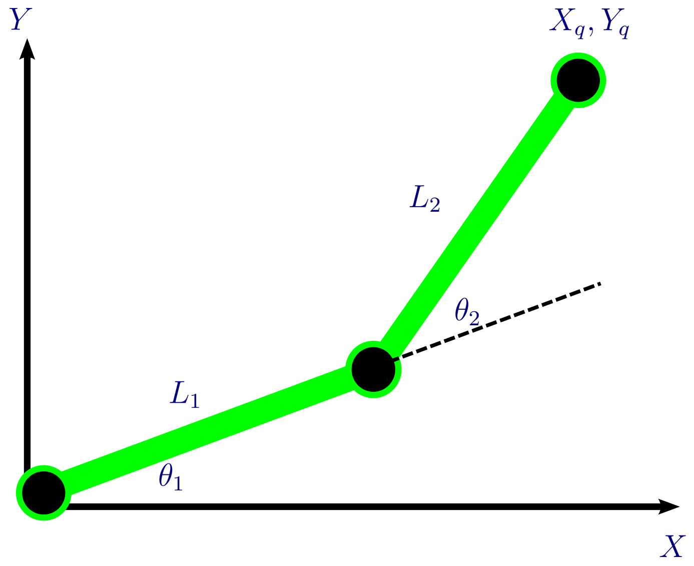


Figure 3.9: Robot forward kinematic.

Based on Figure 3.9, the robot's forward kinematic equations can be found as follows:

$$X_q = L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) \tag{3.1}$$

$$Y_q = L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2) \tag{3.2}$$

Since the prismatic joint moves only in the negative Z direction, Z_q depends only on joint angle θ_3.

$$Z_q = -\theta_3 \tag{3.3}$$

The total rotational angle around the Z is the summation of the three joint angles θ_1, θ_2 and θ_4. So,

$$\theta_q = [\theta_1 + \theta_2 + \theta_4] \tag{3.4}$$

3.3 SCARA Robot DH Parameters

The Denavit-Hartenberg (D-H) parameters for the SCARA robot are defined in Table 3.3.

Table 3.3: DH parameters for the SCARA robot.

i	θ_i	d_i	a_i	α_i
1	θ_1	0	L_1	0
2	θ_2	0	L_2	0
3	0	θ_3	0	0
4	θ_4	0	0	0

where:

a_i: link length, distance between the z_i and z_{i+1} (projected along z_i).

α_i: link twist, angle between z_i and z_{i+1} (measured around z_i).

d_i: link offset, distance between z_{i-1} and z_i (projected along z_i).

θ_i: joint angle, angle between x_{i-1} and x_i (measured around z_i).

3.4 Jacobian

The Jacobean defines the transformation between the robot end-effector velocity and the joint velocity. In previous sections, we found the relation between the position and orientation of the robot end-effector with respect to the joint angles; therefore, we can differentiate the position with respect to time to get the Jacobian matrix. By differentiating equations (3.1), (3.2), (3.3) and (3.4) we yield:

$$\dot{X}_q = -[L_1 sin(\theta_1) + L_2 sin(\theta_1 + \theta_2)]\dot{\theta}_1 - L_2 sin(\theta_1 + \theta_2)\dot{\theta}_2 \tag{3.5}$$

$$\dot{Y}_q = \ \ [L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2)]\dot{\theta}_1 + L_2 cos(\theta_1 + \theta_2)\dot{\theta}_2 \tag{3.6}$$

$$\dot{Z}_q = -\dot{\theta}_3 \tag{3.7}$$

$$\dot{\theta}_q = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_4 \tag{3.8}$$

If we rewrite the equations above using matrix notation we yield the following:

$$\begin{bmatrix} \dot{X}_q \\ \dot{Y}_q \\ \dot{Z}_q \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2) & -L_2 sin(\theta_1 + \theta_2) & 0 & 0 \\ L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) & L_2 cos(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{3.9}$$

Fundamentally $\dot{q} = J_q(\theta)\dot{\theta}$; therefore based on this relationship the Jacobian matrix for the SCARA robot is:

$$J_q(\theta) = \begin{bmatrix} -L_1 sin(\theta_1) - L_2 sin(\theta_1 + \theta_2) & -L_2 sin(\theta_1 + \theta_2) & 0 & 0 \\ L_1 cos(\theta_1) + L_2 cos(\theta_1 + \theta_2) & L_2 cos(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{3.10}$$

3.5 SCARA Robot Dynamics

For serial manipulators, the general form of the dynamic equations of motion follows the following form [29]:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{3.11}$$

where $M(\theta)$ is a symmetric positive definite 4×4 mass matrix, $C(\theta, \dot{\theta})$ is a 4×4 matrix of Coriolis and centrifugal terms, $G(\theta)$ is a 4×1 gravitational vector, and τ is a 4×1 vector of generalized torques. To find the dynamic equations for the SCARA robot, the

Euler-Lagragian method was used to obtain all of the dynamic equations in symbolic form. The links' kinetic, rotational and potential energies were found by calculating:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \dot{\theta}_i} = Q_i \tag{3.12}$$

for each link i in Figure 3.10

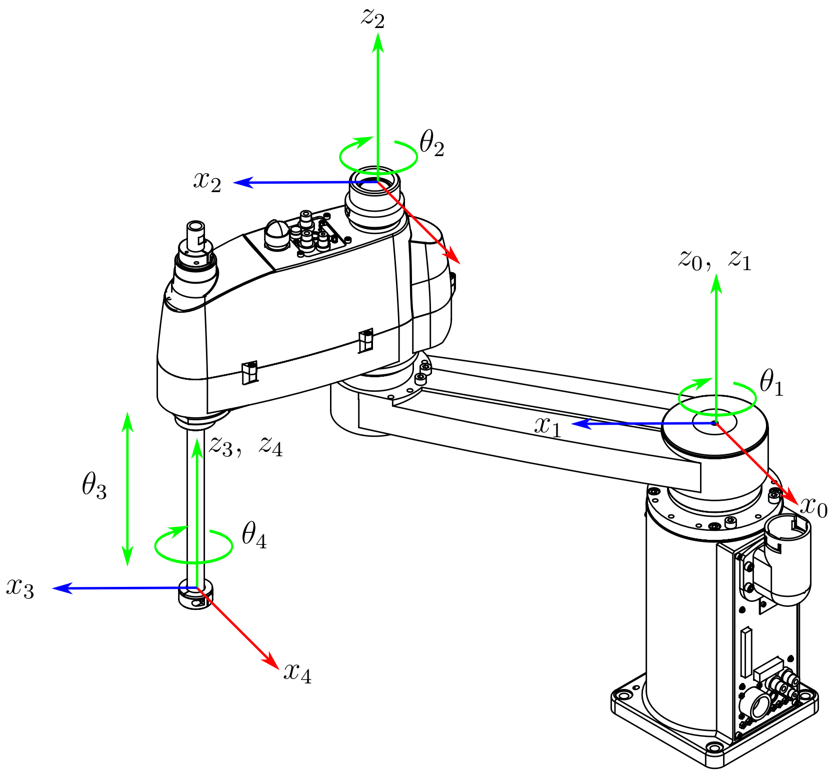


Figure 3.10: 3D-Model of Epson SCARA robot showing the coordinate frames.

These equations are summarized in a simple matrix form as follows:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} p_1 + p_2c_2 & p_3 + 0.5p_2c_2 & 0 & -p_5 \\ p_3 + 0.5p_2c_2 & p_3 & 0 & -p_5 \\ 0 & 0 & p_4 & 0 \\ -p_5 & -p_5 & 0 & -p_5 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} -p_2s_2\dot{\theta}_2 & -0.5p_2s_2\dot{\theta}_1 & 0 & 0 \\ 0.5p_2s_2\dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -p_4g \\ 0 \end{bmatrix} \tag{3.13}$$

s_2 and c_2 represent $sin(\theta_2)$ and $cos(\theta_2)$ respectively, and

$$p_1 = \sum_{i=1}^{4} I_i + m_1x_1^2 + m_2(x_1^2 + a_1^2) + (m_3 + m_4)(a_1^2 + a_2^2) \tag{3.14}$$

$$p_2 = 2a_1x_2m_2 + 2a_1a_2(m_3 + m_4) \tag{3.15}$$

$$p_3 = \sum_{i=2}^{4} I_i + m_2x_2^2 + (m_3 + m_4)a_2^2 \tag{3.16}$$

$$p_4 = m_3 + m_4 \tag{3.17}$$

$$p_5 = I_4 \tag{3.18}$$

τ_i is the input torque (or force), I_i is the moment of inertia around the centroid, m_i is the mass of link i, x_i is the center of mass from the centroid, and a_i is the length of link i.

3.6 Manipulator Modeling Using MapleSim

To precisely control a serial manipulator, the robot dynamics have to be modeled accurately. Many methods were used to find an accurate dynamic model for the SCARA industrial manipulator. The Euler-Lagrange method [30, 31] was used to determine the SCARA dynamic model. This method depends on many parameters such as inertia, mass and the center of mass location for each link. Therefore incorrect measurement of any of these parameters leads to inaccurate modeling. Wildner et al. [32] used the neural networks to estimate the dynamic model of the SCARA manipulator; however additional algorithms should be added here to eliminate the linearization error and to estimate the correct model. Gautier [33], on the other hand, used the linear inverse kinematic model to let the robot follow a prescribed trajectory to estimate the dynamic parameters. This method solves the parameters numerically with the requirement of updating the control gains in each iteration. Yet, it is hard to obtain the dynamic model of any robot as a closed block that gives the symbolic dynamic relation between the input torque and the output joint's position. Nowadays with the high speed computers and the quick revolution of multi-body simulation software, the dynamic equations of motion can be found in seconds by setting the relation between each two links. Although these software packages require the user to input the links parameters such as inertia, mass and location of center of mass, these parameters can be found effortlessly by using computer-aided design (CAD) software such as Soildworks.

MapleSim is one of the advanced system level modeling software packages that offers the capability to model any multidomain system. Moreover, by using the Maple software in parallel with MapleSim, the full symbolic equations can be derived as a block that precisely describes the relation between specified input and output variables. This block can be used in MapleSim to control the overall system. It also can be exported to any other control simulation software packages such as MATLAB, in order to test the model with different controllers and environments.

In this section, a method for finding a dynamic model of any robot is proposed. As an example, Epson SCARA industrial manipulator dynamic model will be derived using the proposed method. The first step in finding the dynamic model is to obtain the manipulator CAD data. In this work Soildworks is used to produce the CAD model of the SCARA robot. The CAD data in Figure 3.12 will be used to find manipulator links parameters such as inertia and mass. Furthermore, it will be exported later to MapleSim to obtain the 3D multi-body model and to perform simulations.

After obtaining the CAD data, it is imported into MapleSim. MapleSim software is a powerful tool for developing a symbolic dynamic model for multi-body systems. The

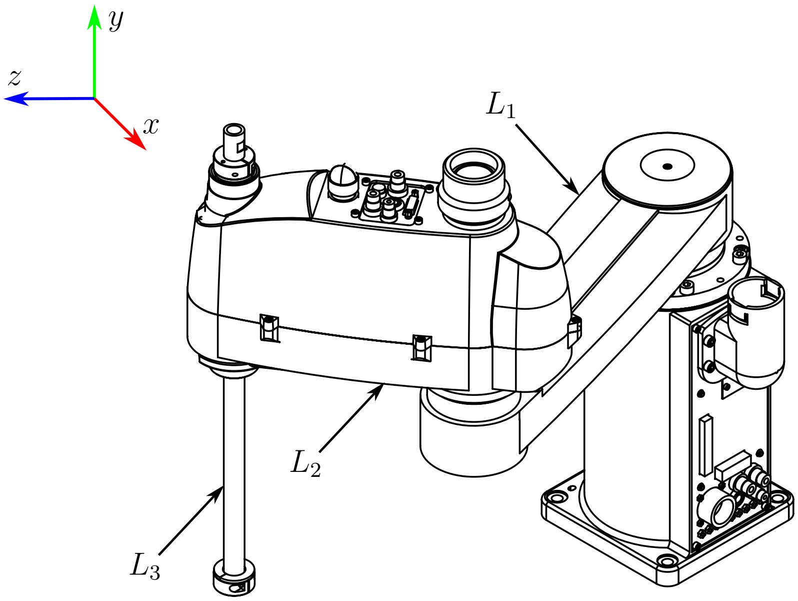


Figure 3.11: Robot CAD model in Soildworks. Images reproduced from [26].

multi-body modeling-simulation library in MapleSim was used to create the model. Within MapleSim users are able to create complex multi-body physical systems that may include bodies, joints, forces, sensors, and other components. The SCARA MapleSim model, shown in Figure 3.12, contains many subsystems. Each subsystem represents one link. The subsystems are identical with different parameters. For example one of the subsystems, shown in Figure 3.12, contains a body, rigid body frame and CAD geometry which is imported from Soildworks as a .STL file.

After running the MapleSim software, MapleSim engine compiles the system to obtain the model. Maple software was used then to find the symbolic dynamic model $G(s)$ shown in Figure 3.13.

3.7 Contact Force Modeling

The Hunt-Crossley contact model [34] is used to model the normal external interaction force between the end-effector and the environment in the Y direction. Using this method,

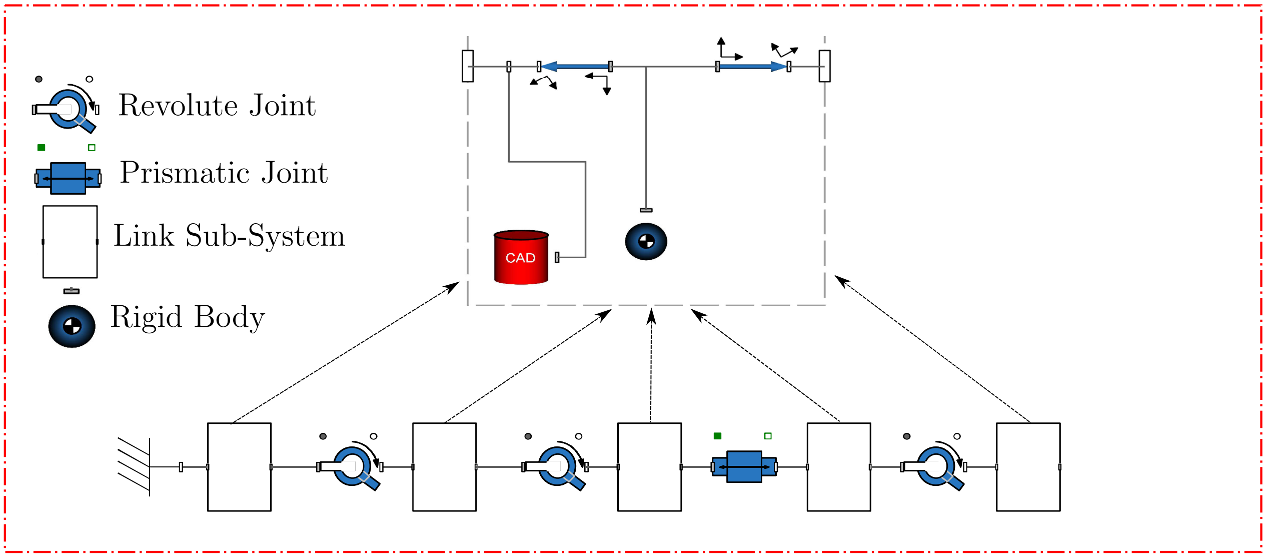


Figure 3.12: Robot model in MapleSim.

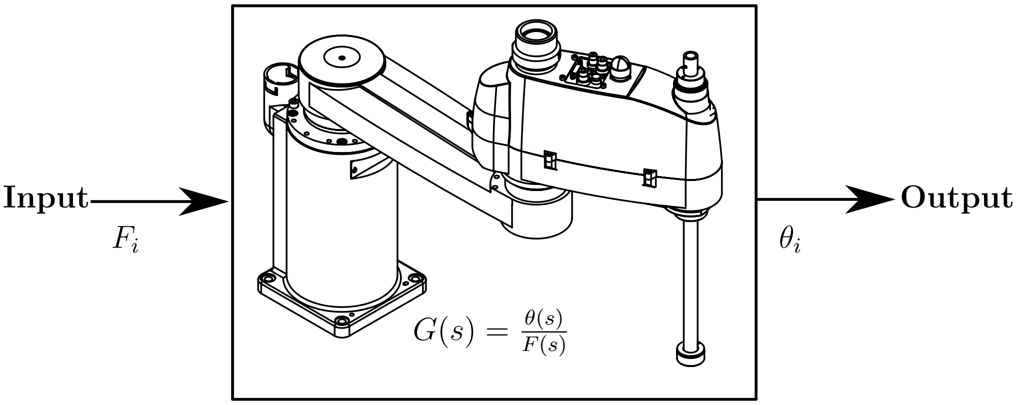


Figure 3.13: SCARA robot transfer function.

the interaction force is modeled by a linear spring K_e and a nonlinear damper D_e. In this project, the ZX plane assumed to be the external environment, i.e, if the end-effector reaches the elevation $Y < 0$ (ground); a normal interaction force F_{ext} will push the end effector in the positive Y direction and it is equal to:

$$F_{ext} = \begin{cases} K_e y \left(1 + \frac{D_e}{K_e}\dot{y}\right) & if\ y \leq 0 \\ 0 & Otherwise \end{cases} \tag{3.19}$$

In MapleSim, there are different ways to model the contact force. For instance, the relational signal blocks are used to find the elevation of the end-effector. Then the mathematical signal blocks are used to represent the force. Figure 3.14 represents the MapleSim model.

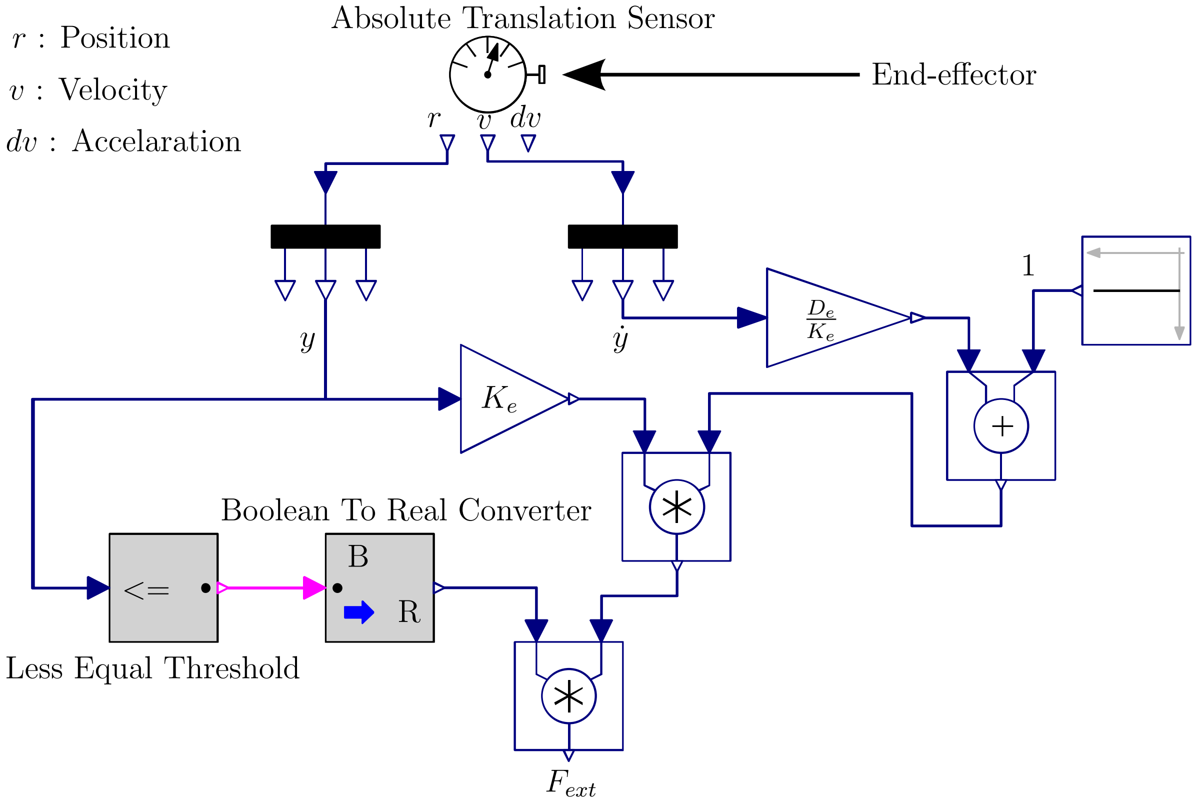


Figure 3.14: Contact force modeling.

Chapter 4

Communication Module

Master-slave configuration has been used recently in many applications. Much research has been done to improve the robustness and accuracy of this configuration. However, position and force are the two parameters that should be controlled in any master-slave system. In chapter 5, the force control will be discussed. Position tracking, on the other hand is the most important factor in any teleoperation system. Therefore an improved communication system is proposed in this chapter with reduced time delay between the master and slave systems. The source of latency between the master and slave systems is studied and formulated. Furthermore, real-time filtering and control algorithms are developed to successfully automate this system for practical applications. In addition, this chapter also describes the main architecture of the designed teleoperation system.

4.1 Master-Slave Teleoperated System

Teleoperation is widely used in different applications such as surgical robots and in hazardous environment. A standard teleoperation system Figure 4.1 consists of the following components:

1. Operator
2. Master system
3. Slave system
4. Environment
5. Shared memory
6. Communication system

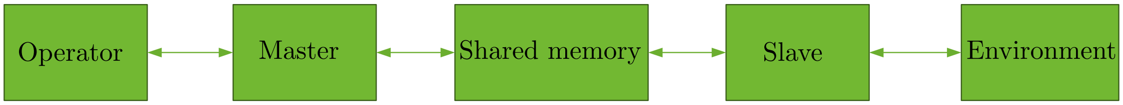
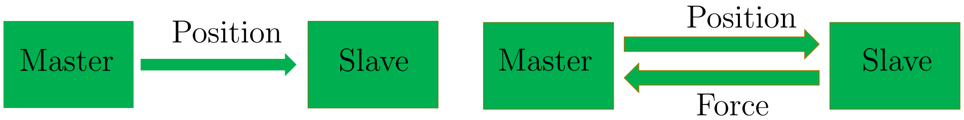


Figure 4.1: A standard single master-slave teleoperation system model.

In teleoperation, the master system is used by the operator to manipulate the environment. However, there are two kinds of teleoperation as shown in Figure 4.2:

1. Unilateral teleoperation: the slave follows the master without force feedback from the slave unit to the master unit. Therefore, the position of the slave device is only used for feedback to the shared memory between the two systems.
2. Bilateral teleoperation: the slave follows the master with force feedback from the slave unit to the master unit. In this case, force and position are both fed back to the shared memory.



(a) Unilateral Teleoperation.

(b) Biletral Teleoperation.

Figure 4.2: Teleoperation systems types.

In this study, a unilateral teleoperated master-slave system is developed using a four DOF Epson manipulator (the slave manipulator) equipped with an RC-180 controller. Also, a Polaris Vicra optical tracking device is used to represent the master device. All systems are connected to the main PC in order to control the integrated master-slave system. The slave manipulator (teleoperator) exactly replicates the movement of the master tracking device. However due to the environment, force control will be used to avoid high contact forces between the slave manipulator and the environment.

4.2 System Architecture

4.2.1 Setup Configuration

The main architecture of our master-slave teleoperated system is given schematically in Figure 4.3. The system consists of:

1. The slave system: is based on four DOF SCARA Epson manipulator equipped with an RC-180 controller. The controller communicates with the Epson RC+ 5 software installed on the main workstation using a USB port. However, since the slave manipulator needs to follow an external position signal that is sent from the master device, there is a need to establish a TCP/IP connection to continuously feed the RC+ 5 software with the position of the master system.

2. The master system: a Polaris Vicra optical tracking device is used to represent the master system.

3. Main workstation: by communicating with all systems, the workstation is used to control the master-slave system.

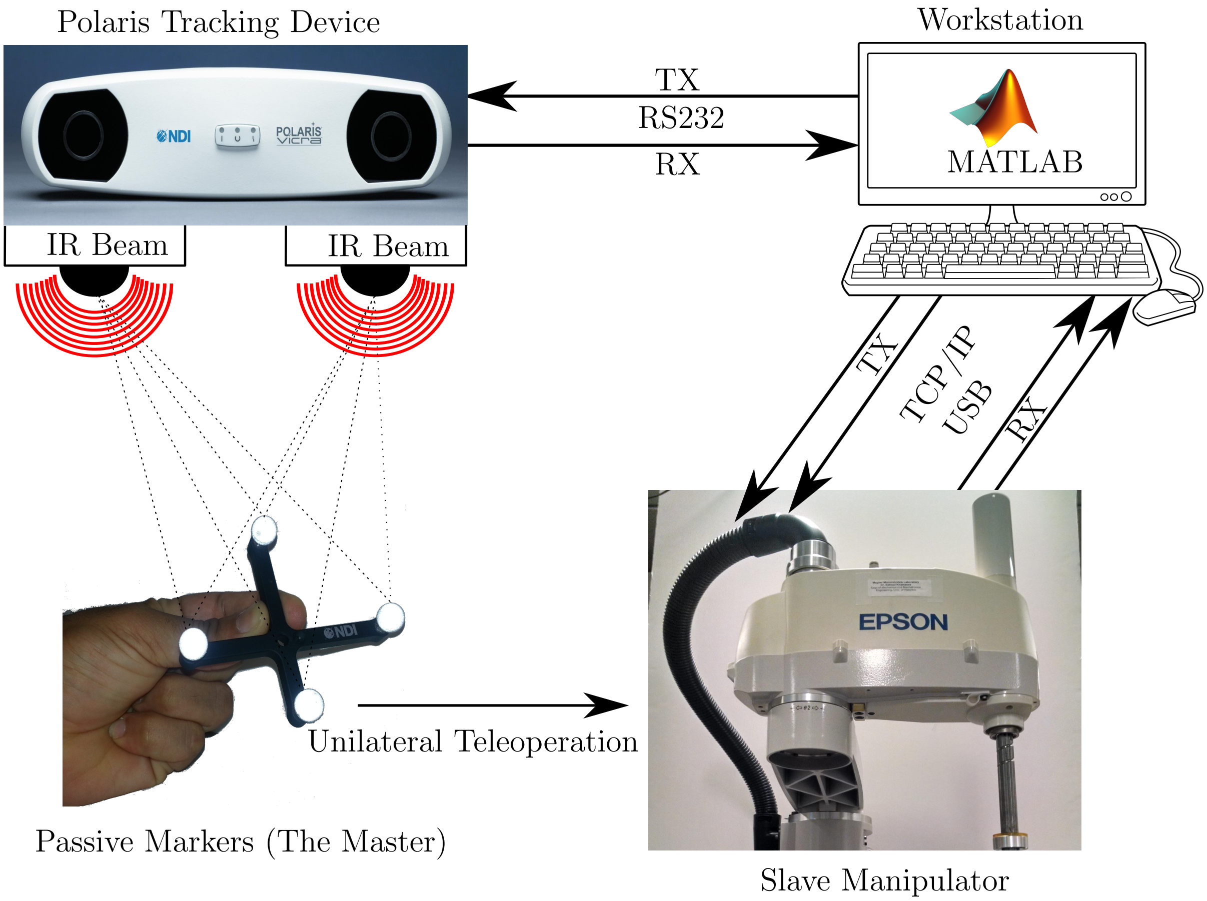


Figure 4.3: Communication architecture of the master-slave setup.

4. The Communication system: communication is used to carry information between all sub-systems of the teleoperation system. Communications between robot's controller unit and the main workstation is implemented through TCP/IP and USB connections. On the other hand, RS232 is used to establish communication between the Polaris Vicra optical tracking device and the workstation. Also, between the force-torque sensor and the main PC workstation.

The Nyquist sampling theorem states that the sampling frequency has to be at least twice that of the maximum frequency of the input signal (as known as the Nyquist Frequency). In our system, the slave robot is attempting to follow human hand motion, which operates at frequency less than 4.5 Hz [35]. Based on the technical specifications for the

Polaris tracking device, a sampling frequency of 20 Hz can be achieved. Thus the Nyquist criterion is met and the master system is capable of capturing hand motion.

4.2.2 Latency

The communication system is used to link the master and slave systems together. The first experiment is to benchmark the system behavior, and the results are presented in Figure 4.4.

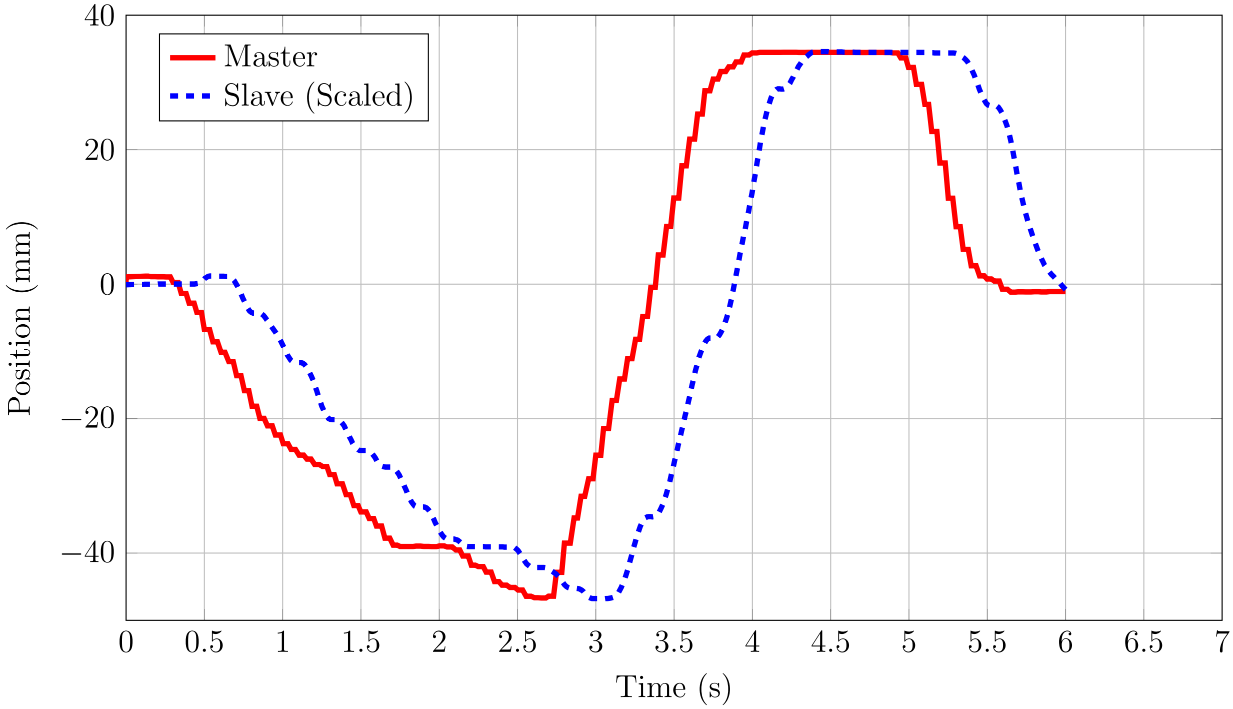


Figure 4.4: Master and slave positions in x direction.

The position of the slave was observed to be jagged and delayed compared to the master position. To accurately improve the system performance, the source of the latency needs to be studied. Therefore, the master and slave systems were investigated separately to find the source of the time delay. On the slave side, the manipulator was programmed to follow a unit step input, which is shown in Figure 4.5. The experiment was repeated using different step inputs with varying magnitudes. Based on the response of each input, the latency on the slave side can be described by the following:

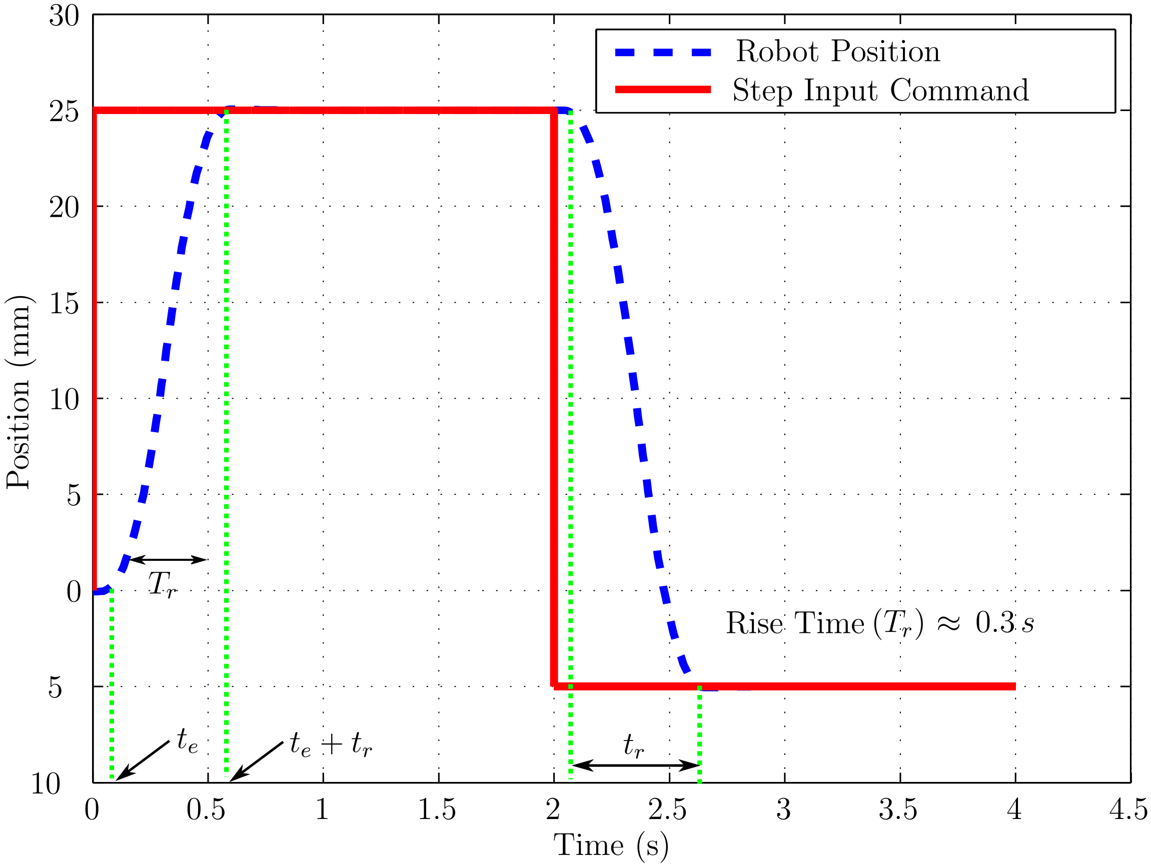


Figure 4.5: Slave robot performance in one DOF.

i. Program execution latency: There is an inherent time delay in the program execution, resulting in the manipulator not responding immediately to the given commands. According to the results illustrated in Figure 4.5, the program execution latency (t_e) is approximately $10\,ms$.

ii. Robot movement latency: The slave manipulator has four DOF with large inertia. Thus, it needs time to reach to the desired position. This time increases if the manipulator changes its direction of movement or if the total distance increases. However, this time is hard to model since it varies and depends on many parameters. Robot movement latency time (t_r) is measured for different step inputs and it is found to be approximately between $[500 - 1000]\,ms$ for a range of $[25 - 120]\,mm$.

On the master side, the maximum update rate for the Polaris Vicra tracking system is $20\,Hz$. However when the slave is connected to the master and the main workstation PC, the actual reading time for Polaris Vicra is approximately $80\,ms$ per sample. By adding all of the latency sources together the total time delay is about $[600\ -\ 1100]\,ms$.

Furthermore, in order to understand the robot movement, Matlab was used to find the manipulator dynamic model in one Cartesian DOF using the data presented in Figure 4.5. The obtained transfer function model G_m, shown in equation (4.1) is a first order system with time delay. This transfer function gives the relationship between the desired position and the commanded position. It is obvious from this equation that there is a time delay (τ_d) between the manipulator's actual position (X_a) and the desired position (X_d).

$$G_m = \frac{X_a(s)}{X_d(s)} = \frac{1}{1+s\tau}\,e^{-s\tau_d} = \frac{1}{1+s0.5}\,e^{-s0.01} \tag{4.1}$$

4.2.3 Methodology

Due to the communication latency, the teleoperated master-slave system should be stable and capable of working in real-time. As found in the literature, there are different techniques to reduce the latency. One of the best attempts is to use prediction methods such as Kalman filter, Smith predictor and linear predictive coding. However the use of predictive methods is not suitable in our application for two reasons. First a surgeon's hand motion is not predictable. Secondly, the prediction techniques are applicable only at high frequency signals (more than $2000\,Hz$). Predicting a signal requires access to at least 10 previous data points. In our case, the sampling frequency of the master device is low. Therefore the latency due to predicting the signal at the beginning is about $500\,ms$. This delay in comparison to the existing delay is high, and therefore this prediction approach is not an effective method to improve the system's performance. To improve the precision of the motion control and to decrease the time shift between the master and the slave systems, a simple algorithm is used. This method is based on asynchronous communication instead of synchronous communication. Using the asynchronous communication approach, the teleoperated master-slave system does not work based on the same timing signal. Thus the slave works independently to follow the master signal. Furthermore the slave manipulator program is divided in two threads; one is to continuously read the position of the master device, while the other is an infinite loop to execute the latest position reading from the master system. Using this approach there is a problem that affects the master-slave tracking, which is the time of executing the movement commands. As mentioned before there is a program execution latency time. This time is constant; hence there is a need to decrease interrupting times on the slave side. To address this need, a similar technique to [36] is used in this thesis. The number of position readings in the master system is decreased to a lower number of position commands on the slave side. To use this method the master signal should be smooth and not sporadic. Usually human hand motion is not smooth; therefore filtering is needed; this filtering is presented in the next section.

4.3 Filtering

As previously defined filtration is necessary, since the master, and consequently the slave are attempting to track a human movement. The filter design should be selected to work in real-time. Therefore, two types of filters are proposed.

4.3.1 Exponential Moving Average

An Exponential Moving Average (EMA) is a type of predictive filter. At any time step k, the output of the EMA filter depends on the current measurement and the last filtered data. If the X position of the passive marker is read by the Polaris Vicra tracking device at time step k, then the output filtered X_{EMA} is calculated as seen in equation (4.2).

$$X_{EMA}(k) = (1 - \alpha)X_{EMA}(k - 1) + \alpha X(k). \tag{4.2}$$

where α is the filter coefficient which has a value between $[0, 1]$. However, the difficulty in using this type of filter is based on the selection of filter coefficient α. Therefore many experiments were conducted to find this coefficient. The purpose of our experiments was to capture the Polaris Vicra tracking device signal and the filtered signal for different values of α. For our application, the best response for the master-slave movement happened when $\alpha = 0.35$.

4.3.2 Simple Moving Average

A Simple Moving Average (SMA) is a type of predictive filter. At any time step k, the output of the SMA depends on the mean of the previous n sensor data. The relation between the filtered output signal X_{SMA} and the actual reading position X is calculated using equation (4.3).

$$X_{SMA}(k) = \frac{1}{n} \sum\nolimits_{i=k-n}^{i=k-1} X(i) \tag{4.3}$$

where n is the number of entries of the average. The sampling frequency of the Polaris Vicra tracking device is low, therefore n is selected to be two in order to reduce the lag time between the filtered and the actual Polaris Vicra tracking device measurement.

Chapter 5

Force Control Algorithms

Robots force control has been studied for many years. Many schemes were introduced, but most of them are not useful in position-controlled industrial manipulators. Position-controlled manipulators have closed controllers; therefore accessing joint variables is not possible. Therefore, only position based force control techniques are suitable for these types of manipulators. Moreover, due to limitations of position control, many of the advanced force control techniques are not applicable. On the other hand, the integrated master-slave system is composed of different components connected using different types of connectors. Hence, the controller should easily work in real-time in order to reduce latency. Based on these limitations, and to accomplish any given task correctly (without delay); a simple approach that lets the manipulator work in real-time is required. Three types of controllers are presented to control the interaction force.

For industrial manipulators, there are two types of control:

i. Position control mode: in the position control mode the input for the internal controller is position. In other words, the given commands to the controller contain the end effector position in the 3-D space (one example of these commands is Move (x, y, z) where x, y and z represent a point in the 3-D space . Based on the command, the robot will adjust the joints' motor angles to get the desired position. The majority of tasks in industry such as packaging only need to control position. Therefore, most industrial manipulators are position-controlled due to this reason and also due to the simplicity of this mode.

ii. Torque control mode: this mode is more sophisticated in term of operation than the position control mode. However, in the torque control mode the robot's operator directly controls the robot's joint torques. Therefore, the input command for the internal controller contains a value for the required joint torques. The end-effector position is calculated by solving the robot's dynamic equation:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{5.1}$$

In the next sections, the proposed controllers are presented. The impedance controller will be applied to a torque-controlled SCARA model, while the admittance and fuzzy logic controllers will be applied to a position-controlled SCARA manipulator.

5.1 Impedance Controller

Impedance control was first introduced by Hogan in 1985. The aim of the impedance controller is to regulate the dynamic behavior between robot manipulator motion and the force exerted on the environment [13]. Using this control technique, the manipulator is modeled as a mechanical impedance, that contains a virtual mass m, spring k and damper D (Figure 5.1).

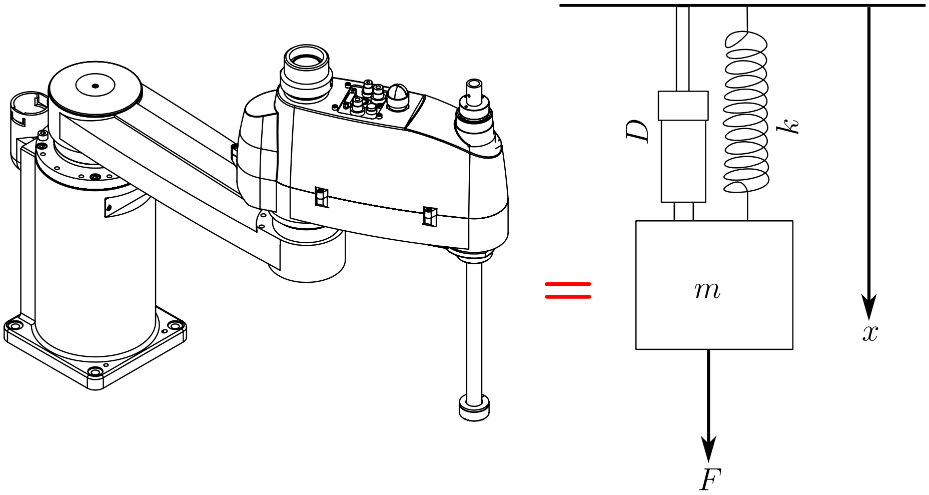


Figure 5.1: Manipulator modeling using the impedance control scheme.

The desired mechanical impedance (Z) for the manipulator end-effector is the relation between the applied force (F) and the velocity ($\dot{X}$). In the frequency domain, the impedance is given by equation (5.2).

$$F(s) = Z(s)\dot{X}(s) \tag{5.2}$$

The manipulator used in this thesis is position-controlled. So, it is difficult to implement the impedance controller to the actual manipulator. Therefore, the impedance controller was tested in the MapleSim simulation. The system block diagram is presented in Figure 5.2.

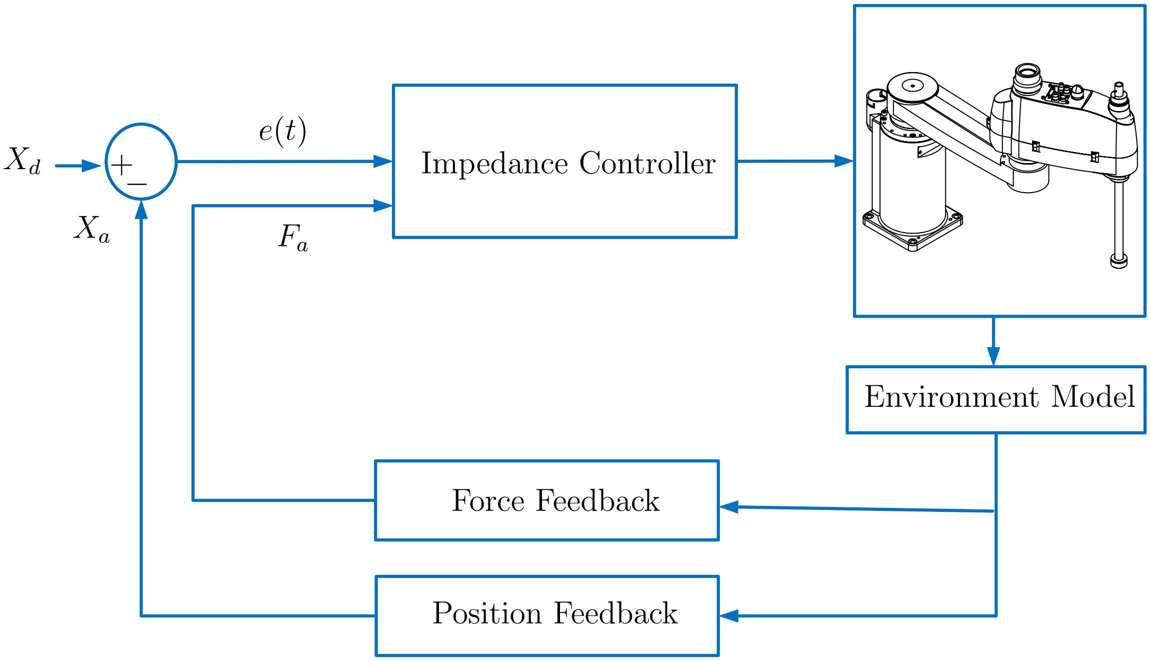


Figure 5.2: Block diagram of the proposed impedance controller.

For a SCARA manipulator, the general form of the dynamic equations of motion follows the following form:

$$\tau + \tau_f = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{5.3}$$

where τ is the input torque and τ_f represents the external interaction torques which is equal to $J_q^T(\theta)F_{ext}$. The desired mechanical impedance for the manipulator end-effector is described by:

$$F_{ext} = M_d\ddot{\theta} + D_d\dot{\theta}_e + K_d\theta_e \tag{5.4}$$

M_d, D_d and K_d represent virtual mass, stiffness and damping coefficients respectively. θ_e is the error signal between the desired position θ_d and actual position θ_a. Impedance controller however can be implemented in many forms. In this thesis, the interaction force is used to adjust the end-effector Cartesian position, which is called position based impedance control [37]. A position based Impedance controller is implemented in the vertical axis along the prismatic joint. For the vertical link, substitute $\ddot{\theta}$ from equation (5.4) in equation (5.3) and then solve for the input force F_y for the prismatic joint.

$$F_y = \left(\frac{m_3 + m_4}{M_d} - 1\right) F_{ext} - \left(\frac{m_3 + m_4}{M_d}\left(D_d\dot{\theta}_e + K_D\theta_e\right)\right) + \left(m_3 + m_4\right) g \tag{5.5}$$

g is the gravitational acceleration, m_3 and m_4 are the masses of the links 3 and 4 respectively.

5.2 Fuzzy Logic Controller

Simplicity in controller design is always desired, especially in the systems that the accuracy and fast response time are required. Fuzzy logic control (FLC) is one of the best solutions to these cases due to its simplicity and easy implementation. FLC was first introduced and implemented in 1974, to design a controller for the complex and nonlinear systems where the relationship between the input and output parameters is difficult to model analytically [38]. FLC is widely used and it solves real problems. FLC is implemented with the general form shown in equation (5.6).

$$U_{PID}(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \tag{5.6}$$

where U_{PID} is the output of the fuzzy logic controller and $e(t)$ is the error signal at time t. k_p, k_i and k_d represent proportional, integral and derivative gains respectively. It is difficult however to remove steady state error using proportional derivative (PD) FLC; in addition, it is sensitive to noise. Therefore, proportional integral (PI) FLC is more suitable for our application, and it is also simple to implement in real-time to obtain fast response. The general form for a conventional FLC-PI, follows the following form:

$$U_{PI}(t) = k_p e(t) + k_i \int e(t)dt \tag{5.7}$$

By differentiating equation (5.7) and rewriting it in the discrete time domain we obtain:

$$U_{PI}(k+1) = U_{PI}(k) + k_p de(k) + k_i e(k) \tag{5.8}$$

U_{PI} is the output of the fuzzy logic controller which represents the input increment in the z direction for the position-controlled manipulator. The input fuzzy variables are error $e(k)$ and error variation $de(k)$ at a sample interval of k. They are calculated using equations (5.9) and (5.10) respectively.

$$e(k) = F_d(k) - F_a(k) \tag{5.9}$$

$$de(k) = \frac{e(k) - e(k-1)}{T_s} \tag{5.10}$$

F_d and F_a are the desired and actual forces respectively, at time step k. Finally, the output of the fuzzy logic controller was scaled by a constant control gain k_m. The block diagram of the controller is shown in Figure 5.3 below:

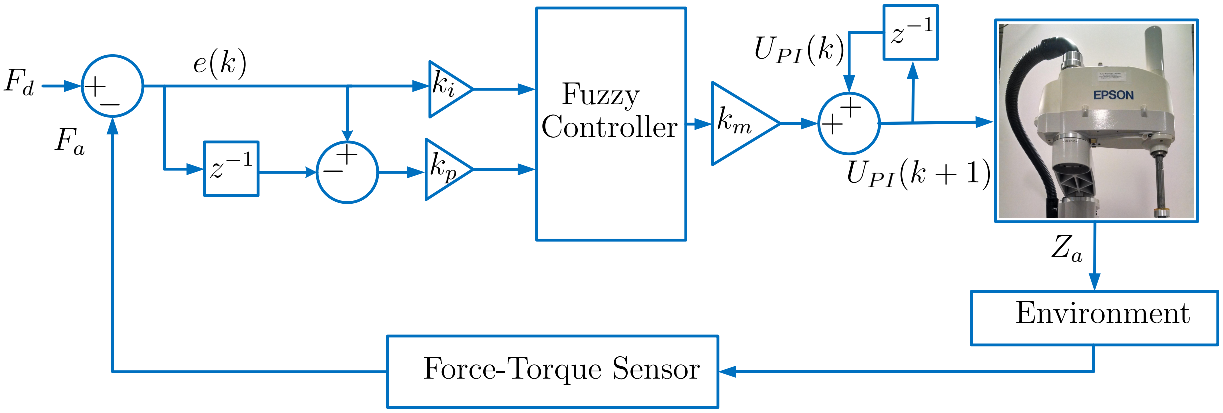


Figure 5.3: Block diagram of the proposed PI fuzzy logic controller.

The Matlab fuzzy logic toolbox was used to implement a Mamdani type fuzzy PI controller. The control variables have seven labels: positive large (PL), positive medium (PM), positive small (PS), Zero, negative large (NL), negative medium (NM), and negative small (NS). All of the control variables have the same range which is defined between -1 and 1. The membership function is shown in Figure 5.4. The rule base approach that is shown in Table 5.1 was adopted in a similar way to [39].

Table 5.1: The rule base representation.

de/e	**NL**	**NM**	**NS**	**Zero**	**PS**	**PM**	**PL**
PL	PS	PM	PM	PM	PL		
PM	Zero	PS	PM	PS			
PS	Ns	Zero	PS	PS			
Zero	NL	NM	NS	Zero	PS	PM	PL
NS	NL			NL	NS	Zero	PS
NM				NM	NS	NS	ZR
NL				NM	NM	NM	NS

The Center of Area (COA) method was used as the defuzzification method and the control gains were tuned by trial and error to obtain the best response.

5.3 Admittance Controller

As found in robot force control literature, admittance and impedance control are usually used interchangeably. However, the aim of the controller in both cases is to regulate the dynamic behavior between robot manipulator motion and the force exerted on the environment [13]. By knowing the position, speed and acceleration of all joints; the impedance controller aim is to calculate the corresponding force. The aim of the admittance control, on the other hand, is to find the position corresponding to the given contact force. In other words, the output of the impedance controller is force, while the output is position in the admittance control scheme. As a result, admittance control is more useful and widely used in industrial robots since most industrial robots have a position interface. The desired mechanical admittance for the manipulator end-effector is the relation between the velocity and resulting force. In the frequency domain, it is given by:

$$\frac{F(s)}{\dot{Z}(s)} = M_d s + \frac{K_d}{s} + D_d \tag{5.11}$$

where M_d, K_d and D_d represent the virtual mass, stiffness and damping coefficients respectively. Z represents the Cartesian position in the vertical z direction and F is the resultant normal force. Using the contact force reading at each sample time k, the increment displacement in the z direction is calculated in the discrete time domain as following:

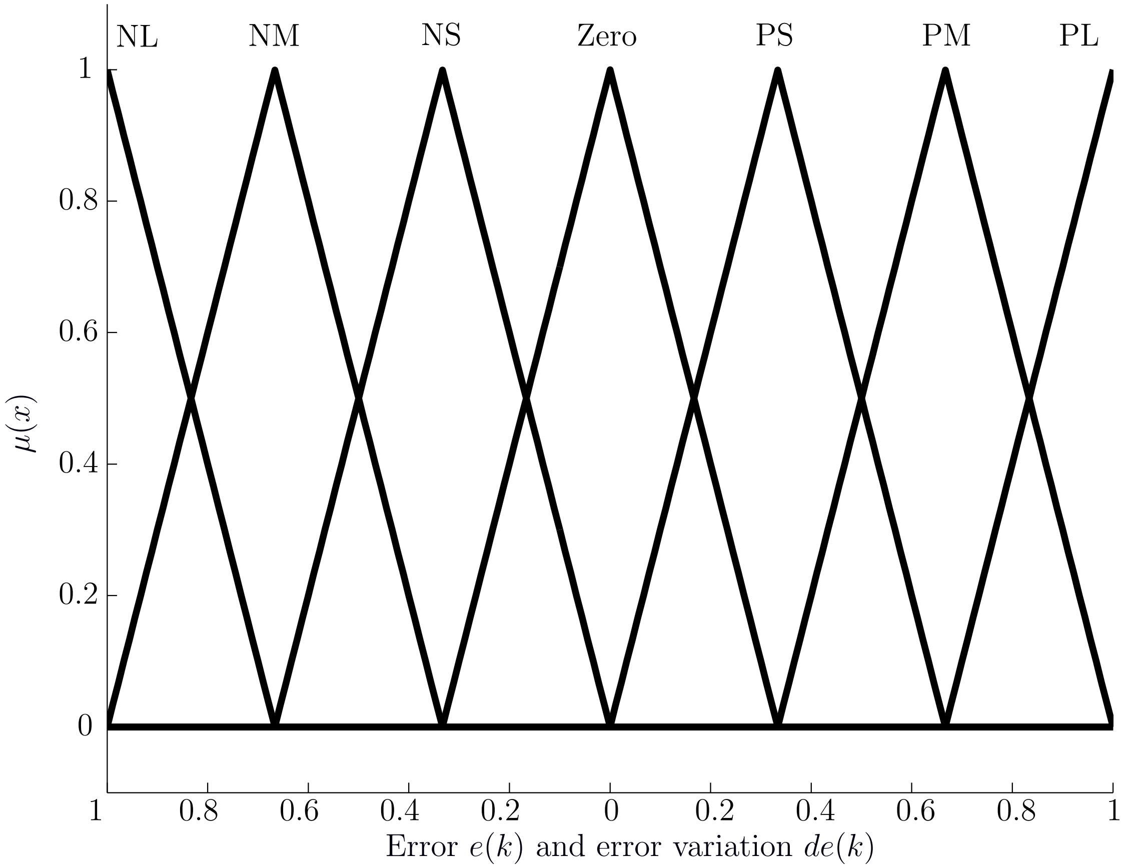


Figure 5.4: Membership function for input and output variables.

$$\Delta Z(k) = \frac{\Delta F(k) - M_d \Delta \ddot{Z}(k) - D_d \Delta \dot{Z}(k)}{K_d} \tag{5.12}$$

where:

$$\Delta \dot{Z}(k) = \frac{Z(k) - Z(k-1)}{T_s} \tag{5.13}$$

$$\Delta \ddot{Z}(k) = \frac{\Delta \dot{Z}(k) - \Delta \dot{Z}(k-1)}{T_s} \tag{5.14}$$

$$\Delta F(k) = F_d(k) - F_a(k) \tag{5.15}$$

F_d and F_a represent the desired and actual force respectively, at time step k. T_s is the system sampling time.

The admittance controller is used to control the force and diminish the error to zero. However, since the working environment is unknown, a feed-forward control is required to compensate the error that comes from the tracking device and the unknown geometry of the environment. As a result, the total incremental input to the robot position controller is ΔZ_t and given by equation (5.16) below:

$$\Delta Z_t = \Delta Z_f + \Delta Z_a \tag{5.16}$$

ΔZ_f represents the increment from the feed-forward controller and ΔZ_a the increment due to the admittance controller. The block diagram of the admittance controller is shown in Figure 5.5. The stiffness gain was found using Hooke's law. After contact, the robot was commanded a small increment, then a resultant force was calculated. Finally, by trial and error, the tuning of the mass and damping gains was achieved.

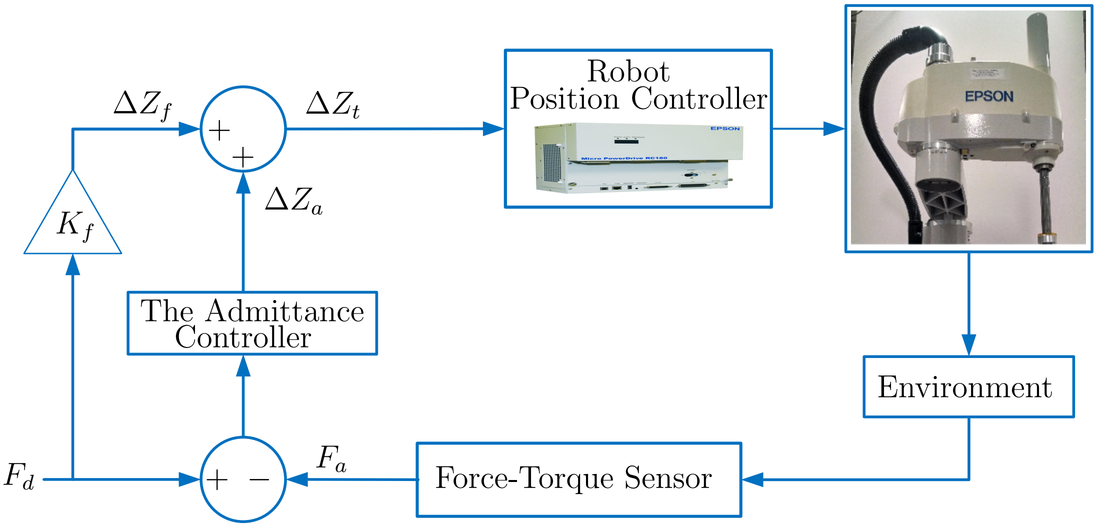


Figure 5.5: Block diagram of the proposed admittance controller.

Chapter 6

Experimental Validation

6.1 Communication Module

In order to test the presented filtering methods; the slave manipulator position was recorded and plotted in real-time. The experimental results for the position tracking are shown in Figures 6.1 and 6.2. Figure 6.1 shows the change with respect to time in the master and slave positions in the x direction using the EMA filter. By comparing this result to the initial results presented in Figure 4.4, the tracking performance has increased. The time delay is now between $[150\,-\,300]\,ms$. The motion of the slave manipulator is stable and smooth. Figure 6.2 shows the position tracking in the x direction in the case of SMA filter. Using this filtering technique the slave manipulator motion is also stable and smooth. However since the SMA filter output depends only on the current and previous data of the master position; it is not capable of removing all noise from the master device input. This noise will increase the time shift between the master and the slave systems to around $[150\,-\,450]\,ms$ due to the program execution time. In both filtering approaches, the slave manipulator was able to follow the master position with an acceptable time delay and in a stable manner. As a result, the EMA filter was chosen for our application because of the smaller time delay and stability.

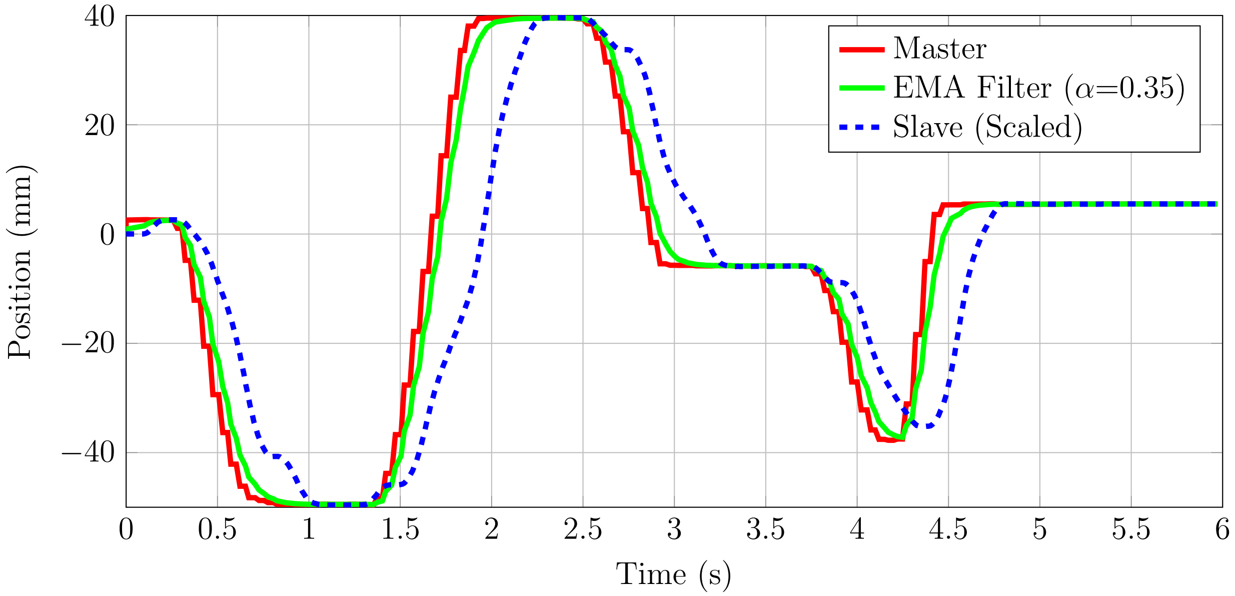


Figure 6.1: Position tracking in x direction using the EMA filter.

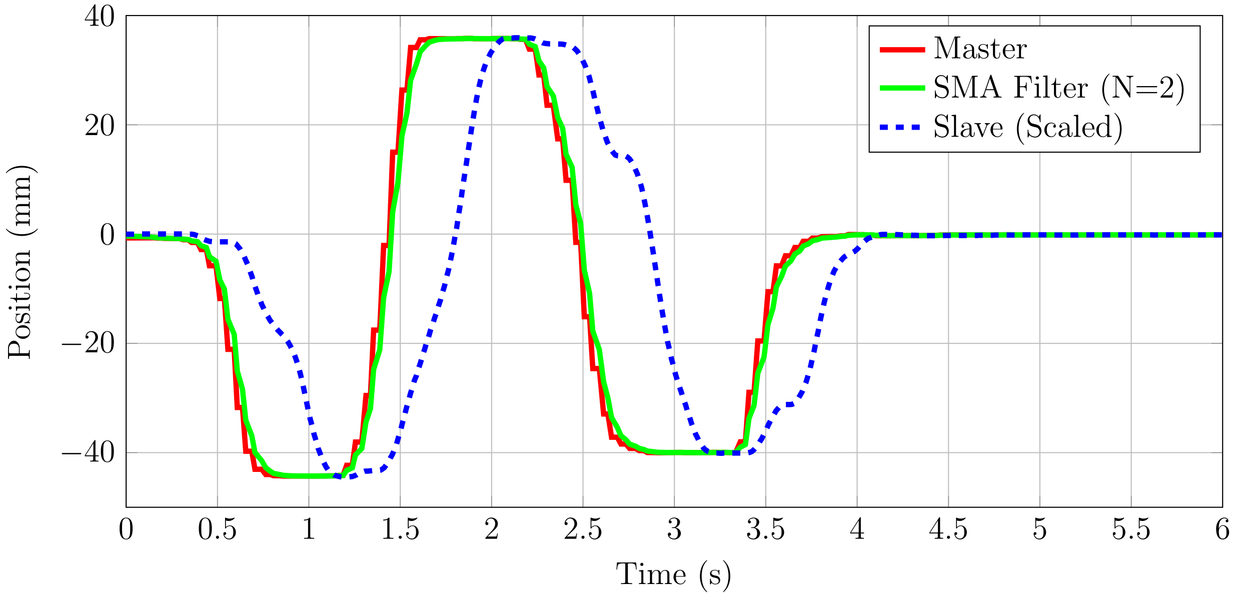


Figure 6.2: Position tracking in x direction using the SMA filter.

6.2 Dynamic Model Comparison

MapleSim and Maple were used to generate the full dynamic model of the system; the model equations are summarized in the following matrix form:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} p_1 + p_2 c_2 & p_3 + 0.5 p_2 c_2 & 0 & -p_5 \\ p_3 + 0.5 p_2 c_2 & p_3 & 0 & -p_5 \\ 0 & 0 & p_4 & 0 \\ -p_5 & -p_5 & 0 & -p_5 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} -p_2 s_2 \dot{\theta}_2 & -0.5 p_2 s_2 \dot{\theta}_1 & 0 & 0 \\ 0.5 p_2 s_2 \dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -p_4 g \\ 0 \end{bmatrix} \tag{6.1}$$

s_2 and c_2 represent $sin(\theta_2)$ and $cos(\theta_2)$ respectively, and

$$p_1 = \sum_{i=1}^{4} I_i + m_1 x_1^2 + m_2(x_1^2 + a_1^2) + (m_3 + m_4)(a_1^2 + a_2^2) \tag{6.2}$$

$$p_2 = 2a_1 x_2 m_2 + 2a_1 a_2 (m_3 + m_4) \tag{6.3}$$

$$p_3 = \sum_{i=2}^{4} I_i + m_2 x_2^2 + (m_3 + m_4) a_2^2 \tag{6.4}$$

$$p_4 = m_3 + m_4 \tag{6.5}$$

$$p_5 = I_4 \tag{6.6}$$

τ_i is the input torque (or force), I_i is the moment of inertia around the centroid, m_i is the mass of link i, x_i is the center of mass from the centroid, and a_i is the length of link i.

The results are in agreement with the results presented in [40] (which also in agreement with the results derived analytically using the Euler-Lagragian method presented in section 3.5). In addition MapleSim was able to find the full symbolic relationship between specific input and specific output. Furthermore, the full transfer function that describes the system as a closed block which can be used in other software such as Matlab Simulink.

6.3 Position-based Control for Position-controlled SCARA Manipulator

In this part, the experimental results for the admittance and fuzzy logic controllers are presented. The actual manipulator was programmed to contact with the environment (a square piece of wood with different inclined angles) at the first step. After contacting the environment, MATLAB keeps reading the desired XY position from the position tracking device (the master system) and then calculates the ΔZ increment based on the force sensor readings, and finally feeds it to the manipulator controller. The slave manipulator then moves to the desired position while keeping the required contact force with the environment. The manipulator position controller will allows the end-effector to move smoothly from the initial position (X_i, Y_i) to the desired position (X_f, Y_f) that is received from the tracking system. However, interrupting the motion or the manipulator is not possible while executing movement commands. Since the environment is unknown and the robot needs to be adjusted in the Z direction continuously, there is a need to change the final path in real-time based on the force reading and tracking device. To implement this successfully in real-time, the total distance was divided into sections with a length of $\Delta l = 10\,mm$. In both control techniques the X and Y trajectories in real-time should follow the modified programmed path [41] due to the unknown environment. The manipulator velocity in X and Y directions was set to $10\,mm\,s^{-1}$. The desired force F_d in both control schemes was set to $10\,N$. The flowchart for the process is shown in Figure 6.3.

6.3.1 Admittance Controller Results

The control parameters for the admittance controller are shown in Table 6.1. As shown in Figures 6.4 and 6.5, the admittance controller allows the robot to continue sliding along the inclined plane with an average normal contact force of $10\,N$.

Table 6.1: Admittance controller gains.

Control Gain	**Value**
K_d	$270\,N/mm$
D_d	$0.015\,N\,s/mm$
M_d	$0.009\,N\,s^2/mm$
K_f	$217\,mm/N$

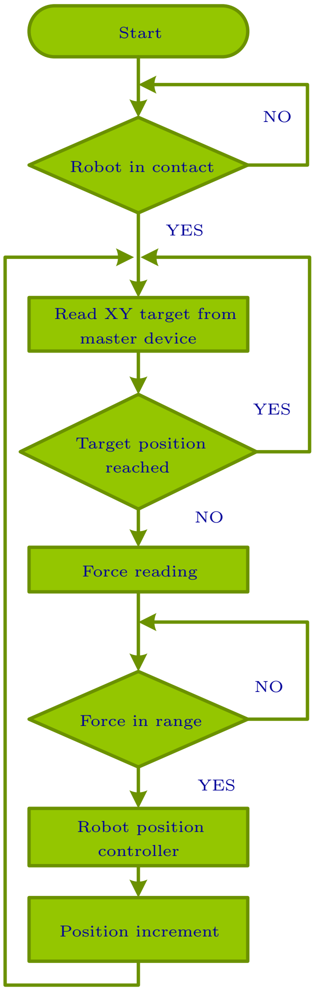


Figure 6.3: Process flowchart of the proposed method.

Using this control strategy, the error is mitigated to approximately $\pm 0.75\,N$. The damping coefficient could be further tuned in order to reduce the overshoot, though at the cost of a slower overall system response. The amount of overshoot, however, is tolerable and having fast system response is more advantageous. The final destination (X_f, Y_f) that was read from the Polaris Vicra optical tracking device was approximately $(35, 0)\,mm$. Referring to Figure 6.4, which shows the end-effector position versus time the final destination was reached while keeping the desired normal force.

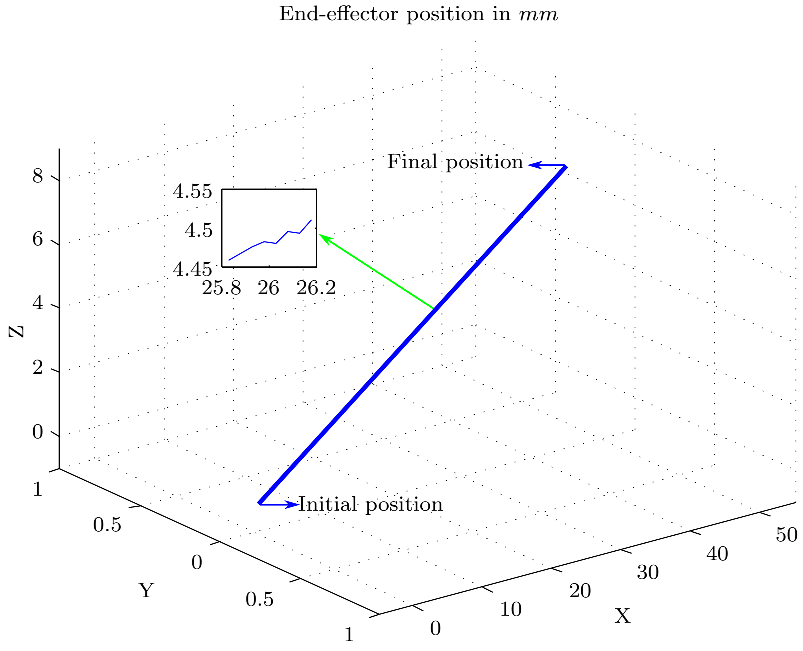


Figure 6.4: End-effector position in the 3D space using admittance control scheme.

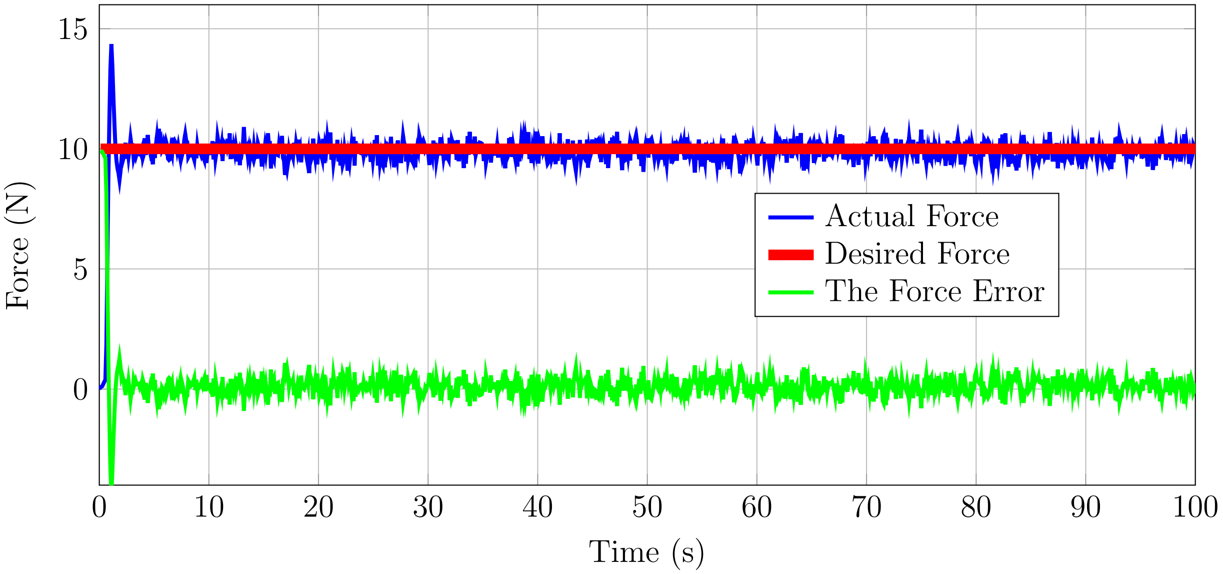


Figure 6.5: Measured normal force at the end-effector using admittance control scheme.

6.3.2 Fuzzy Logic Controller Results

The control parameters for the PI fuzzy logic controller are shown in Table 6.2. Referring to Figure 6.6, the fuzzy logic-PI controller produces less overshoot relative to the admittance controller, but a higher error (approximately ±1.5 N). To compare the two controllers with respect to the input position, the final destination (X_f, Y_f) that was read from the Polaris Vicra optical tracking device was set to approximately $(35, 0)$ mm. Referring to Figures 6.6 and 6.7, it is clear that the PI fuzzy logic control scheme was able to track the optical tracking device and control the force with an error falls between $\pm 1.5N$.

Table 6.2: Fuzzy logic controller gains.

Control Gain	**Value**
K_i	0.0167
K_p	0.3333
K_m	0.009

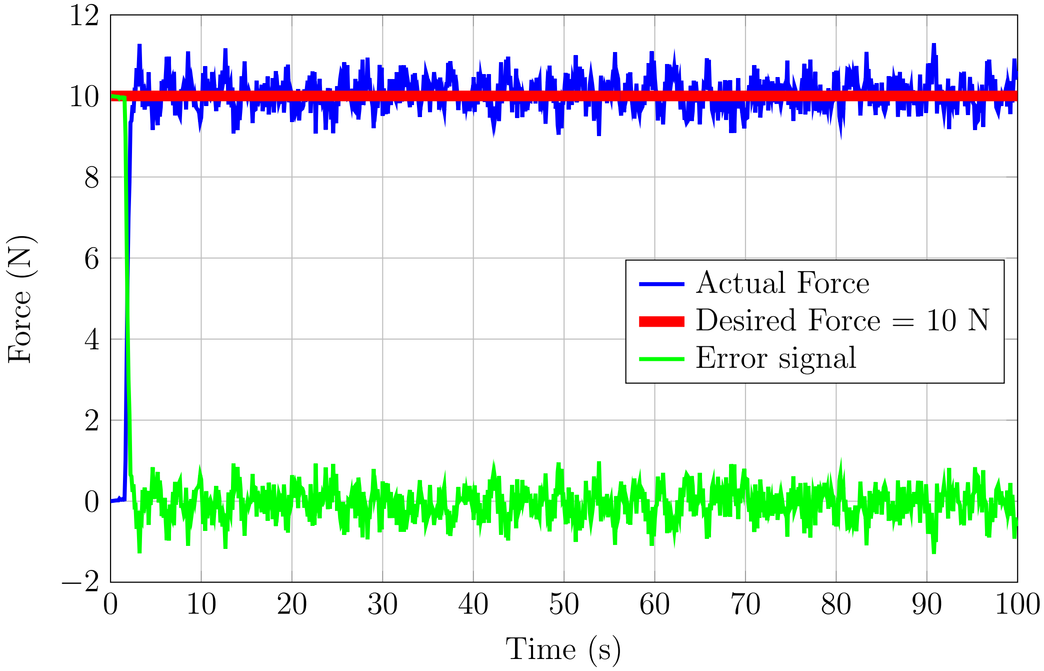


Figure 6.6: Measured normal force at the end-effector using fuzzy logic control scheme.

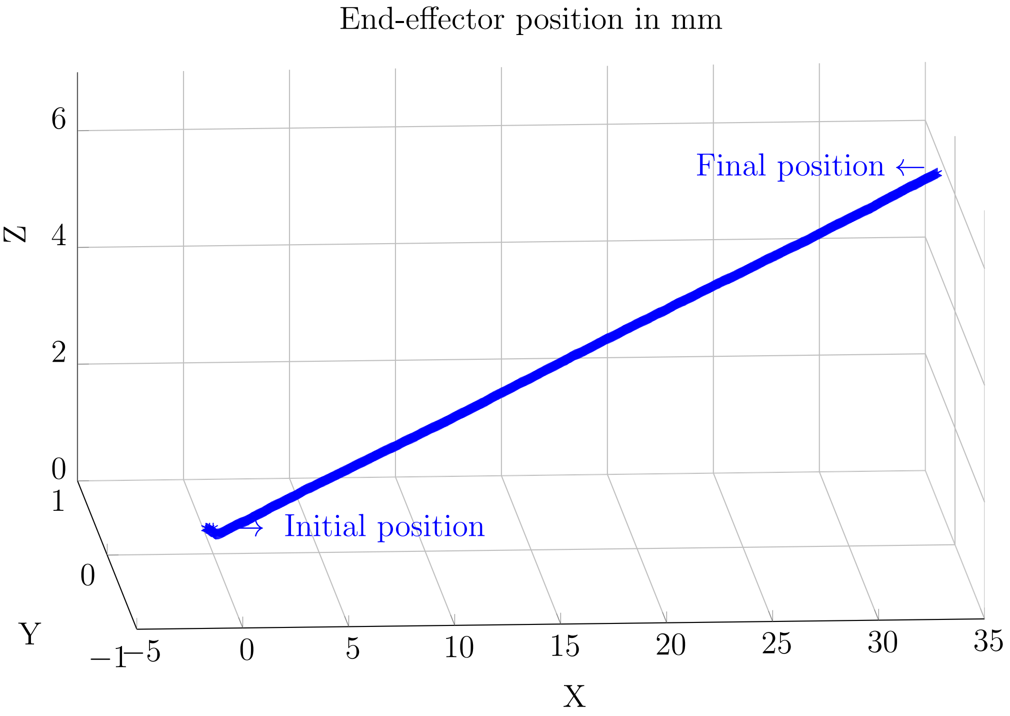


Figure 6.7: End-effector position in the 3D space using fuzzy logic control scheme.

6.4 Impedance Controller Results

In the simulation, the following control parameters are used: $K_d = 30\,N/mm$, $D_d = 10N\,s/mm$ and $M_d = 1.6\,N\,s^2/mm$. The simulation responses for position tracking and interaction force are shown in Figure 6.8. It can be seen that, after contact the end-effector position is tracking the desired position to reach the required impedance. However, due to the contact force, the tracking position error may not diminish to zero to avoid high contact force, but it is shown that both position and force are stable.

In order to test the stability and robustness of the impedance controller, the simulation was performed using different mass coefficients (stiffness and damping coefficients were fixed). Figure 6.9 shows the simulation responses for position tracking and interaction force when $M_d = 2.5\,N\,s^2/mm$. In this case, the end-effector position is also tracking the desired position to reach the required impedance. The tracking error in this case is zero, and it is better than previous simulation when $M_d = 1.6\,N\,s^2/mm$. However, the interaction force limit increased to around $40\,N$ ($25\,N$ in previous simulation).

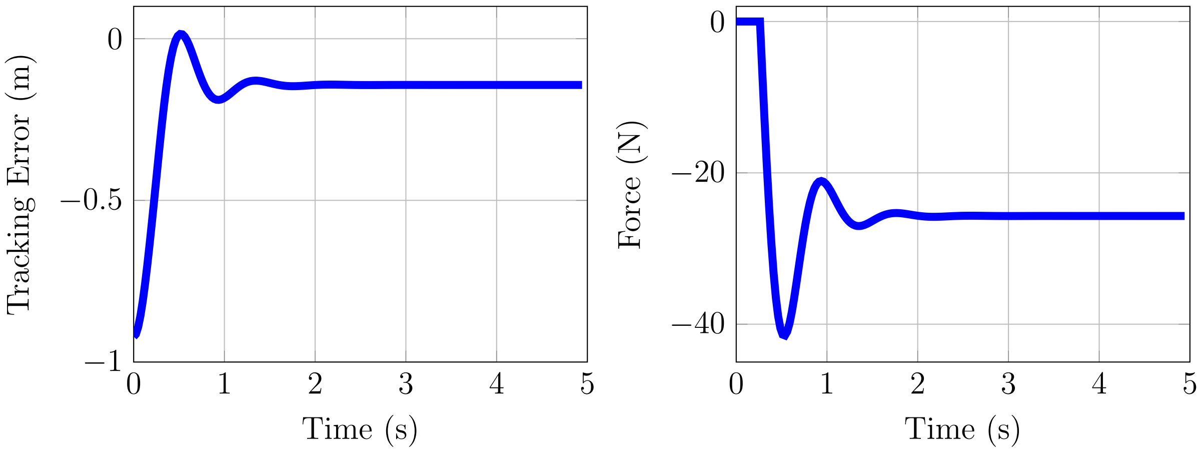


Figure 6.8: Simulation results of position tracking and interaction force using impedance control scheme $M_d = 1.6\, N\, s^2/mm$.

Referring to equation (5.5), the stability of the system depends on the dimensionless proportional gain $\frac{(m_3+m_4)}{M_d}$. If the virtual mass coefficient M_d is increased to a value of $10\, N\, s^2/mm$ (Figure 6.10), the system behavior becomes unstable. It is obviously clear that the position tracking and interaction force are unstable and diverge to infinity. As a conclusion, the system is not stable if $M_d >> (m3 + m4)$, otherwise the system is stable. Furthermore, the interaction force limit decreases accordingly with a reduction of the virtual mass gain M_d. For a specific application, the virtual mass, stiffness and damping coefficients can be optimized to limit the required interaction force.

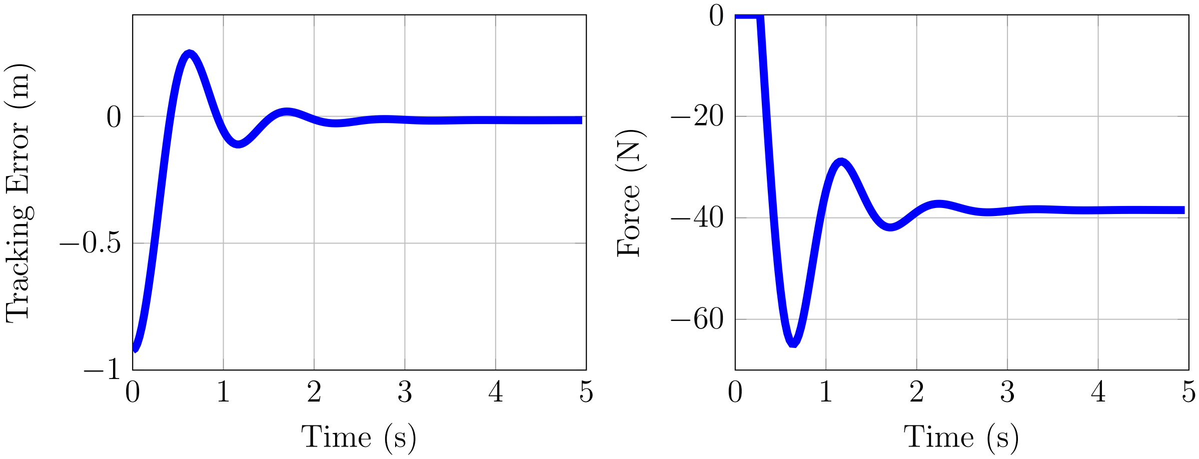


Figure 6.9: Simulation results of position tracking and interaction force using impedance control scheme $M_d = 2.5\,N\,s^2/mm$.

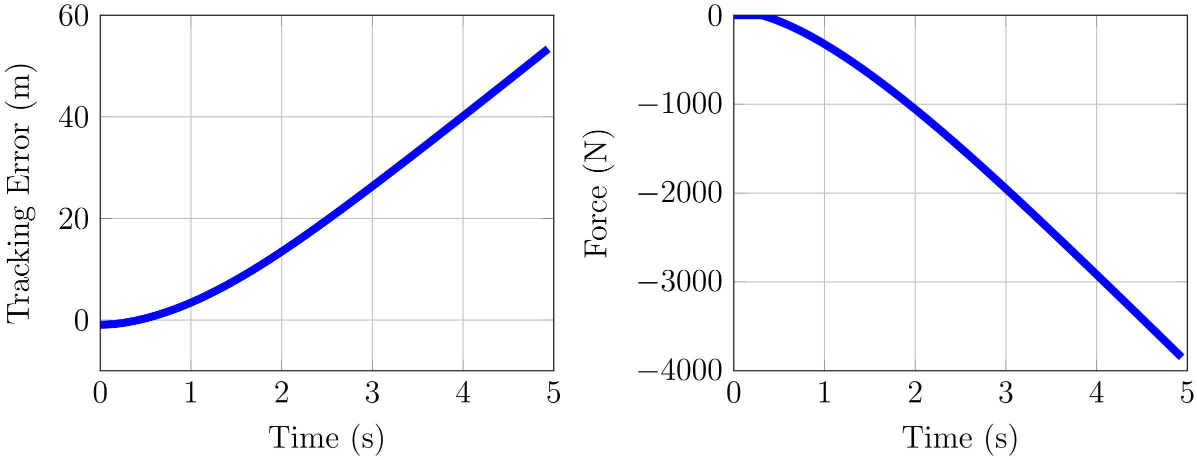


Figure 6.10: Simulation results of position tracking and interaction force using impedance control scheme $M_d = 10\,N\,s^2/mm$.

Chapter 7

Conclusions and Future Work

In this thesis, a unilateral master-slave system was implemented for different medical applications. The teleoperated system consists of the following:

1. The slave system: is based on four DOF SCARA manipulator.
2. The master system: a Polaris Vicra optical tracking device was used to represent the master system.

The position tracking and time delay between the master and slave devices are important factors in any teleoperated system. Due to this fact, the source of latency was studied and different algorithms and filtering techniques were used to reduce the time delay. The system was tested using an asynchronous communication technique. To evaluate the robustness of the presented method, a number of experiments were conducted. The results obtained show the ability of the slave manipulator to follow the master tracking system in real-time. The motion of the slave manipulator was observed to be stable and the time delay was acceptable.

The slave manipulator in any teleoprated system may interact with environment. Therefore, two different control schemes were proposed to test the developed master-slave system in the presence of contact force. However, since the slave manipulator in this work did not support force sensing, a force-torque sensor was integrated into the Epson slave manipulator to measure contact force. After measuring the contact force, force feedback was used to close the loop for the desired contact force. The Epson manipulator used in this thesis is a position-controlled manipulator. In fact, this kind of manipulator is only programmed using position input commands. On the other hand, the output parameter of the majority of the force control techniques is a torque signal. Hence, using the majority of force control techniques is not useful with this Epson manipulator. Based on these limitations, two control techniques are presented: the admittance control and the fuzzy logic control schemes. In both control schemes, the controller calculates the position increment (ΔZ) based on the force error signal. Finally, the controller sends that position increment to the manipulator internal position controller.

To test the presented controllers, several experiments were conducted. A square piece of wood with different inclined angles was used as the working object, i.e. (the environment). The unilateral master-slave system was tested in the presence of contact force as following:

1. The operator moves a rigid body consisting of four retro reflective passive markers that is connected to the operator's hand.
2. The master system (Polaris Vicra optical tracking device) measures the 2-D position of the rigid body and send it to the main workstation.
3. Matlab installed on the workstation is used to read the 2-D position (XY) of the master device. Also, based of the reading of contact force, the presented control algorithms are used to calculate the increment position in the Z direction.
4. Matlabe sends the 3-D position increment accommodation to the position-controlled SCARA manipulator.
5. The slave SCARA manipulator is moved finally to the required position while controlling the contact force.

The results show the ability of the presented control schemes (admittance and fuzzy-PI controllers), to track the operator signal while keeping the force within desired range.

In the future, the majority of robot manufacturer may embed both control modes in their robots. The user will be able to select which control mode to use (position or torque). Hence many advanced force control techniques can be used easily in industrial manipulators. The problem with the majority of advanced force control techniques is that they depend on the dynamic model of the manipulator. This fact leads to the motivation of finding an easy way of obtaining the dynamic model of any robot. Controlling a robot is successfully achieved by obtaining an accurate dynamic model of the robot. In this thesis, the dynamic model of a four DOF SCARA robot was derived using MapleSim software. To verify the derived model, an impedance controller with a specified mass, damping and stiffness was implemented to control the interaction force. In addition, MapleSim was successfully used to validate the control algorithm by providing a visual confirmation of the proposed controller. From the results, it is shown that the robot's performance can be evaluated virtually before assigning the robot to real-world tasks.

Although the results presented in this thesis have demonstrated the effectiveness of the presented teleoperated system, some improvements can be made to the current proposed teleoperated system. The proposed future work is as follows:

i. Joint feedback (position and torque) is not provided in the Epson manipulator considered in this thesis. Therefore, all of the control algorithms discussed are defined in Cartesian space. In general, applying the control algorithms in joint space is more efficient. As an improvement, the presented control algorithms should be applied in joint space.

ii. All of the control algorithms discussed in this thesis are applied to one DOF system. The manipulator motion and response to the interaction force will be better and smother if the proposed controllers are applied to all DOF's.

iii. The sample rate for the Polaris Vicra optical tracking device is low. The teleoperated system would be improved by using a higher sampling rate. As a result, the time delay between the master and slave devices will be decreased.

iv. The impedance controller in this work was not applied to the actual Epson manipulator. The admittance controller is useful in torque control manipulators and the used Epson manipulator is position-controlled. As avenues for future research, the impedance control scheme should be applied to an actual manipulator. Testing the robot in the presence of a real environment would give the opportunity to test the robustness and effectiveness of the presented controller. In addition, it would allow for better comparison between the experimental and simulation results.

v. It is recommended to implement the fuzzy logic controller with a better defuzzification method. This would allow the robot to adapt to all types of environments in order to commercialize the system. The number of rules that were using the Mamdani type needs to be reduced. In addition it also should cover all of the input fuzzy sets to work in real time with a more adaptive form. Therefore, as an improvement for the system, applying the TSK method [42] is suggested instead of the center of area defuzzification method.

vi. During the dynamic modeling of the system, the joint input motors were modeled as a pure torque source. In reality, the robot joints are driven by a motor and harmonic drives. The model for the motors and harmonic drives are usually neglected when finding the robot dynamic model. In some applications this may affect the control process or the stability of the system. Therefore, it is imperative incorporate the nonlinear models for drive system dynamics in the robots general dynamic model in order to design a high performance controller [43].

References

[1] I. Spectrum, "Robot world robot population." `http://spectrum.ieee.org/automaton/robotics/industrial-robots/041410-world-robot-population/`.

[2] NASA, "Nasa's mars curiosity rover." `http://www.nasa.gov/`.

[3] Honda, "Asimo honda." `http://asimo.honda.com/`.

[4] R. Murphy, *Introduction to AI robotics.* MIT Press, 2000.

[5] K. Aktiengesellschaft, "Kuka industrial robot." `http://www.kuka-robotics.com/en/products/industrial_robots/`.

[6] T. S. Portal, "Worldwide sales of industrial robots." `http://www.statista.com/statistics/264084/worldwide-sales-of-industrial-robots/`.

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

[8] D. E. Whitney, "Force feedback control of manipulator fine motions," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 91–97, 1977.

[9] R. C. Groome, "Force feedback steering of a teleoperator system," Master's thesis, Massachusetts Institute of Technology, 1972.

[10] P. C. Watson, *A multidimensional system analysis of the assembly process as performed by a manipulator.* Cambridge, MA: Charles Stark Draper Laboratory, 1976.

[11] J. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *IEEE Conference on Decision and Control*, vol. 19, pp. 95–100, 1980.

[12] J. L. Nevins and D. E. Whitney, *The force vector assembler concept.* Springer, 1972.

[13] N. Hogan, "Impedance control: An approach to manipulation: Part i-iii," *Journal of Dynamic Systems, Measurement, and Control*, vol. 10, pp. 1–24, 1985.

[14] S. H. Kang, M. Jin, and P. H. Chang, "A solution to the accuracy/robustness dilemma in impedance control," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 3, pp. 282–294, 2009.

[15] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[16] H. Zhang., "Kinematic stability of robot manipulators under force control," in *Robotics and Automation*, vol. 1, pp. 80–85, 1989.

[17] V. F. Filaretov and A. V. Zuev, "Adaptive force/position control of robot manipulators," in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pp. 96–101, IEEE, 2008.

[18] X. Dong, Z. Shaoguang, L. Xuerong, L. Min, and W. Hongxing, "Impedance control of robot manipulator with model reference torque observer," in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, pp. 994–998, IEEE, 2013.

[19] M. Randazzo, M. Fumagalli, F. Nori, L. Natale, G. Metta, and G. Sandini, "A comparison between joint level torque sensing and proximal f/t sensor torque estimation: implementation on the icub," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4161–4167, IEEE, 2011.

[20] T. B.Sheridan, *Telerobotics, automation, and human supervisory control.* Cambridge, MA: MIT Press, 1992.

[21] M. Tavakoli, R. V. Patel, and M. Moallem, "A force reflective master-slave system for minimally invasive surgery," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3077–3082, IEEE, 2003.

[22] M. Schurr, G. Buess, B. Neisius, and U. Voges, "Robotics and telemanipulation technologies for endoscopic surgery," *Surgical endoscopy*, vol. 14, no. 4, pp. 375–381, 2000.

[23] G. Guthart and J. K. Salisbury Jr, "The intuitivetm telesurgery system: Overview and application.," in *ICRA*, pp. 618–621, 2000.

[24] H. Reichenspurner, R. J. Damiano, M. Mack, D. H. Boehm, H. Gulbins, C. Detter, B. Meiser, R. Ellgass, and B. Reichart, "Use of the voice-controlled and computer-assisted surgical system zeus for endoscopic coronary artery bypass grafting," *The Journal of thoracic and cardiovascular surgery*, vol. 118, no. 1, pp. 11–16, 1999.

[25] I. Surgical, "Divinci system." `http://www.intuitivesurgical.com/`.

[26] EPSON, "Epson robotics." `http://robots.epson.com/`.

[27] NDI., "Polaris vicra." `http://www.ndigital.com/`.

[28] Northern Digital Inc., Waterloo-Canada, *Polaris Vicra User Guide*, 5 ed., April 2012.

[29] J. Craig, *Introduction to Robotics: Mechanics and Control.* NJ, USA: Prentice Hall, 2004.

[30] M. A. Al-Khedher and M. S. Alshamasin, "Scara robot control using neural networks," in *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*, vol. 1, pp. 126–130, IEEE, 2012.

[31] T. Benjanarasuth, N. Sowannee, and N. Naksuk, "Two-degree-of-freedom simple servo adaptive control for scara robot," in *Control Automation and Systems (ICCAS), 2010 International Conference on*, pp. 480–484, IEEE, 2010.

[32] C. Wildner and J. E. Kurek, "Calculation of scara robot model using neural nets.," in *Robot Motion and Control, 2002. RoMoCo '02. Proceedings of the Third International Workshop on*, pp. 435–439, 2002.

[33] M. Gautier, A. Janot, and P.-O. Vandanjon, "A new closed-loop output error method for parameter identification of robot dynamics," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 2, pp. 428–444, 2013.

[34] K. Hunt and F. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Journal of applied mechanics*, vol. 42, no. 2, pp. 440–445, 1975.

[35] Y. Xiong and F. Quek, "Hand motion gesture frequency properties and multimodal discourse analysis," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 353–371, 2006.

[36] E. Aoki, T. Suzuki, E. Kobayashi, I. Sakuma, K. Kozo, and M. Hashizume, "Modular design of master-slave surgical robotic system with reliable real-time control performance," in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pp. 80–86, IEEE, 2006.

[37] S. H. Kang, M. Jin, and P. H. Chang, "A solution to the accuracy/robustness dilemma in impedance control," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 3, pp. 282–294, 2009.

[38] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," in *Proceedings of the Institution of Electrical Engineers*, vol. 121, pp. 1585–1588, IET, 1974.

[39] H.-X. Li and H. Gatland, "A new methodology for designing a fuzzy logic controller," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 3, pp. 505–512, 1995.

[40] V. Philip, S. Anton, and M. Antonello, "Dynamic performance of a scara robot manipulator with uncertainty using polynomial chaos theory.," *ROBOTICS, IEEE Transactions on*, vol. 25, no. 1, pp. 206–210, 2009.

[41] F. Lange, J. Werner, J. Scharrer, and G. Hirzinger, "Assembling wheels to continuously conveyed car bodies using a standard industrial robot," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3863–3869, IEEE, 2010.

[42] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 1, pp. 116–132, 1985.

[43] M. Good, L. Sweet, and K. Strobel, "Dynamic models for control system design of integrated robot and drive systems," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 53–59, 1985.