Adaptive Fuzzy Logic Control for Time-Delayed Bilateral Teleoperation

by

Jiayi Zhu

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Biomedical Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In recent years, teleoperation has shown great potentials in different fields such as spatial, mining, under-water, etc. When teleoperation is required to be bilateral, the time delay induced by a potentially large physical distance prevents a good performance of the controller, especially in case of contact.

When bilateral teleoperation is introduced to the field of medicine, a new challenge arises: the controller needs to be used in both hard and soft environments. For example, in the context of telesurgery, the robot can enter in contact with both bone (hard) and organ (soft).

In an attempt to enrich existing controller designs to better suit the medical needs, an adaptive fuzzy logic controller is designed in this text. It simulates human intelligence and adapts the controller to environments of different stiffness coefficients. It is compared to three other classical controllers used in the field of bilateral teleopeartion and demonstrates very interesting potential.

Acknowledgements

First and foremost I'd like to thank God for guiding me into and through this adventure. I have learned much during thesis time.

With all my heart, I want to thank my parents for their continuous and unchanging love, care and support.

To my dear church family and friends, thank you so much for your prayers, suggestions and encouragements throughout this time! Thank you also for constantly reminding me that thesis is not and should not be the centre of my life.

Many thanks to my classmates as well, old and new, for the fun times we had together during my master's degree.

I would also like to thank my supervisor Dr. Gueaieb for his support, guidance and help.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Telerobotics and Telemedicine

The word telerobotics consists of two parts: “tele”, which is a Greek prefix meaning distant, and “robotics”, which refers to the study of robots. Together, they form a single word to designate the field of robot control over distance: telerobotics.

As early as 1898, the father of robotics Nikolas Tesla successfully controlled a robotic boat over distance through radio waves [85]. He also developed some of the first principles and systems to perform teleoperation in the late 19th century. But it was in 1948 that Ray Goertz first designed the modern telerobotic system in the United States [20]. This master-slave telerobotic system configuration as shown in figure 1.1 consists of two robots: the **master robot** is directly manipulated by human operator, its movements are replicated on the **slave robot** through the use of controllers and a communication channel. Therefore, the human operator can interact with a remote environment through this system.

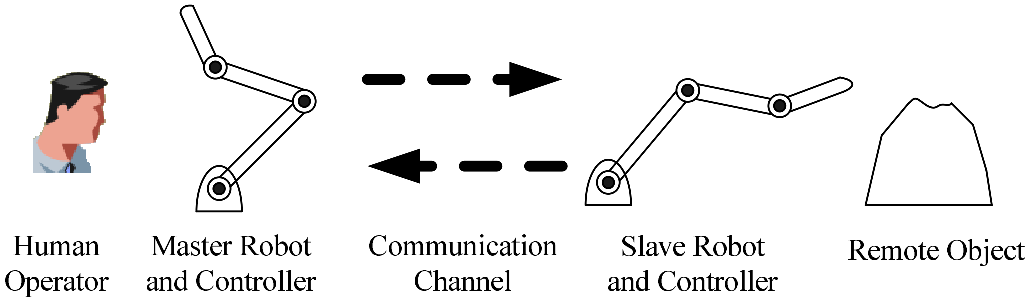


Figure 1.1: An illustration of the bilateral teleoperation system

In the case of Goertz, the system was built for handling radioactive materials and only visual information was fed back from the remote environment to the human operator. With the decline of nuclear industries in the following decades, interests were shifted to other fields. In the 1960's, the benefits of telerobotics were extended to under-water operations when different countries implemented the system on their submarines [20]. Fast forwarding to 30 years from then, the "Hot-line telerobot system" was developed to allow safe reparation of high-voltage electrical power lines by human operator from the ground level [20]. At approximately the same time in 1993, the first remotely controlled space robot Rotex flew on board of the space shuttle Columbia [20]. About a decade later in 2001, the first telerobotic surgery was successfully performed by Dr. Jacques Marescaux in New York on a patient in Strasbourg, France. It was titled Operation Lindbergh and was performed with the ZEUS robotic surgical system [51].

While being applied to nuclear, under-water, electrical and spatial fields, telerobots primarily helped to eliminate or decrease the negative effects of environment's hostility on the human operator. However when it comes to the medical field, especially in surgeries, telerobots are mainly praised for their incomparable precision as well as their great potential of expanding medical services to more patients. Indeed, robotic surgical systems can eliminate any tremor coming from surgeon's hand, moreover they can scale human movements, which makes any position shifting inside the human body more precise during the surgery. From another point of view, since the surgeons are more commonly located in bigger cities, it may sometimes be difficult for a smaller town patient to get services from them without travelling, which can be cumbersome for someone affected by illness. Therefore, telesurgery would be a great solution to make the specialist's expertise accessible to more patients. Also, having surgeons available remotely can decrease patients' waiting time, which is critical in emergency situations.

1.2 Current Challenges in Telerobotics

1.2.1 Incorporation of force feedback — bilateral telerobotics

All the telerobotic systems previously mentioned have at least one point in common: the human operator cannot feel any accurate force feedback from the remote site. In other words, the operator can feel neither hardness nor texture of the remote object. In the literature, this kind of system is described as uni-lateral because of the lack of force feedback.

Among the fields where telerobots are put into service, medicine is the one where force feedback plays a critical role due to the importance of palpation during diagnosis, treatments, and surgery. This haptic information is traditionally obtained by a surgeon while being in direct contact with the patient's skin or internal organ and it can greatly impact his/her decision making. In contrast, the need for this sensory information is not nearly as crucial in the other fields.

Indeed, if force feedback was incorporated in the telerobotic surgical system, the surgeon would be able to remotely verify the quality of suture, locate veins and bones beneath tissues, have a better force control while cutting and puncturing, and much more.

To the author's best knowledge, the currently used commercial surgical telerobotic systems (the da Vinci systems and some of the remaining ZEUS systems) don't provide any precise force feedback to the operator. Currently, the absence of the sense of touch is compensated with the images transmitted back from the remote site, a process called sensory substitution. However, task completion time and quality would be improved if force feedback was incorporated in the telerobotic system. To differ from the unilateral system mentioned in the previous section, a telerobotic system with force feedback is qualified as bilateral.

Among the five senses, seeing and hearing over distance can already be performed as images/videos and sounds can be transferred without problem. But the sense of touch cannot yet be extended to remote areas since the information to be carried over the communication channel is of a different kind. In fact, seeing and hearing only implies informatics interaction with the distant environment while touching involves the transfer of mechanical energy. It has been shown in the literature that if this force information was transferred directly, the bilateral telerobotic system will become unstable due to the time delay and would hence be unsafe to use [1, 21].

1.2.2 Presence of time delay

Time delay in teleoperation systems is the flip side of the challenge for the designers. It is always present and triggers undesirable effects in the bilateral teleoperation system. It is mainly caused by the potentially large distance between master and slave robots, by controllers' processing time and by the restriction of communication bandwidth. It is a very serious problem in surgery if combined with force feedback since it may increase the doctor's reaction time. In fact, instead of being able to stop the slave robot instantaneously upon contact, the doctor needs to wait on the communication channel. There is a delay between the time slave robot senses the contact and the time master robot senses it. There is also another delay from the time master robot issues the stopping command to the time slave robot executes it. It should be noted that for security reasons, the upper limit of time delay during a surgery is 0.33 s as suggested by Dr. Marescaux in [48].

It has been shown in [1] and [21] that even a small amount of time delay in bilateral teleoperation may cause instability to the system if the force information is directly transmitted. Numerous research works have been carried out in order to ameliorate the control system for this application, as discussed in chapter 2.

The problem of time delay was treated with a whole new perspective when the fast-growing technology of internet came into the picture. The idea of using it as a new mean of communication channel arose in the 1990's. The high availability (especially with wireless) and cheap cost of the internet can make telerobotic systems even easier to implement in the remote areas. However the designers of telerobotic systems would also need to face the new challenges brought up by the new technology, namely the varying time delay.

This variation in time delay is caused by the changes of the network congestion status. Therefore the information in different packets would take different routes to travel from master robot to slave robot and vice versa. Those routes would have different lengths which would produce a variation in time spent along the way, hence would cause the variable time delay. If the user datagram protocol (UDP) is being used, data loss and packet ordering would also be issues to be addressed. With variable delay, it is possible to have only a few data arriving in a long period of time, causing the signal to be stretched. On the other hand, if a lot of data arrive almost simultaneously, they can lead to shock-waves.

1.3 Thesis Objectives and Contributions

The objective of this work is to design a controller for bilateral telerobotic system under constant time delay suitable for both soft and hard contact environments. The controller will be created based on the concepts of adaptive fuzzy logic control. The design criteria include system stability and maximisation of system transparency. Stability refers to the fact that a bounded input from the human operator results in a bounded output. Transparency on the other hand is twofold. On one hand, it denotes the degree of faithfulness slave is following master position-wise. On the other hand, it indicates the accuracy at which the force feedback is being reflected from remote to local environment. It should be noted that the design of control systems in the field of bilateral teleoperation usually needs to deal with a trade-off between stability and transparency. Because stability involves a limited bandwidth for data transmission, but if the bandwidth is too small, not all the data will be transferred at once, which causes low transparency.

This thesis mainly contributes to the field of bilateral teleoperation with time delay by exploring how an adaptive fuzzy logic controller (AFLC) can be designed and how does it perform in this branch of robotics. A specific controller is designed and its stability rigorously proven. It is also tested in simulation against environments of different stiffness coefficients. Also, a comparison study is conducted using the results of the AFLC and that of three bench-marker controllers: PD controller, wave variable controller and second order sliding-mode controller.

It should be noted that even though the word "controller" is referred to in singular term, there are two controllers in the case of adaptive fuzzy logic controller. The one on the slave side is responsible for position tracking and the one on the master side is in charge of force feedback. They are based on the same concept but are slightly different.

The design is expected to be suitable for medical settings. Even though the concepts of telerobotics is quite similar for lots of fields in which they are in use, medicine is much more demanding than the others due to security concerns. While working in fields such as space and nuclear, human operators are well outside of the robot's workspace, but in the case of telesurgery, a human body, or part of the human body is the workspace itself. Therefore more restricted requirements should apply.

Note that since this is a first attempt of the use of AFLC in this field, the overall configuration is kept simple. The constant delay is used as it is the most basic type of time delay. Also, robots of one degree of freedom are used for the simulation in order to keep calculations and results simple and clear. Furthermore, all the controllers mentioned in the literature review (chapter 2) are simulated and/or experimented with robots of one degree of freedom for the same reason.

1.4 Summary and Thesis Outline

As described in this chapter, telerobotics is the field designating robot control over distance. While this technology has been well in use for a few decades, its proper integration in the field of medicine requires some major challenges to be overcome, namely the negative consequences caused by the incorporation of force feedback and the presence of time delay. In an attempt of building a controller suitable for the circumstances, an adaptive fuzzy logic controller is developed in this text while keeping stability and transparency as the main design criteria.

In order to present this controller, the text will be outlined in a structured manner. Readers will first be introduced to the field of bilateral teleoperation with a literature review in chapter 2, where the latest research results on controller design are categorized and explained. The design of the adaptive fuzzy logic controller is detailed mathematically in chapter 3. The subsequent chapter will explain the simulation environment and set-up. In chapter 5, the results will be shown and analysed. A conclusion will end this text in chapter 6.

Chapter 2

Literature Review

It was first with Ferrell's work [21] that the negative effects of time delay in teleoperation systems were observed. Later in the 1990s, a few scholars specifically investigated into the characteristics of bilateral time-delayed teleoperation and the relationship between its stability and transparency [15, 29, 40, 101]. Since then, researchers proposed different control schemes aiming to achieve both stability and transparency during bilateral teleoperation despite of the time delay. Those two design goals are defined as in section 1.3. Some of the proposed schemes are surveyed in [2, 5, 20, 33, 50, 68, 69, 80, 105]. These designs have mainly been proposed over the last two decades. They are presented in this chapter and are classified into categories of passivity based, prediction based and model based control.

2.1 Passivity Based Control

2.1.1 Basic concept of passivity and of wave variables

The passivity-based control was first proposed by Anderson and Spong [1] before being further developped by Niemeyer and Slotine [55]. It was considered as a major breakthrough by many scholars and triggered much research interests in the field of bilateral teleoperation starting around 1990. In this method, the communication channel of the teleoperation system was redefined in order to guarantee passivity independent of time delay. Passivity is a sufficient condition for stability, therefore guaranteeing passivity implies system stability.

A passive system is one that can only consume energy without being able to produce any. In other words, its outgoing energy is limited to the summation of the incoming

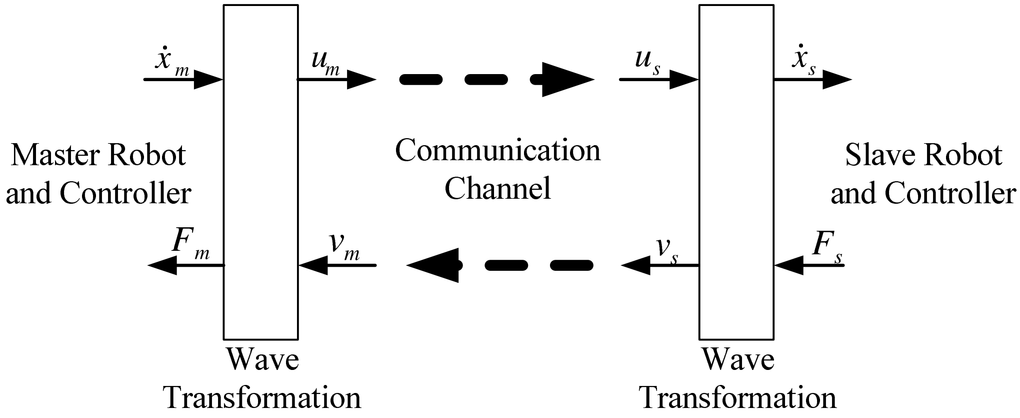


Figure 2.1: Concept of passivity control scheme based on wave variables

energy and the initially stored energy. This can be illustrated by the equation:

$$\int_0^t P_{in}(\tau)d\tau + E_{store}(0) \geq 0 \tag{2.1}$$

where $E_{store}(0)$ is the intially stored energy and P_{in} is the net power flowing into the system defined as

$$P_{in} = \dot{x}_m F_m - \dot{x}_s F_s \tag{2.2}$$

In the above equation, $\dot{x}$ represents velocity, F represents force, subscript m represents the master side and subscript s represents the slave side. By switching the power variables $\dot{x}$ and F to wave variables u and v, equation 2.1 will always be satisfied regardless of the amount of time delay, hence ensuring system's passivity. This idea is well represented by figure 2.1. The wave transformation which converges the power variables F and $\dot{x}$(force and velocity) into wave variables are as follow:

$$u = \frac{b\dot{x} + F}{\sqrt{2b}}, \qquad v = \frac{b\dot{x} - F}{\sqrt{2b}} \tag{2.3}$$

where u and v are the wave variables, $\dot{x}$ and F are the power variables velocity and force, respectively. The parameter b representing wave impedance is strictly positive and can be tuned to modify system behaviour. Stability problem is solved under the assumption that each blocks of the overall system is passive, including human operator, environment, controllers, robots and communication channel.

In [55], the application of wave variables in bilateral teleoperation was experimented using the hard contact and back drivability tests. The results show that the constant time delay can be in the range of 40 ms to 2 s without destabilizing the system. But the tracking performance of the position and force commands (transparency) of the system was quite poor. When the time delay was increased to 2 s, the force and position tracking began to severely degrade.

In fact, since passivity-based control schemes guarantee system stability, the main problem that scholars are trying to solve is transparency. Even though some passivity-based control schemes are solely based on scattering theory and not wave variables, the majority of papers still puts a large emphasis on wave variables and builds on them. Different ideas are suggested and lots of improvements have been made compared to the basic scheme.

2.1.2 Wave impedance matching

The poor transparency in the original wave variable method is believed to be mainly caused by wave reflections occurring at the junctions and terminations. These reflections take place every time a wave signal hits an element with different impedance. Since the wave variables circulate several times in the system before dying out, the reflection phenomenon can seriously degrade system's transparency. This problem is thoroughly examined in [57] and [58] along with its cause and consequence.

To cope with it, the use of wave impedance matching was proposed in the original 1991 paper by Niemeyer and Slotine [55]. It treats each subsystem as a damper, then selects and adjusts the impedance of each subsystem to be the same as the characteristic wave impedance b of the entire teleoperation system. This adjustment method is proved to ameliorate system transparency and is used fairly frequently in the wave-variable based control schemes, such as in [31, 42, 54, 56, 58, 88].

It should be noted however that impedance matching is usually used as a basic amelioration of wave variable method and is often combined with one or more other techniques to further improve system transparency.

2.1.3 Wave integrals

Another method used to increase transparency consists of transferring the wave integrals. In fact, another main cause for the lack in system transparency is the use of wave variables obtained from velocity information. Theoretically, a perfect velocity tracking would result in perfect position tracking, however practically this is impossible due to instrumentation errors and uncertainties. Hence, the need of directly using position information arose.

By integrating the wave variables u and v, the new variables obtained are then based on position(integral of velocity) and momentum (integral of force) where only the position information is used. Because of the integration/differentiation relationship between position and velocity, this method can transmit both of them together. The transfer may involve the use of an additional filter but doesn't require any extra bandwidth.

This method was investigated in [28, 49, 58, 59, 75, 100, 104].

2.1.4 Additional transfer

Similar to the above idea of transferring wave integrals, the transmission of additional power variable on top of wave variable passification was also proposed. x_m and/or x_s are often the extra variables being transferred since they contain the critical information about robot position. The force information was already carried in the original wave variables so there is no need of transferring it again.

Comparing to the transfer of wave variables, directly transferring the position information avoids redundancy (i.e. does not need to transfer momentum) and uses the information more directly instead of going through the wave transformation. However this transfer involves the use of extra bandwidth since it is using a completely different transfer line than the wave variables.

Quite a few papers explored the potentials of this idea in [4, 6, 11, 12, 16, 70, 89].

2.1.5 Energy methods

When dealing with variable time delay, the use of wave variables alone is no longer enough. Furthermore, since variable time delay is usually associated with the internet as the communication channel, the problem of data loss (while using UDP) should also be addressed simultaneously. Other issues such as packet re-ordering has not yet been looked at in depth in the controller design.

Along the idea of passification using energy concept, scholars proposed the use of adaptive gains and energy margin method. The energy methods make the system passive by limiting the output energy. A few papers used energy methods to passify teleoperation systems with variable time delay as in [10, 32, 47, 75, 86, 89].

2.1.6 Other methods

Some control schemes use different methods than those mentioned in the previous sections to passify the system. The majority of them still builds on the concept of wave variables, but uses other techniques to improve system transparency.

In [7], time delay identification for variable time delay was performed in parallel with the use of wave variables. In other papers, reset control was applied along with wave variables [19, 90]. A modified wave variables method was proposed in [35]. An augmented wave-varable-based approached was suggested in [99]. Virtual delay was used with wave variables to cope with variable time delay in [39].

Other passivity based methods not involving wave variables include the use of 3-channel control architecture [30] and the adaptive modification of wave impedance b [42].

2.1.7 Analysis

Many scholars are pursuing their research in the branch of passivity-based control to build the ideal controller. The biggest advantage of the wave variable based methods is that they guarantee stability and are completely robust to constant delays. Also, the wave commands control the output energy in terms of strength and direction only, so it can converge it to either motion or force. Hence, the controller can be the same for when manipulator is in free space and when it is in contact with the environment, without any need for switching. Additionally, those methods don't require any knowledge of the system dynamics since stability is evaluated based on inputs and outputs only.

The main disadvantage of this control method is the lack of good transparency especially when time delay grows bigger or becomes variable. Also, the control schemes in this category are limited by the assumption of human operator and environment being both passive. Some additional practical limitations are reviewed more in depth in [87].

Passivity is a sufficient condition for system stability. Therefore the passivity based control schemes reviewed in this section can all guarantee the stability of the bilateral teleoperation system. When the system is tested under constant time delay, the scattering theory plays an important role in the scheme design. When the delay is varying, some extra variables, control parameters, or special communication channels are needed to ensure the stability and improve the tracking performance of the system.

Passivity based control methods have also been combined with the other major control types in order to reach system stability and good transparency. Those methods are discussed in section 2.2.4.

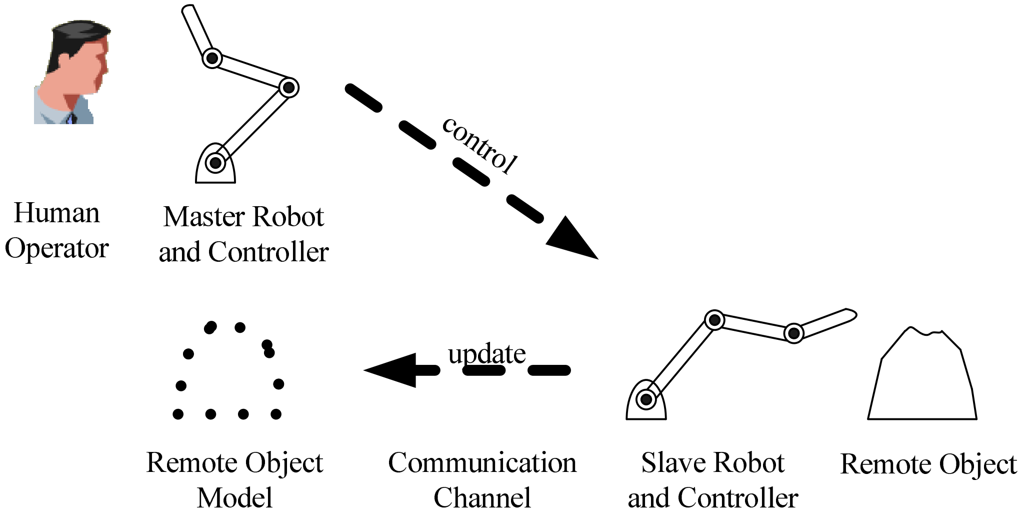


Figure 2.2: Concept of prediction based control scheme

2.2 Prediction Based Control

2.2.1 Basic concept of prediction based control

The concept of Smith controller was first applied to bilateral teleoperation systems around the time wave variables were suggested. Smith controller was first proposed in 1957 to deal with delays in chemical plants [84]. In this method, the main idea is to replicate the response of the slave and the environment on the master side by building a local model on the master side. Hence this predictive local model is a representation of the remote object. Once established, the model can completely replicate the response of the remote object, taking out the negative effects of time delay. The basic concept of the prediction based control is shown in figure 2.2.

2.2.2 Predicting environment

In the majority of prediction based control schemes, the slave robot and the remote environment form the plant which needs to be forecast, as per the original idea of Smith controller. By modelling this plant on the master side, the time delay would be removed and stability guaranteed.

Different kinds of predictors have been suggested for this purpose over the years such as those in [22, 34, 37, 78, 82, 93, 94]. Those predictors are proposed to be located in the side of the master robot. They make use of methods such as Kalman filter, linear predictor, adaptive linear predictor, neural network estimation or other in order to predict the future states of the plant. Their performance greatly depends on the assumptions made by the respective authors, for example, linear predictor works well only with unchanging or slowly varying environment while neural network can be used for fast changing sites.

2.2.3 Predicting other parts of the system

There are also a few papers in the literature that use predictors in order to estimate the characteristics of other parts in the system. For example, to predict the states of the master robot [3], the human input [83], or environment only [102].

2.2.4 Combination of prediction and passivity based control

Another trend that is gaining popularity is the combination of passivity based control with prediction based control. In this method, the predictor is used to estimate wave variables. Because the wave variable based control schemes cannot provide a satisfactory system transparency especially for large and/or varying time delay, its combination with prediction can ameliorate the performance. Hence stability is guaranteed by the use of wave variables and possibly additional energy methods while transparency is enforced with a predictive model on the master side.

A few scholars experimented this method now known as wave prediction in [3, 8, 24, 45, 49, 52, 53].

2.2.5 Analysis

Prediction based control can be robust against relatively longer time delays (above one second), which is a greatly desired feature especially for space teleoperation. However, as readers may have noticed, it requires a very accurate model of the slave and/or of the environment in order to reach stability and good transparency and this is where the main difficulty lies. By eliminating the negative effects of time delay, the presence of the prediction model makes the relationship between transparency and stability to be no longer a clear trade-off as it has traditionally been. But as previously mentioned, finding and maintaining the prediction model at an acceptable accuracy may be very challenging, especially for remote environments with time-varying characteristics.

2.3 Model Based Control

2.3.1 Basic concept of model based control

This group of control schemes includes all the concepts based on master and slave robot model dynamics. It is the traditional way to build robot controllers and in the case of teleoperation, it usually incorporates the identified delayed data in the equation to design the controllers accordingly. This is different from the prediction based category which has to completely eliminate time delay in order to preserve stability.

One of the main challenges in the model based category consists of how to accurately model the system despite of the external disturbances, frictions, payload variations, etc. It is only with an accurate information that the controller can produce the desired force. A linear dynamic model of the system can be written as follow:

$$M_m\ddot{x}_m + B_m\dot{x}_m + K_m x_m + f_{fm} = f_h + f_{mc} \tag{2.4}$$

$$M_s\ddot{x}_s + B_s\dot{x}_s + K_s x_s + f_{fs} = f_e + f_{sc} \tag{2.5}$$

where M, B, K are the system mass, viscous, and spring coefficient, x, $\dot{x}$ and $\ddot{x}$ are the robot end effector's position, velocity and acceleration. Subscripts m and s denote master and slave respectively, f_h is the force applied by the human operator, f_e is the opposing force reflected from the environment and is always against the motion of the robot, and f_{mc}, f_{sc} are the forces to be obtained from the respective controllers. f_f is the summation of the other forces applied on the system such as disturbance, friction, payload variation, etc. The choice on their inclusion depends on the designer's assumptions.

The nonlinear model can be written using Lagrange's equation as follow:

$$D_m(x_m)\ddot{x}_m + C_m(x_m\dot{x}_m)\dot{x}_m + G_m(x_m) + f_{fm} = f_h + f_{mc} \tag{2.6}$$

$$D_s(x_s)\ddot{x}_s + C_s(x_s, \dot{x}_s)\dot{x}_s + G_s(x_s) + f_{fs} = f_e + f_{sc} \tag{2.7}$$

where D is the moment of inertia, C is the centripetal forces and Coriolis forces, G is the gravitational forces, and the rest are as defined for the linear equation set.

2.3.2 Impedance control

As a typical model based control scheme, impedance control is one that relates the desired motion trajectory to a contact force, while taking into account the dynamics of the robot.

Several researchers investigated into the potentials of combining impedance control with sliding mode control to ensure system robustness. Sliding mode control is a method that forces the system to slide along a specific system subspace by first approaching it and then remaining on it once it is reached. In robot tracking situations such as in telerobotics, a first-order sliding mode surface is usually defined as $S = \dot{e} + e\lambda$, where $\dot{e}$ is the error variation rate between master and slave robots, e is the position error and λ is a positive constant. Several papers have contributed to the development of such controllers in [9, 25, 26, 27, 63, 64, 65].

In impedance control, the matching of impedance parameters is sometimes used as a technique to ameliorate system transparency as done in [46, 74]. It might be referred to as "impedance matching", however it is different than the one discussed in section 2.1.2 as it considers the impedance of the robot and not the wave impedance.

2.3.3 H_∞ optimal control

Even when the robot model is well represented with the corresponding parameters, different external sources can disturb the system. This includes noise, errors from the sensor, other external disturbances, etc. Any of them can easily destabilize the system. But since they are not predictable, some scholars are confronting the uncertainties they bring by using robust control. More specifically, by using the H_∞ optimal control. Some of their work can be found in [18, 43, 76, 77, 98].

2.3.4 Combination of prediction and model based control

More recently, there are attempts of using model based control together with predictive control. The combination called model predictive control was attempted in [61, 62, 79, 73, 81, 72]. In this method, the internal dynamics of the robot are heavily involved. Some authors assumed system stability with the hypothesis that a good model takes away the negative effects of time delay, as in the case of prediction based control. Other scholars used Lyapunov's method to mathematically prove system stability, as in the case of model based control.

2.3.5 Other methods

Some other classical control theories have also been applied to bilateral telerobotic system with time delay, such as shared compliance control [38], small-gain approach [67], stochastic switching approach [91], adaptive control [106], sliding mode controllers used with hybrid control [71], sliding mode observers [14], robust controller designed using Lyapunov-Krasovskii technique [41] and linear matrix inequality approach [95], etc.

2.3.6 Analysis

The model based control schemes differ from the passivity based approach with the fact that they thoroughly examine the system dynamics, while the passivity based approach only analyses system inputs and outputs. The model based control schemes contrasts from the prediction based approach with the fact that they do not attempt to reach stability by the elimination of time delay. Instead, in the equations for model based control schemes, the delayed terms are usually used directly and identified clearly.

This category of controllers needs to be mathematically proven stable since they do not base on any pre-defined conditions for stability as the two other categories. Lyapunov's method is usually used for this purpose, but some scholars used Llewellyn's stability criteria instead.

The model based controllers are fairly new compared to the two other categories, however there is still a considerable amount of scholars who persist in this field.

2.4 Summary

As detailed in this chapter, the majority of the control schemes proposed in the literature for bilateral teleoperation with time delay can be categorized into passivity based control, prediction based control and model based control, with possibility of combination between them. There are very few papers in the literature that go into other trends (such as event-based control discussed in [96, 97]).

With this literature review in mind, the reader should now have a better idea of the techniques previously attempted in the literature and what has already been tried. In the remaining chapters of the thesis, a new control method in the category of model based controller will be discussed.

Chapter 3

Adaptive Fuzzy Logic Controller Design

Fuzzy logic control or fuzzy control is a nonlinear control method first proposed in 1965 by Dr. Lotfi A. Zadeh [103]. It controls systems heuristically using a knowledge base specified by the designer, hence imitating human logic in order to perform control actions. The biggest advantage of this controller is its ability to deal with systems that are uncertain(fuzzy) due to complexity, incompleteness, disturbances etc. which are difficult to model using conventional controllers. In this chapter, the design of an adaptive fuzzy logic controller (AFLC) is presented after a brief summary of the theory. The design of both controllers (one at the master side and one at the slave side) is similar to each other. The AFLC is designed with product inference engine and centre average defuzzifier. Its adaptive law will be derived based on Lyapunov's stability theorem [1].

3.1 Adaptive Fuzzy Logic Control

In this section, the indirect adaptive fuzzy logic controller is presented. Interested readers can refer to [66] and [92] for further information on classical fuzzy logic control and other types of AFLC.

Each classical fuzzy logic controller contains a fuzzifier, a rule base, a fuzzy inference engine and a defuzzifier. There are many alternatives to choose from during the design process for the membership function type associated with the input and output fuzzy sets, the fuzzy inference type, the T-Norm operator, the S-Norm operator and the defuzzifier

[1] Lyapunov's stability theorem is explained in Appendix A

type. However once the choice is made, it is final. All the calculation methods as well as the parameters are fixed during real time operation.

With the development of adaptive control in the 1980s and 1990s, several types of adaptive fuzzy logic controllers (AFLC) have been suggested. In this section, one of the most commonly used type, the indirect AFLC is examined.

Let M be the total number of fuzzy rules in the fuzzy knowledge base and $l = 1, 2, ...M$ be its counter. Let the input vector be $x = [x_1, x_2, ...x_n]$ where n is the total number of elements in the input vector. Let p_i be the maximum number of fuzzy sets for the input x_i and $l_i = 1, 2, ..., p_i$ be its counter. For the classic fuzzy logic control, a common combination uses the product inference engine, singleton fuzzifier and centre average defuzzifier to give the following:

$$\hat{f}(x|\theta) = \frac{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} \bar{y}^{l_1 \ldots l_n} (\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i))}{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} (\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i))} \tag{3.1}$$

where $\bar{y}^{li}$ denotes the center value for each outputted fuzzy set B^l. In order to well illustrate the concept of adaptive fuzzy control, the equation 3.1 is broken down into the product of two elements θ and ξ, where ξ contains all the fixed values and θ contains all the values to be adapted in real-time.

$$\hat{f}(x|\theta) = \theta^T \xi(x) \tag{3.2}$$

where θ and ξ are defined as follow:

$$\xi_{l_1 \ldots l_n}(x) = \frac{\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} (\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i))} \tag{3.3}$$

$$\theta = [\bar{y}^{l_1}, \bar{y}^{l_2}, ...\bar{y}^{l_n}] \tag{3.4}$$

With the AFLC, all the elements of this vector are prone to be changed in order to ameliorate the controller output. An adaptation law has to be designed based on the error between the actual output and the desired output of the AFLC. In the literature, this adaptation law is usually designed based on Lyapunov's stability theorem [2]. It assures that the AFLC output tends toward the desired output and also ensures system stability by the use of the same theorem.

It is very important to point out that sometimes the AFLC's outputs may depend on multiple inputs. However each added input will cause the rule base to grow exponentially

[2] Please refer to appendix A

and this may require exceedingly large computing resources. In order to cope with this problem, an AFLC with multiple inputs is usually broken down into several AFLCs with two inputs. However there should be a common input for all two-inputs-AFLCs.

The AFLC has the combined advantages of having human-brain-like intelligence and an adaptation procedure which ameliorates it. Compared to the conventional fuzzy logic controller, it is more demanding in terms of computational power, however it also offers the possibility of the real-time adaptation for a better performance.

3.2 Slave Side Controller

The dynamics of the slave robot can be written as follow:

$$D_s\ddot{q} + C_s\dot{q} + G_s = \tau_{sc} + \tau_e \tag{3.5}$$

where $D_s(\ddot{q})$ (D_s for short) is the inertia matrix, $C_s(q,\dot{q})$ (C_s for short) is the centripetal and Coriolis torques, $G_s(q)$ (G_s for short) is the gravitational torques due to weight, τ_{sc} is the control torque to be generated by the actuator on the slave side, and τ_e is the external torque produced upon contact.

Consider the following property characterizing the robot model 3.5

$$q^T(\dot{D}_s - 2C_s)q = 0 \qquad \forall q \in R^n \tag{3.6}$$

where q is the joint position vector of dimension n. Consider q_d to be the desired position (which is the delayed master robot position: $q_d = q_m^{dy}$ where dy indicates a delayed value) and Λ to be a positive definite matrix whose eigenvalues are strictly in the right-half complex plane. The following terms are defined:

$$e = q - q_d \tag{3.7}$$

$$\dot{e} = \dot{q} - \dot{q}_d \tag{3.8}$$

$$\ddot{e} = \ddot{q} - \ddot{q}_d \tag{3.9}$$

$$s = \dot{e} + \Lambda e \tag{3.10}$$

$$\dot{s} = \ddot{e} + \Lambda\dot{e} \tag{3.11}$$

$$\dot{q}_r = \dot{q}_d - \Lambda e \tag{3.12}$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda\dot{e} \tag{3.13}$$

Let Lyapunov's function candidate be

$$V = \frac{1}{2}(s^T D_s s + \sum_{i=1}^{n} \tilde{\theta}_i^{\,T} \Gamma_i \tilde{\theta}_i) \tag{3.14}$$

where n is the number of degree of freedom, Γ_i is a positive constant and $\tilde{\theta}_i$ is the vector containing the parameters of the AFLC. Its time derivative is then

$$\dot{V} = \frac{1}{2}(\dot{s}^T D_s s + s^T \dot{D}_s s + s^T D_s \dot{s} + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i + \tilde{\theta}_i^{\,T} \dot{\Gamma}_i \tilde{\theta}_i + \tilde{\theta}_i^{\,T} \Gamma_i \dot{\tilde{\theta}}_i)) \tag{3.15}$$

Sub in the equation of slave robot property (equation 3.6) to replace $\dot{D}_s$ and knowing that $\dot{\Gamma}_i = 0$, the above equation becomes

$$\begin{aligned} \dot{V} &= \frac{1}{2}(\dot{s}^T D_s s + s^T 2C_s s + s^T D_s \dot{s} + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i + \tilde{\theta}_i^{\,T} \Gamma_i \dot{\tilde{\theta}}_i)) \\ &= s^T D_s \dot{s} + s^T C_s s + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\ &= s^T (D_s \dot{s} + C_s s) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \end{aligned} \tag{3.16}$$

Expand s by subbing in equations 3.10 and 3.11 gives

$$\dot{V} = s^T (D_s(\ddot{e} + \Lambda \dot{e}) + C_s(\dot{e} + \Lambda e)) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \tag{3.17}$$

Expand $\ddot{e}$ and $\dot{e}$ by further subbing in equations 3.8 and 3.9 gives

$$\begin{aligned} \dot{V} &= s^T (D_s((\ddot{q} - \ddot{q}_d) + \Lambda \dot{e}) + C_s((\dot{q} - \dot{q}_d) + \Lambda e)) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\ &= s^T (D_s(\ddot{q} - (\ddot{q}_d - \Lambda \dot{e})) + C_s(\dot{q} - (\dot{q}_d - \Lambda e)) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \end{aligned} \tag{3.18}$$

From the definitions of $\dot{q}_r$ and $\ddot{q}_r$ as given in equations 3.12 and 3.13, the two inner brackets can be replaced by the respective terms as follow and the terms in the equation can be reordered.

$$\begin{aligned} \dot{V} &= s^T (D_s(\ddot{q} - \ddot{q}_r) + C_s(\dot{q} - \dot{q}_r) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\ &= s^T (D_s \ddot{q} - D_s \ddot{q}_r + C_s \dot{q} - C_s \dot{q}_r) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\ &= -s^T (D_s \ddot{q}_r + C_s \dot{q}_r - D_s \ddot{q} - C_s \dot{q}) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \end{aligned} \tag{3.19}$$

Sub in the equation of slave robot dynamics equation 3.5 gives

$$\dot{V} = -s^T(D_s\ddot{q}_r + C_s\dot{q}_r + G_s - \tau_e - \tau_{sc}) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \tag{3.20}$$

Taking into consideration that D_s, C_s and G_s are not exactly known due to friction, disturbance, etc. Consider the following controller:

$$\tau_{sc} = \hat{D}_s\ddot{q}_r + \hat{C}_s\dot{q}_r + \hat{G}_s - \tau_e - K_D s + \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) \tag{3.21}$$

where $\hat{D}_s$, $\hat{C}_s$ and $\hat{G}_s$ are estimates of D_s, C_s and G_s, respectively. τ_e is the torque generated from contact with the external environment, and $\hat{F}$ is the adaptive fuzzy logic controller that compensates for the uncertainties in robot dynamics.

Then, rewriting the last line of $\dot{V}$ and sub in τ_{sc} as defined in 3.21 gives

$$\begin{aligned}
\dot{V} &= -s^T(D_s\ddot{q}_r + C_s\dot{q}_r + G_s - \tau_e - \tau_{sc}) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\
&= -s^T(D_s\ddot{q}_r + C_s\dot{q}_r + G_s - \tau_e - (\hat{D}_s\ddot{q}_r + \hat{C}_s\dot{q}_r + \hat{G}_s - \tau_e - K_D s + \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta))) \\
&\quad + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i \\
&= -s^T(\tilde{D}_s\ddot{q}_r + \tilde{C}_s\dot{q}_r + \tilde{G}_s + K_D s - \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta))) + \sum_{i=1}^{n} \dot{\tilde{\theta}}_i^T \Gamma_i \tilde{\theta}_i
\end{aligned} \tag{3.22}$$

where $\tilde{D}_s = D_s - \hat{D}_s$, $\tilde{C}_s = C_s - \hat{C}_s$ and $\tilde{G}_s = G_s - \hat{G}_s$ are the estimation errors of robot dynamics. $\hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta)$ in the above equation is the estimate of the function $F(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})$, where

$$F(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) = \tilde{D}_s\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{G} \tag{3.23}$$

Consider θ to represent the array of parameters in an adaptive fuzzy logic controller (AFLC) used to perform this estimation. It is defined as $\hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) = \theta^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})$ where θ contains the set of all adaptive parameters and ξ contains all the other parameters in the AFLC as discussed in section 3.1. Then a minimum approximation error vector is defined as

$$w = F(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) - \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta^*) \tag{3.24}$$

where θ^* is the vector of optimal parameters.

Rewriting the $\dot{V}$ equation and sub in estimation equation 3.23 to replace the three first terms gives

$$\begin{aligned}\dot{V} &= -s^T(\tilde{D}_s\ddot{q}_r + \tilde{C}_s\dot{q}_r + \tilde{G} + K_D s - \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta))) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i \\ &= -s^T(F(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) + K_D s - \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta))) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i\end{aligned} \tag{3.25}$$

Further sub in equation 3.24 to replace $F(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})$ and note that $\theta^* - \theta = \tilde{\theta}$, then

$$\begin{aligned}\dot{V} &= -s^T(w + \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta^*) + K_D s - \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta))) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i \\ &= -s^T(w + K_D s + \hat{F}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\tilde{\theta})) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i \\ &= -s^T(w + K_D s + \tilde{\theta}^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i \\ &= -s^T w - s^T K_D s - s^T\tilde{\theta}^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) + \sum_{i=1}^{n}\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i \\ &= -s^T w - s^T K_D s + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i - s_i^T\tilde{\theta}_i^T(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})))\end{aligned} \tag{3.26}$$

To ensure that $\dot{V}$ is smaller or equal to zero, the elements in the summation should be smaller than or equal to zero. To make calculations easier, make it to be equal to zero. In other words, where $i = 1, ..., n$,

$$\begin{aligned}\sum_{i=1}^{n}(\dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i - s_i^T\tilde{\theta}_i^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}))) &= 0 \\ \dot{\tilde{\theta}}_i^T\Gamma_i\tilde{\theta}_i - s_i^T\tilde{\theta}_i^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) &= 0 \\ \dot{\tilde{\theta}}_i^T\Gamma_i - s_i^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) &= 0\end{aligned} \tag{3.27}$$

Then

$$\dot{\tilde{\theta}}_i = \Gamma_i^{-1}s_i^T\xi(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) = \dot{\theta}^* - \dot{\theta} \tag{3.28}$$

Note that $\dot{\theta}^* = 0$. Then, for $i = 1, ..., n$

$$\dot{\theta}_i = -\Gamma_i^{-1}s^T\xi_i(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) \tag{3.29}$$

This gives the adaptation equation for a stable AFLC.

3.3 Master Side Controller

The dynamics of the master robot is written as

$$D_m\ddot{q} + C_m\dot{q} + G_m = \tau_h + \tau_{mc} \tag{3.30}$$

where $D_m(\ddot{q})$ (D_m for short) is the inertia matrix, $C_m(q,\dot{q})$ (C_m for short) is the centripetal and Coriolis torques, $G_m(q)$ (G_m for short) is the gravitational torque due to weight, τ_h is the control torque coming from the human operator and τ_{mc} is the control torque to be generated by the actuator on the master side.

The closed-loop impedance error equation is

$$\bar{D}_m\ddot{q} + \bar{C}_m\dot{q} + \bar{G}_m = \tau_h + \tau_e^{dy} \tag{3.31}$$

where $\bar{D}_m$,$\bar{C}_m$, $\bar{G}_m$ >0 are the desired D, M, G matrices which are are pre-set by the user. Consider the following property for the robot model eqn.m.Dynamics:

$$q^T(\dot{D}_m - 2C_m)q = 0 \qquad \forall q \in R^n \tag{3.32}$$

where q is the joint position vector of dimension n. Consider q_d to be the desired position (which is the delayed slave robot position $q_d = q_s^{dy}$ where dy indicates a delayed value) and Λ to be a positive definite matrix whose eigenvalues are strictly in the right-half complex plane. The following terms are defined:

$$e = q - q_d \tag{3.33}$$

$$\dot{e} = \dot{q} - \dot{q}_d \tag{3.34}$$

$$\ddot{e} = \ddot{q} - \ddot{q}_d \tag{3.35}$$

$$s = \dot{e} + \Lambda e \tag{3.36}$$

$$\dot{s} = \ddot{e} + \Lambda\dot{e} \tag{3.37}$$

$$\dot{q}_r = \dot{q}_d - \Lambda e \tag{3.38}$$

$$\ddot{q}_r = \ddot{q}_d - \Lambda\dot{e} \tag{3.39}$$

Design wise, the mater controller is different than the slave controller. The slave AFLC is solely responsible for estimating and compensating the errors caused by uncertainties in the slave robot dynamics parameters. However the master AFLC is responsible for the uncertainties in both master robot dynamics parameters as well as the desired impedance parameters pre-fixed by the user. Therefore the AFLC designed for the master robot will contain two parts. Whenever applies, the different parts will be identified by subscripts 1 and 2. Let Lyapunov's function candidate be

$$V = \frac{1}{2}(s^T D_m s + \sum_{i=1}^{n} \tilde{\theta}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \tilde{\theta}_{2i}) \tag{3.40}$$

where n is the number of degree of freedom, Γ_{1i} and Γ_{2i} are positive constants and $\tilde{\theta}_1$ and $\tilde{\theta}_2$ are the vectors containing AFLC's adaptive parameters. Its time derivative is

$$\begin{aligned} \dot{V} = \frac{1}{2}(&\dot{s}^T D_m s + s^T \dot{D}_m s + s^T D_m \dot{s} + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{1i}^T \dot{\Gamma}_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{1i}^T \Gamma_{1i} \dot{\tilde{\theta}}_{1i} \\ &+ \dot{\tilde{\theta}}_{2i}^T \Gamma_{2i} \tilde{\theta}_{2i} + \tilde{\theta}_{2i}^T \dot{\Gamma}_{2i} \tilde{\theta}_{2i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i})) \end{aligned} \tag{3.41}$$

Sub in equation 3.32 to replace $\dot{D}_m$ and knowing that $\dot{\Gamma}_{1i}$ and $\dot{\Gamma}_{1i}$ are both zero, the above equation becomes

$$\begin{aligned} \dot{V} &= \frac{1}{2}(\dot{s}^T D_m s + s^T 2C_m s + s^T D_m \dot{s} + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i})) \\ &= s^T D_m \dot{s} + s^T C_m s + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i}) \\ &= s^T (D_m \dot{s} + C_m s) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i}) \end{aligned} \tag{3.42}$$

Expand s by subbing in equations 3.36 and 3.37 gives

$$\dot{V} = s^T (D_m(\ddot{e} + \Lambda \dot{e}) + C_m(\dot{e} + \Lambda e)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i}) \tag{3.43}$$

Expand $\ddot{e}$ and $\dot{e}$ by further subbing in equations 3.34 and 3.35 gives

$$\begin{aligned} \dot{V} &= s^T (D_m((\ddot{q} - \ddot{q}_d) + \Lambda \dot{e}) + C_m((\dot{q} - \dot{q}_d) + \Lambda e)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i}) \\ &= s^T (D_m(\ddot{q} - (\ddot{q}_d - \Lambda \dot{e})) + C_m(\dot{q} - (\dot{q}_d - \Lambda e)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T \Gamma_{1i} \tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T \Gamma_{2i} \dot{\tilde{\theta}}_{2i}) \end{aligned} \tag{3.44}$$

From the definitions of $\dot{q}_r$ and $\ddot{q}_r$ as given in equations 3.38 and 3.39, the two inner brackets can be replaced by the respective terms as follow and the terms in the equation can be reordered.

$$\begin{aligned}\dot{V} &= s^T(D_m(\ddot{q} - \ddot{q}_r) + C_m(\dot{q} - \dot{q}_r) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \\ &= s^T(D_m\ddot{q} - D_m\ddot{q}_r + C_m\dot{q} - C_m\dot{q}_r) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \dot{\tilde{\theta}}_{2i}^T\Gamma_{2i}\tilde{\theta}_{2i}) \\ &= -s^T(D_m\ddot{q}_r + C_m\dot{q}_r - D_m\ddot{q} - C_m\dot{q}) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i})\end{aligned} \tag{3.45}$$

Sub in the master robot dynamic equation 3.30 gives

$$\dot{V} = -s^T(D_m\ddot{q}_r + C_m\dot{q}_r + G_m - \tau_h - \tau_{mc}) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \tag{3.46}$$

Sub in the close-loop impedance error equation 3.31 to replace τ_h gives

$$\begin{aligned}\dot{V} = &-s^T(D_m\ddot{q}_r + C_m\dot{q}_r + G_m + \tau_e^{dy} - \bar{D}_m\ddot{q} - \bar{C}_m\dot{q} - \bar{G}_m - \tau_{mc}) \\ &+\sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i})\end{aligned} \tag{3.47}$$

The dynamical terms D_m, C_m, G_m are considered as unknown while the dynamical terms $\bar{D}_m$, $\bar{C}_m$, $\bar{G}$ are pre-fixed by the user. Consider the controller

$$\begin{aligned}\tau_{mc} = \hat{D}_m\ddot{q}_r + \hat{C}_m\dot{q}_r + \hat{G}_m + \tau_e^{dy} - \hat{\bar{D}}_m\ddot{q} - \hat{\bar{C}}_m\dot{q} - \hat{\bar{G}} \\ -K_Ds + \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) + \hat{F}_2(\ddot{q}, \dot{q}, q|\theta)\end{aligned} \tag{3.48}$$

where τ_e is the torque generated upon contact with the environment, $\hat{F}$ is the adaptive fuzzy logic controller that compensates for the uncertainties in robot dynamics. $\hat{D}_m$, $\hat{C}_m$ and $\hat{G}_m$ are estimates of D_s, C_s and G_s respectively, while $\hat{\bar{D}}_m$, $\hat{\bar{C}}_m$ and $\hat{\bar{G}}_m$ are estimates of the ideal $\bar{D}_m$, $\bar{C}_m$, $\bar{G}_m$ respectively. It should be pointed out that the estimates for $\bar{D}_m$, $\bar{C}_m$, $\bar{G}_m$ are necessary since this set of number is considered as pre-set by user and fixed at all time. Therefore, in order to cope with environments of different stiffness, their ideal values are being estimated using $\hat{\bar{D}}_m$, $\hat{\bar{C}}_m$ and $\hat{\bar{G}}_m$.

Then,

$$\begin{aligned}
\dot{V} &= -s^T(D_m\ddot{q}_r + C_m\dot{q}_r + G_m + \tau_e^{dy} - \bar{D}_m\ddot{q} - \bar{C}_m\dot{q} - \bar{G}_m - (\hat{D}_m\ddot{q}_r + \hat{C}_m\dot{q}_r + \hat{G}_m \\
&\quad +\tau_e^{dy} - \hat{\bar{D}}_m\ddot{q} - \hat{\bar{C}}_m\dot{q} - \hat{\bar{G}}_m - K_D s + \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) + \hat{F}_2(\ddot{q}, \dot{q}, q|\theta)) \\
&\quad + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \\
&= -s^T(D_m\ddot{q}_r + C_m\dot{q}_r + G_m + \tau_e^{dy} - \bar{D}_m\ddot{q} - \bar{C}_m\dot{q} - \bar{G}_m - \hat{D}_m\ddot{q}_r - \hat{C}_m\dot{q}_r - \hat{G}_m \\
&\quad -\tau_e^{dy} + \hat{\bar{D}}_m\ddot{q} + \hat{\bar{C}}_m\dot{q} + \hat{\bar{G}}_m + K_D s - \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) - \hat{F}_2(\ddot{q}, \dot{q}, q|\theta)) \\
&\quad + \sum_{i=1}^{n}(\dot{\theta}_{1i}^{\tilde{T}}\Gamma_{1i}\tilde{\theta_{1i}} + \dot{\theta}_{2i}^{\tilde{T}}\Gamma_{2i}\tilde{\theta_{2i}}) \\
&= -s^T(\tilde{D}_m\ddot{q}_r + \tilde{C}_m\dot{q}_r + \tilde{G}_m - \tilde{\bar{D}}_m\ddot{q} - \tilde{\bar{C}}_m\dot{q} - \tilde{\bar{G}}_m + K_D s - \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) \\
&\quad -\hat{F}_2(\ddot{q}, \dot{q}, q|\theta)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i})
\end{aligned} \tag{3.49}$$

where $\tilde{D}_m = D_m - \hat{D}_m$, $\tilde{C}_m = C_m - \hat{C}_m$ and $\tilde{G}_m = G_m - \hat{G}_m$ are the estimation errors of robot dynamics, $\tilde{\bar{D}}_m = \bar{D}_m - \hat{\bar{D}}_m$, $\tilde{\bar{C}}_m = \bar{C}_m - \hat{\bar{C}}_m$, $\tilde{\bar{G}}_m = \bar{G}_m - \hat{\bar{G}}_m$ are the estimation errors of desired master robot impedance. $\hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta)$ is the estimate of the function $F_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})$ and $\hat{F}_2(q, \dot{q}, \ddot{q}|\theta)$ is the estimate of the function $F_2(q, \dot{q}, \ddot{q})$, where

$$F_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) = \tilde{D}_m\ddot{q}_r + \tilde{C}_m\dot{q}_r + \tilde{G}_m \tag{3.50}$$

$$F_2(q, \dot{q}, \ddot{q}) = -\tilde{\bar{D}}_m\ddot{q} - \tilde{\bar{B}}_m\dot{q} - \tilde{\bar{G}}_m \tag{3.51}$$

Consider θ to represent the array of parameters used in two different adaptive fuzzy logic controllers (AFLCs) used to perform the estimations. They are defined as $\hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta_1) = \theta_1^T\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})$ and $F_2(q, \dot{q}, \ddot{q}|\theta_2) = \theta_2^T\xi_2(q, \dot{q}, \ddot{q})$. A minimum approximation error vector is defined as

$$w_1 = F_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) - \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta_1^*) \tag{3.52}$$

$$w_2 = F_2(q, \dot{q}, \ddot{q}) - \hat{F}_2(q, \dot{q}, \ddot{q}|\theta_2^*) \tag{3.53}$$

where θ_1^* and θ_2^* are the vector of optimal parameters. Rewriting the last equation for $\dot{V}$

and sub in F_1 and F_2 as given in equations 3.50 and 3.51 gives

$$\begin{aligned}\dot{V} &= -s^T(\tilde{D}_m\ddot{q}_r + \tilde{C}_m\dot{q}_r + \tilde{G}_m - \tilde{\tilde{D}}_m\ddot{q} - \tilde{\tilde{B}}_m\dot{q} - \tilde{\tilde{G}}_m + K_D s - \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) \\ &\quad -\hat{F}_2(\ddot{q}, \dot{q}, q|\theta)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \\ &= -s^T(F_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) + F_2(q, \dot{q}, \ddot{q}) + K_D s - \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta) - \hat{F}_2(\ddot{q}, \dot{q}, q|\theta) \\ &\quad + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i})\end{aligned} \tag{3.54}$$

Further sub in equations 3.52 and 3.53 to replace F_1 and F_2 gives

$$\begin{aligned}\dot{V} &= -s^T(w_1 + \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta_1^*) + w_2 + \hat{F}_2(q, \dot{q}, \ddot{q}|\theta_2^*) + K_D s \\ &\quad -\hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\theta_1) - \hat{F}_2(\ddot{q}, \dot{q}, q|\theta_2)) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i})\end{aligned} \tag{3.55}$$

Note that $\theta_1^* - \theta_1 = \tilde{\theta}_1$ and $\theta_2^* - \theta_2 = \tilde{\theta}_2$, then

$$\begin{aligned}\dot{V} &= -s^T(w_1 + \hat{F}_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}|\tilde{\theta}_1) + w_2 + \hat{F}_2(q, \dot{q}, \ddot{q}|\tilde{\theta}_2) + K_D s) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \\ &= -s^T(w_1 + w_2 + \tilde{\theta}_1\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) + \tilde{\theta}_2\xi_2(q, \dot{q}, \ddot{q}) + K_D s) + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i}) \\ &= -s^T w_1 - s^T w_2 - K_D s + \sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i} - s_i^T\tilde{\theta}_{1i}\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) \\ &\quad -s_i^T\tilde{\theta}_{2i}\xi_2(q, \dot{q}, \ddot{q}))\end{aligned} \tag{3.56}$$

To ensure that $\dot{V}$ is smaller or equal to zero, the elements in the summation should be smaller than or equal to zero. To make calculations easier, make it to be equal to zero. In other words,

$$\sum_{i=1}^{n}(\dot{\tilde{\theta}}_{1i}^T\Gamma_{1i}\tilde{\theta}_{1i} + \tilde{\theta}_{2i}^T\Gamma_{2i}\dot{\tilde{\theta}}_{2i} - s_i^T\tilde{\theta}_{1i}\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) - s_i^T\tilde{\theta}_{2i}\xi_2(q, \dot{q}, \ddot{q})) = 0 \tag{3.57}$$

Then, looking at AFLC 1 and 2 separately, where $i = 1, ...n$

$$\begin{aligned} \dot{\tilde{\theta}}_{1i}^{T}\Gamma_{1i}\tilde{\theta}_{1i} - s_i^T\tilde{\theta}_{1i}\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) &= 0 \\ \dot{\tilde{\theta}}_{1i}^{T}\Gamma_{1i} - s_i^T\xi_{1i}(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) &= 0 \\ \dot{\tilde{\theta}}_{1i} &= \Gamma_{1i}^{-1}s_i^T\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q})) \\ &= \dot{\theta}_{1i}^{*} - \dot{\theta}_{1i} \\ \dot{\theta}_{1i}^{*} &= 0 \end{aligned} \tag{3.58}$$

This results in

$$\dot{\theta}_{1i} = -\Gamma_{1i}^{-1}s_i^T\xi_1(\ddot{q}_r, \dot{q}_r, q, \dot{q}, \ddot{q}) \tag{3.59}$$

$$\begin{aligned} \dot{\tilde{\theta}}_{2i}^{T}\Gamma_{2i}\tilde{\theta}_{2i} - s_i^T\tilde{\theta}_{2i}\xi_2(q, \dot{q}, \ddot{q}) &= 0 \\ \dot{\tilde{\theta}}_{2i}^{T}\Gamma_{2i} - s_i^T\xi_{2i}(q, \dot{q}, \ddot{q})) &= 0 \\ \dot{\tilde{\theta}}_{2i} &= \Gamma_{2i}^{-1}s_i^T\xi_2(q, \dot{q}, \ddot{q})) \\ &= \dot{\theta}_{2i}^{*} - \dot{\theta}_{2i} \\ \dot{\theta}_{2i}^{*} &= 0 \end{aligned} \tag{3.60}$$

This results in

$$\dot{\theta}_{2i} = -\Gamma_{2i}^{-1}s_i^T\xi_2(q, \dot{q}, \ddot{q}) \tag{3.61}$$

3.4 Summary

As seen in this chapter, the adaptive fuzzy logic controller is a type of controller that attempts to perform control actions with human logic. As analysed, the master and slave controllers are proven to be stable each on their own under the given assumptions and conditions. Lyapunov's theorem is used to prove controllers' stability as this is a classical method used in the literature. The Lyapunov's function candidate generated the final adaptation law for each controller. By using those adaptation laws, the controllers are expected to always function in a stable manner.

Chapter 4

Simulation

In this section, simulation set-up and results will be shown for four kinds of controllers: PD controller, wave variable controller, second order sliding-mode controller and adaptive fuzzy logic controller. For each of them, simulations are run with three different contact situations on the remote site: no contact, hard contact and soft contact. In this chapter, the general simulation environment is first examined, then the model built for each controller is individually investigated. All simulations are performed using MatLab version 7.1.0.246 (R14SP3) and Simulink version 6.3 (R14PS3).

4.1 General Environment

The simulation set-up used for simulation is as shown in figure 4.1. The master robot is identical to the slave robot. They consist of two one-link robots rotating around a fixed base each with an arm measuring 0.2 m. Both robots are starting in the horizontal position shown in figure 4.2. This is the original position at which robots are starting from in the beginning of each simulation. As indicated on the figures, the gravity force is considered to act downward, the gravitational acceleration term g is shown on the two graphs. The positive direction of rotation motion is in the counter-clockwise direction.

The simulation of the robot arm is done using Newton-Euler dynamics algorithm for a one-link robot (hence one degree of freedom) as detailed in Appendix B. But this method is easily expendable to robots with multiple links (hence multiple degrees of freedom) [13]. The robot mass is set to 0.1 kg and the arm length to 0.2 m. The human input is a sin function given by $f(t) = 0.5sin(2\pi t)$, measured in Newton-meter (N m), where t is time in second. It is assumed that this force is always acting perpendicularly

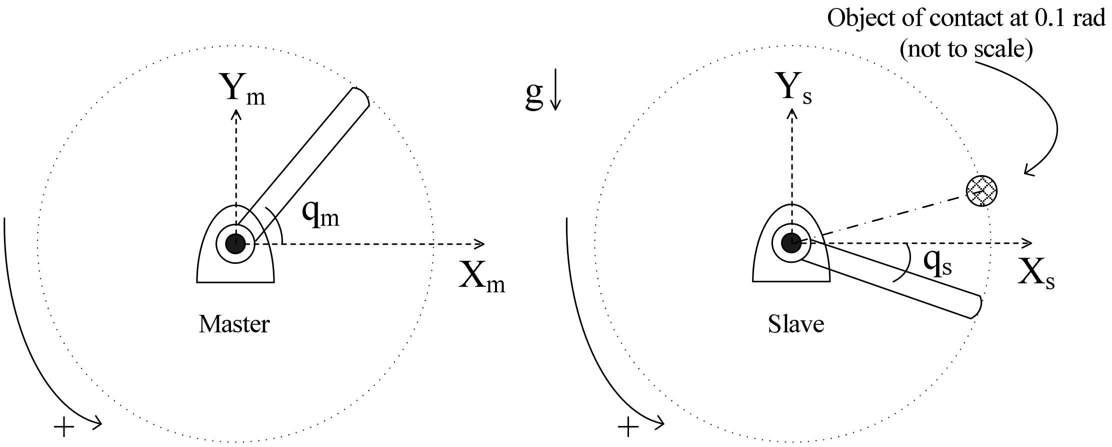


Figure 4.1: Robot arm modelled

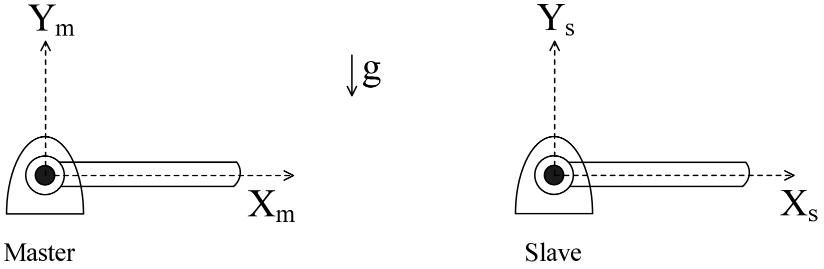


Figure 4.2: Robots' original positions

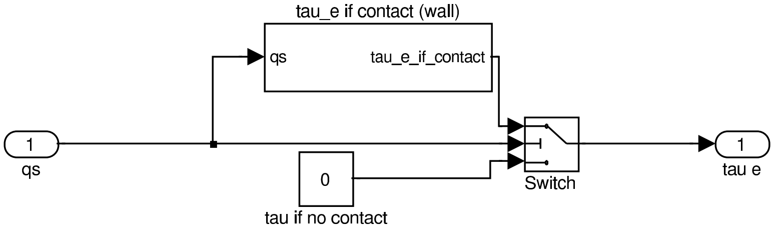


Figure 4.3: Modelling of external force

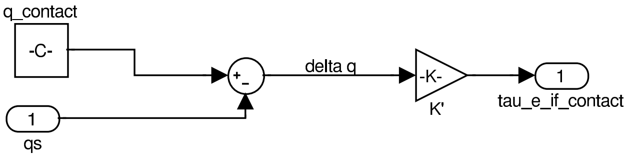


Figure 4.4: Modelling of Fe generation upon contact

to the robot arm. When the function is positive, it indicates a force turning the robot arm in the positive direction (counter-clockwise) and when it is negative, it indicates a force turning the robot arm in the negative direction (clockwise).

The environment for soft contact is modelled as a pure spring with a coefficient $45\,\mathrm{N\,m^{-1}}$ to simulate human tissue[1]. A value of $450\,\mathrm{N\,m^{-1}}$ is chosen to simulate a harder object. Given a robot arm length of $0.2\,\mathrm{m}$, the corresponding rotational coefficients to these two linear coefficients are $1.8\,\mathrm{N\,m\,rad^{-1}}$ and $18\,\mathrm{N\,m\,rad^{-1}}$

The environment is represented as a pure torsional spring and the simulation set-up is shown as in figure 4.3 and figure 4.4. Among them, figure 4.3 illustrates the equation

$$\tau_e = k(q_{contact} - q_s) \tag{4.1}$$

where $q_{contact}$ is the position of the point of contact in rad, q_s is the actual position of the slave robot in rad, and k is the torsional stiffness coefficient of the environment in $\mathrm{N\,m\,rad^{-1}}$.

For the "contact" simulation cases, $q_{contact}$ is set to $0.1\,\mathrm{rad}$ and the stiffness coefficient k varies as per situation of hard or soft contact.

It is important to mention that as shown by equation 4.1 and figure 4.4, the outputted force is non positive because $q_{contact}$ has to be smaller than q_s for contact to occur, hence to generate a force. The τ_e term throughout this work except for in the case of the second

[1] This value was found to be between 47.3 to $128.3\,\mathrm{N\,m^{-1}}$ in [23]. In order to include the correct range of stiffness, the value of $45\,\mathrm{N\,m^{-1}}$ is chosen as a lower bound for soft environment stiffness coefficient

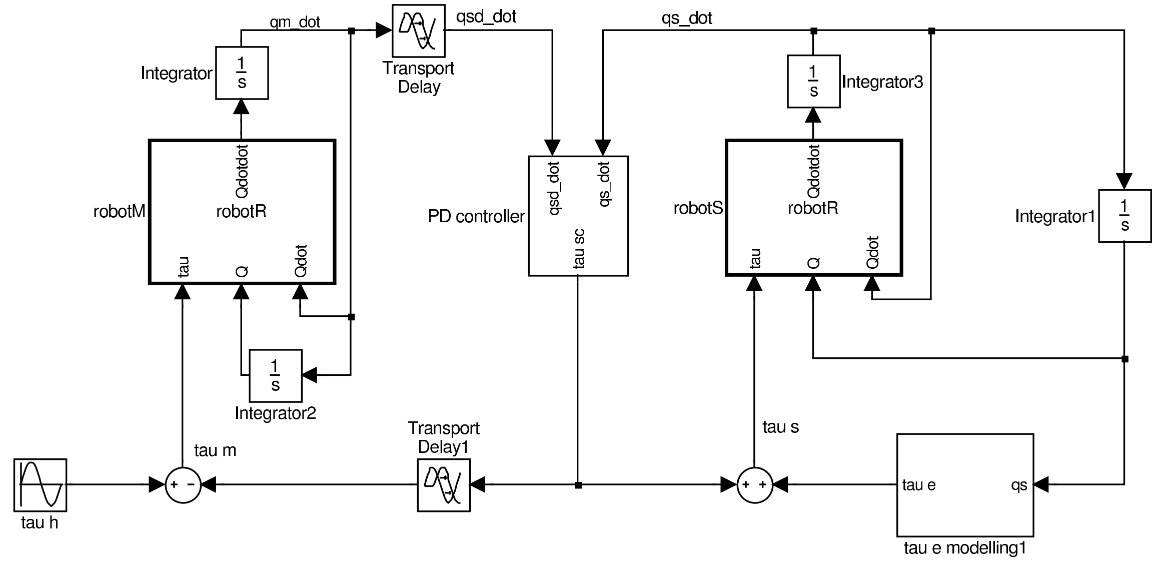


Figure 4.5: Bilateral teleoperation setup with PD controller

order sliding-mode controller, is a term that is negative by definition. This negative sign indicates that the contact force is acting opposite to the direction of motion.

4.2 Basic Proportional-Derivative (PD) controller

The proportional-derivative (PD) controller is one of the most widely used controller in the current industries due to its simple design, high computing efficiency and low cost. However as found in the early works, this method should not be applied directly to time-delayed bilateral teleoperation as it can destabilize the system [1].

In this simulation, the bilateral teleopeartion set-up is done as in figure 4.5.

The PD controller itself is detailed in figure 4.6, where the PD gains are both chosen to be 100. This value is tested with the same set-up without time delay and proved itself stable in that case. This ensures that the instability is caused by the time delay and not the PD gains.

The equation characterizing this controller is shown in equation 4.2 below:

$$\tau_{sc} = K_1(q_{sd} - q_s) + K_2(\dot{q}_{sd} - \dot{q}_s) \tag{4.2}$$

where K_1 and K_2 are the PD gains, q_s is the actual slave robot position, $\dot{q}_s$ is the actual slave robot velocity, q_{sd} is the desired slave robot position (which is the delayed master

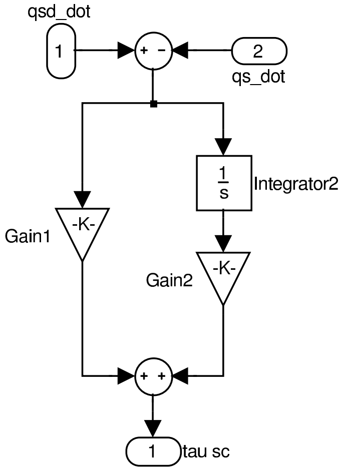


Figure 4.6: PD controller modelling

robot position) and $\dot{q}_{sd}$ is the desired slave robot velocity (which is the delayed master robot velocity).

4.3 Wave Variable Controller

As discussed in the literature review covered by chapter 2, the wave variable controller is considered as a major breakthrough by many researchers. It transformed the traditionally used energy variables $\dot{x}$ and F to wave variables u and v and satisfied the passivity condition. The simulation for such controller is set up as in figure 4.7. Built on the basic PD controller discussed in the previous section, it basically adds the wave transformation right before and right after the communication channel in such a way that only wave variables are being transmitted [55]. The wave variable controller is based on the PD controller with wave transformations as follow:

$$u = \frac{b\dot{x} + F}{\sqrt{2b}}, \qquad v = \frac{b\dot{x} - F}{\sqrt{2b}} \tag{4.3}$$

It should be noted that whenever used in the simulation, the values x and F are carefully translated from q and τ.

With matched impedance to avoid wave reflections, modifications were introduced to equations 4.3 as recommended in [55]. The wave transformation on the master side is shown in figure 4.8 and the one on the slave side is shown in figure 4.9. This wave variable controller is impedance matched and the variable b which refers to the wave impedance is set to 1.

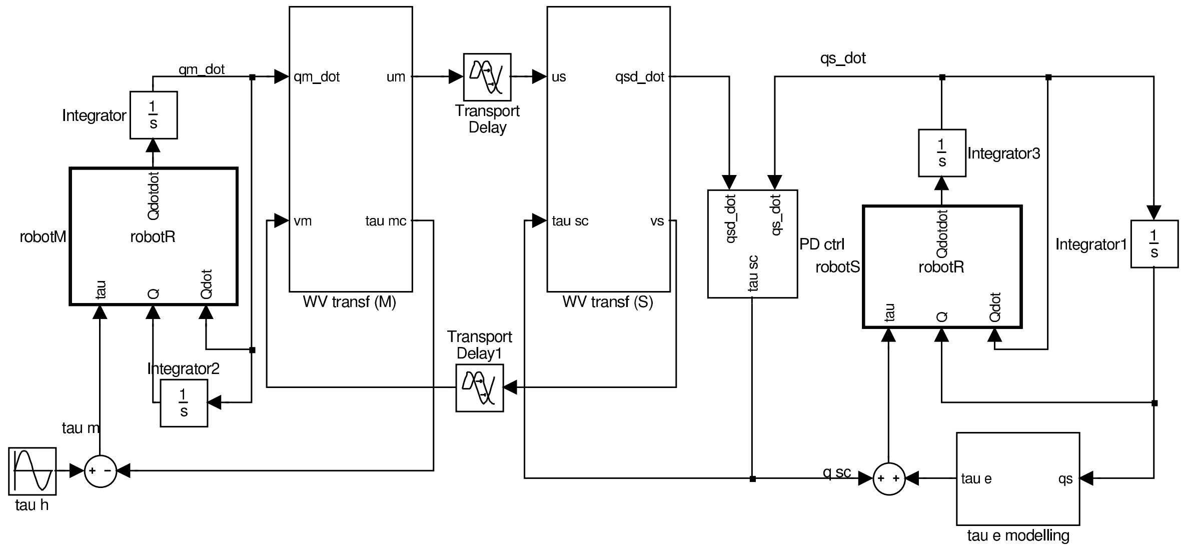


Figure 4.7: Bilateral teleoperation setup with wave variables controller

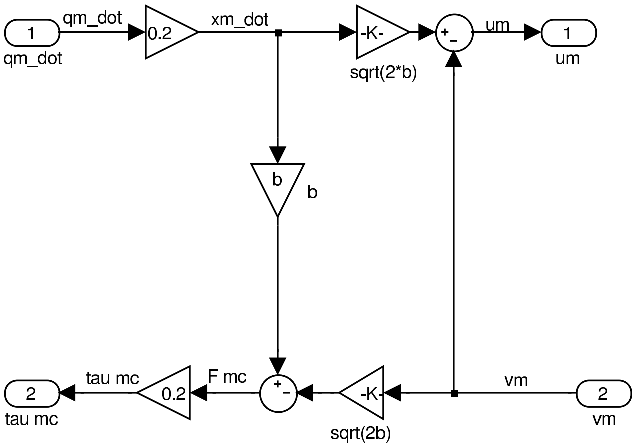


Figure 4.8: Wave transformation on the master side

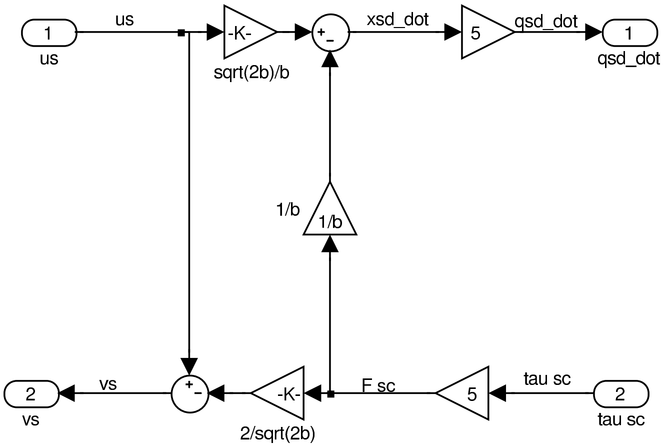


Figure 4.9: Wave transformation on the slave side

4.4 Second Order Sliding-Mode Controller

The observer based sliding-mode controller was proposed in [27] and was found to have a good performance in soft environments. The paper proposed the use of an observer to eliminate any measurement of velocity and acceleration signals. However since the use of such observers may introduce unwanted errors not present in the other controllers being looked at in the text, they are not used in the current simulations. Only the second order sliding-mode controller is investigated. It uses the concept of desired impedance to adjust parameters such as desired inertia, desired damping coefficient and desired stiffness. It should be noted that the original controller was built based on Cartesian coordinates and not joint coordinates as used in this text. For the sake of comparison, it has been translated into the simulation environment in the best possible way that was allowed.

The dynamic equations as used in the paper are:

$$M_m \ddot{x}_m + B_m \dot{x}_m = F_h + F_{mc} \tag{4.4}$$

$$M_s \ddot{x}_s + B_s \dot{x}_s = F_{sc} - F_e \tag{4.5}$$

where M_i and B_i respectively denote mass and viscious friction coefficient, with $i = m, s$ representing master and slave robot, in that order. As the notations used throughout this text, F_h is the input human force, F_e is the external force caused by the contact with the remote object, F_{mc} is the force outputted by the master controller and F_{sc} the force produced by the slave controller.

In order to respect the original design assumptions used in the paper, F_e is kept as a positive term. The robot dynamics equation 4.5 uses a negative sign externally prior to F_e to indicate that it is a force going against the motion of the robot. However in order to keep the consistency in the graph presentations, the values of F_e are plotted as negative in the force feedback graphs, as it is done with the results of the three other controllers.

The master side controller(F_{mc}) and the slave side controller(F_{sc}) are described respectively by the following equations:

$$F_{mc} = -F_h + B_m\dot{x}_m + \frac{M_m}{\bar{M}_m} \times (F_h - k_f F_e^{dy} - \bar{B}_m\dot{x}_m - \bar{K}_m x_m) \tag{4.6}$$

$$\begin{aligned} F_{sc} = & -\frac{M_s}{\bar{M}_s}(\bar{B}_s\dot{\tilde{x}}_s + \bar{K}_s\tilde{x} + F_e + K_i\int_0^t sign(I_e(\tau))d\tau) \\ & +\frac{M_s}{\bar{M}_m}k_p(F_h^{dy} - k_f F_e^{ddy} - \bar{B}_m\dot{x}_m^{dy} - \bar{K}_m x_m^{dy}) \\ & +F_e + B_s\dot{x}_s - K_g\Omega \end{aligned} \tag{4.7}$$

where M_i, B_i, K_i represent mass, viscous friction coefficient, and stiffness respectively, with $i = m, s$ denoting master and slave, respectively. The bar on any of those parameters indicates that it is a desired value which is pre-set and fixed. $\tilde{x}$ is the position difference between the actual slave robot position and the delayed master robot position. Subscript dy indicates that the term is delayed by a single trip time delay while ddy indicates a round-trip delay. k_p and k_f are scaling factors which are both set to 1. $K_g > 0$ is a positive parameter set to 25, $K_i > 0$ is the sliding mode gain set to 0.1, I_e is the sliding surface which should be equal to 0 ideally:

$$I_e = \bar{M}\ddot{\tilde{x}}_s + \bar{B}_s\dot{\tilde{x}} + \bar{K}_s\tilde{x} + F_e = 0 \tag{4.8}$$

and Ω is the extended error variable defined by

$$\Omega = \frac{1}{\bar{M}_s}[\int_0^t I_e(\tau)d\tau + K_i\int_0^t\int_0^\sigma sign(I_e(\tau))d\tau d\sigma] \tag{4.9}$$

This controller set-up is shown in figure 4.10. Its parameters are set based on the parameters given in the original text [27]:

$M_m = 0.1$ $B_m = 0$
$M_s = 0.1$ $B_s = 0$
$\bar{M}_m = 0.1$ $\bar{B}_m = 0.15$ $\bar{K}_m = 0$
$\bar{M}_s = 0.1$ $\bar{B}_s = 0.105$ $\bar{K}_s = 1.5$

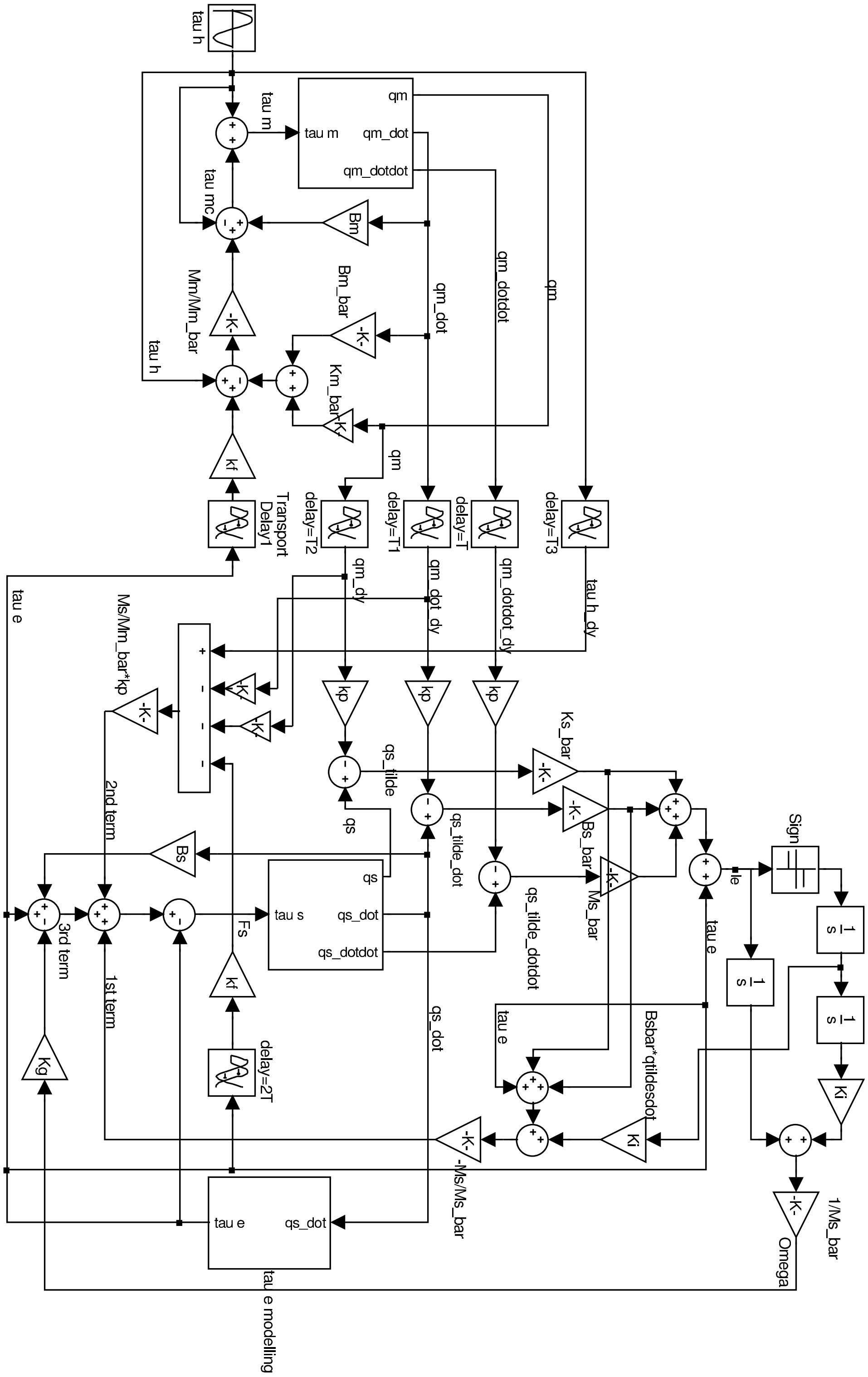


Figure 4.10: Bilateral teleoperation setup with sliding-mode controller

4.5 Adaptive Fuzzy Logic Controller

Last but not least the adaptive fuzzy logic controller (AFLC) is examined. The design along with the prove are shown in detail in chapter 3. The slave side controller is described as follow:

$$\tau_{sc} = \hat{D}_s\ddot{q}_{s_r} + \hat{C}_s\dot{q}_{s_r} + \hat{G}_s - F_e - K_{Ds}s + \hat{F}(\ddot{q}_{s_r}, \dot{q}_{s_r}, q_s, \dot{q}_s, \ddot{q}_s|\theta) \tag{4.10}$$

with the adaptive law

$$\dot{\theta} = -\Gamma_s^{-1}s_s^T\xi_s(\ddot{q}_{s_r}, \dot{q}_{s_r}, q_s, \dot{q}_s, \ddot{q}_s) \tag{4.11}$$

On the master side, the controller is presented as

$$\begin{aligned}\tau_{mc} = {} & \hat{D}_m\ddot{q}_{m_r} + \hat{C}_m\dot{q}_{m_r} + \hat{G}_m + F_e^{dy} - \hat{\hat{M}}_m\ddot{q}_m - \hat{\hat{B}}_m\dot{q}_m - \hat{\hat{G}}_m - K_{Dm}s \\ & + \hat{F}_1(\ddot{q}_{m_r}, \dot{q}_{m_r}, q_m, \dot{q}_m, \ddot{q}_m|\theta_1) + \hat{F}_2(\ddot{q}_m, \dot{q}_m, q_m|\theta_2)\end{aligned} \tag{4.12}$$

with adaptive laws

$$\dot{\theta}_1 = -\Gamma_{s1}^{-1}s_m^T\xi_{m1}(\ddot{q}_{m_r}, \dot{q}_{m_r}, q_m, \dot{q}_m, \ddot{q}_m) \tag{4.13}$$

$$\dot{\theta}_2 = -\Gamma_{s2}^{-1}s_m^T\xi_{m2}(q_m, \dot{q}_m, \ddot{q}_m) \tag{4.14}$$

where all the terms were explained in detail in chapter 3.

The simulation's overall setup is shown in figure 4.11, with the master controller detailed in figure 4.12 and the slave controller in figure 4.13. In order to keep the rule base to a reasonable size (hence keeping the needed computational power relatively low), the AFLCs are broken down into two-input-AFLCs at all times. Each of the individual two-input-AFLC is shown in figure 4.14. They all have an array of numbers as the centres of the output fuzzy membership function at the starting point. The coding used to update them is shown in appendix C. The parameters used for this simulation are:

$\hat{D}_m = \hat{D}_s = 0$

$\hat{C}_m = \hat{C}_s = 0$

$\hat{G}_m = \hat{G}_s = 0$

$\hat{\hat{M}}_m = 0$

$\hat{\hat{B}}_m = 0$

$\hat{\hat{G}}_m = 0$

$\Lambda = 2$

$K_{Dm} = K_{Ds} = 2$

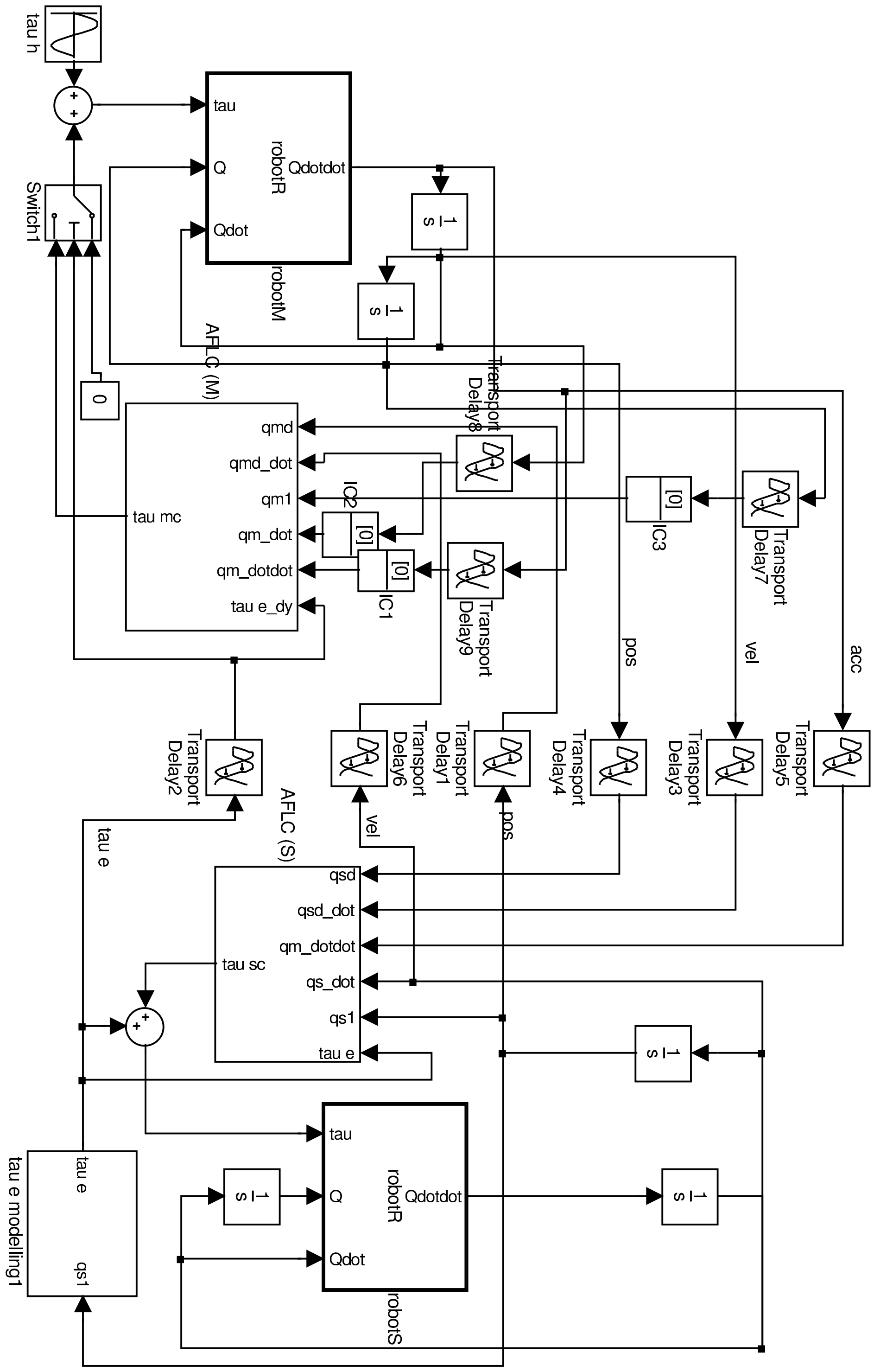


Figure 4.11: Bilateral teleoperation setup with AFLC

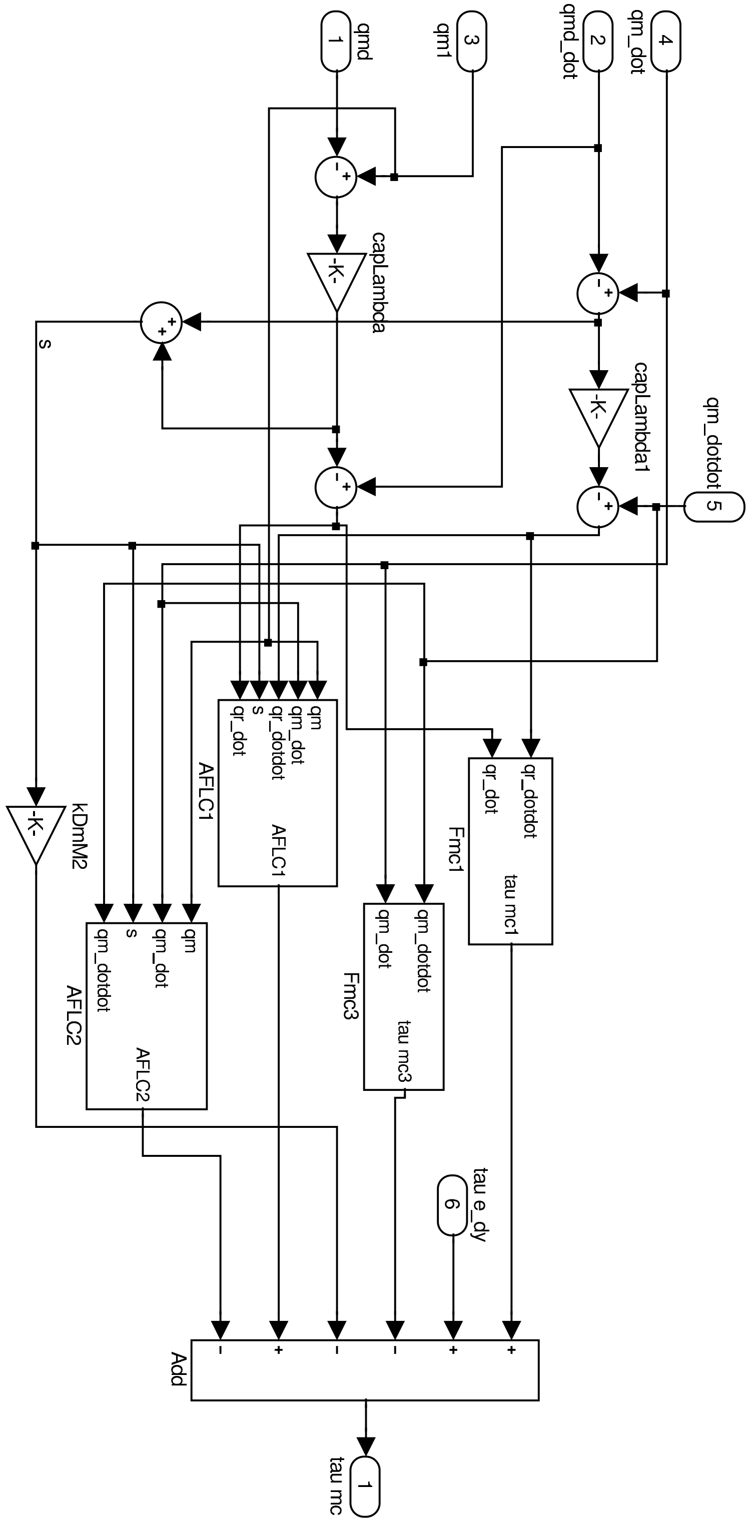


Figure 4.12: Master side AFLC

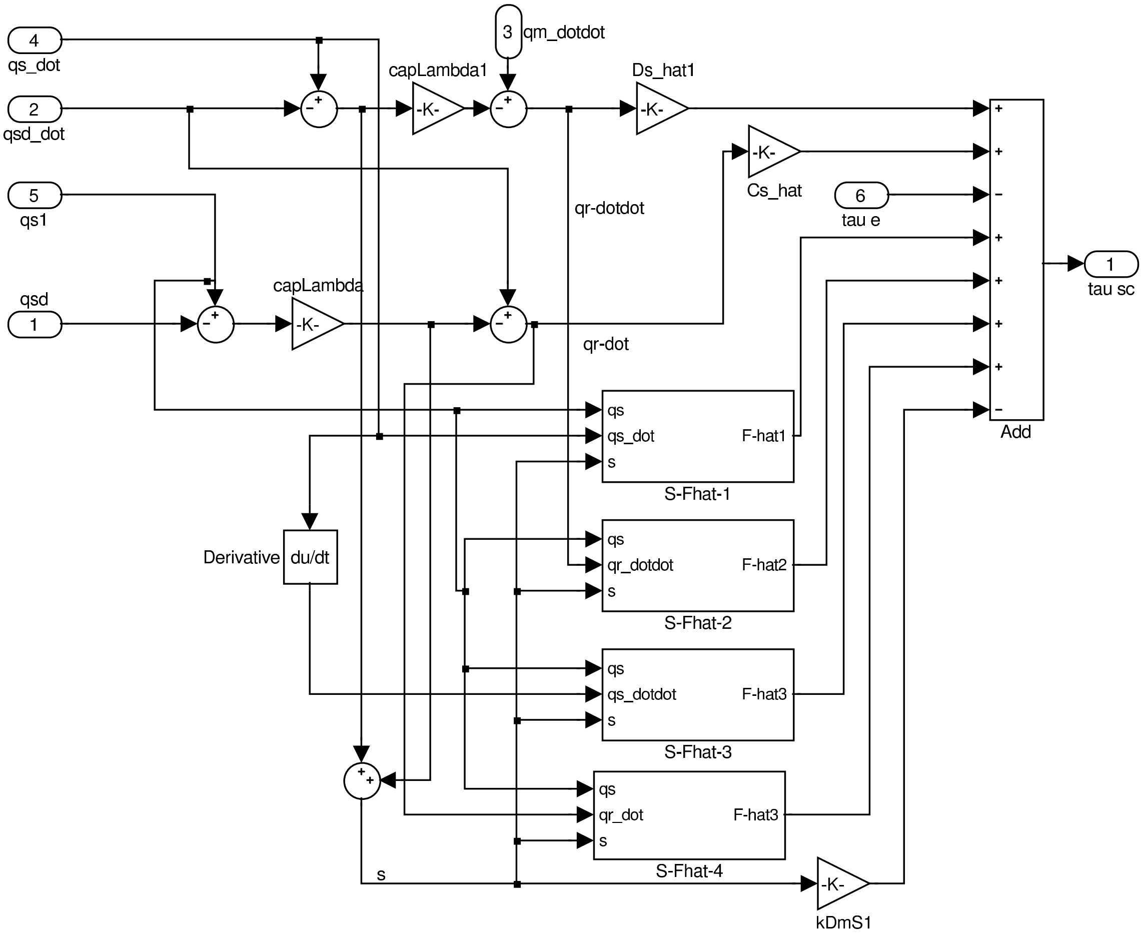


Figure 4.13: Slave side AFLC

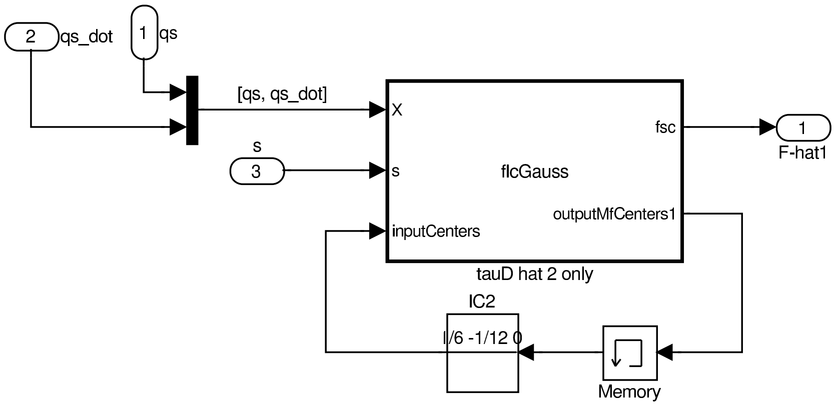


Figure 4.14: Two-input-AFLC

4.6 Summary

As shown in this chapter, the simulation environment and set-ups are presented. The input force, robot system, time delay value and remote objects are the same for all the controllers to be tested against. The four controllers examined are: proportional-derivative (PD) controller, wave-variable controller, second-order sliding-mode controller and adaptive fuzzy logic controller. In the following chapter, the resulting graphs of the simulation will be presented and analysed in detail.

Chapter 5

Results and Discussion

From the simulation set-ups presented in chapter 4, the resulting graphs will be shown and discussed in this chapter. As mentioned previously, each controller is tested against three different contact situations (none, hard contact and soft contact, always in this order). For each scenario, two graphs will be presented: a top graph will show position tracking and will give an idea about how well the slave robot is following the master robot; a bottom graph will demonstrate torque feedback results and will illustrate how well the master controller is reflecting the remote site contact force to the human operator. On the top graph, qm and qs will be used to designate the joint position of master and slave robot respectively, both measured in radian (rad). On the bottom graph, τ_e will refer to the external torque while τ_{mc} will be the torque to be felt at the master side which is generated by the controller, both measured in Newton-meter (N m).

Due to the difficulty of modelling the human input force to counteract any action upon contact (i.e.: to stop the robot, or to apply a smaller force), it is considered normal for the robot to be sent back far away by the contact force, especially in the hard contact cases where the environment stiffness is higher. The main focus here will be on how well the contact force is being reflected and not on how smooth the robot's moving curve is overall.

5.1 Basic Proportional-Derivative (PD) controller

The simulation graphs of the bilateral teleoperation set-up with PD controller will be shown in this section. The system is first tested with no time delay and the resulting curves are presented in figure 5.1. Then it is tested with a delay of 0.25 s each way against the three contact situations. The outcome is illustrated in figures 5.2–5.4.

All the results are as expected based on previous works done by other researchers [1].

Figure 5.1 shows the results for free space simulation without time delay. The system seems to be stable and transparent. Note that the oscillations in the curve representing the torque τ_{mc} are normal as they reflect slave robot's mass to the human operator. This was the way this controller was set up and designed.

Figure 5.2 shows the results for free space simulation with time delay, where no contact occurred. It can be observed that the controller quickly becomes unstable.

Figure 5.3 shows the results for hard contact simulation with the environment having a stiffness coefficient of $450\,\mathrm{N\,m^{-1}}$ or $18\,\mathrm{N\,m\,rad^{-1}}$. It can be seen from top and bottom graphs that the outputs are slowly growing out of bound for both position and force tracking, indicating the controller's instability upon hard contact.

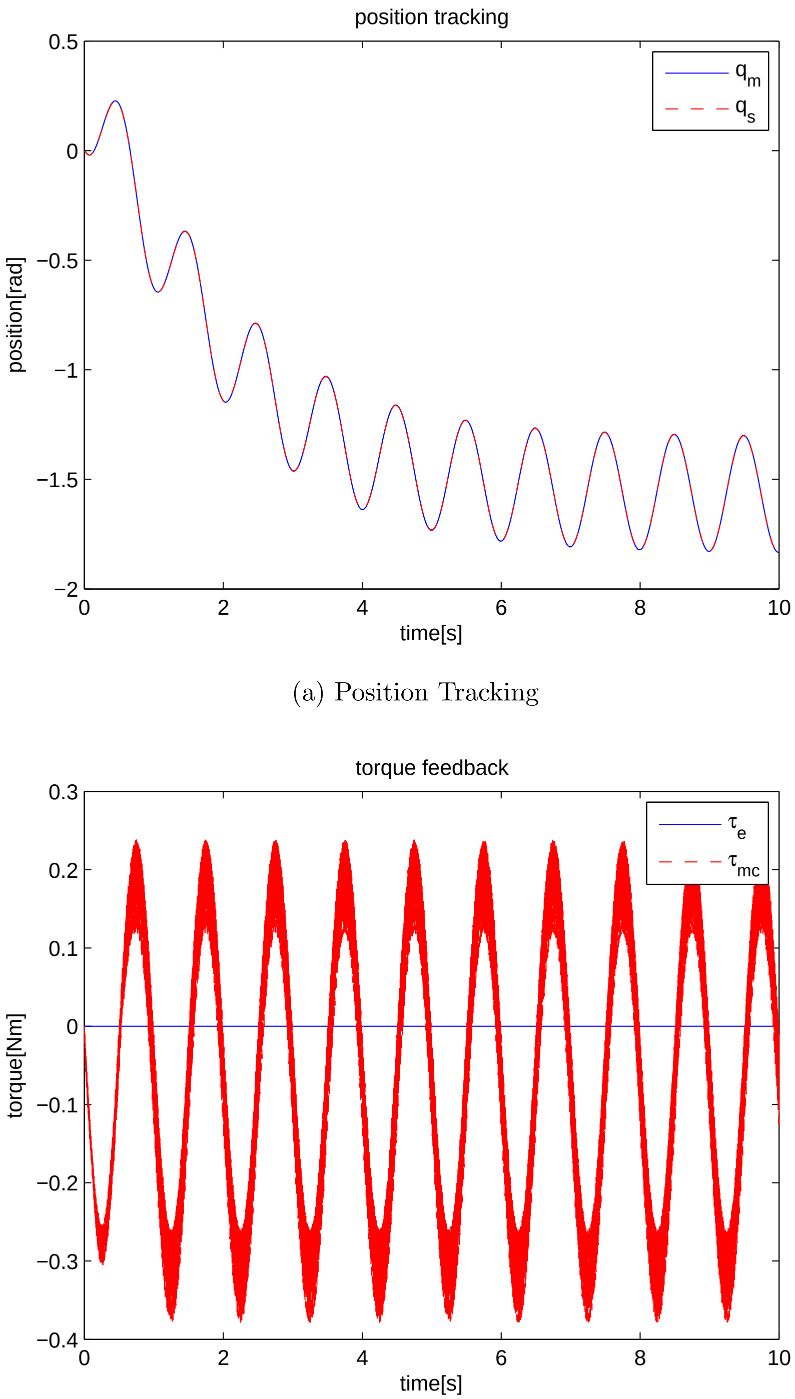
Figure 5.4 shows the results for soft contact simulation with the environment having a stiffness coefficient of $45\,\mathrm{N\,m^{-1}}$ or $1.8\,\mathrm{N\,m\,rad^{-1}}$. The controller still shows instability, but overall it grows out of bound at a slower rate.

Comparing the results, it can be observed that the PD controller caused unstable behaviour to the robots in all time-delayed situations. This can be seen through both position tracking graphs and force tracking graphs. From another point of view, this is a good indication that the environment simulation is well done as the same behaviour was recorded in prior works [1].

Contact on hard environment and soft environment outputted somewhat different position tracking graphs but both are unstable. Observing the output graph of hard environment contact suggests that the robot was sent back quickly by the contact force and then it slowly travels further and further as time goes. The contact force was much smaller in the case of the soft environment, therefore the robot was not sent back as far and its behaviour is more similar to the free space position tracking curve.

Force feedback tracking seems satisfactory for both hard and soft contact environments when the contact was first made. However in both cases, the oscillations which followed are growing higher and higher, indicating that the system is unstable. Again, the soft environment graph shows a slower rate of growth.

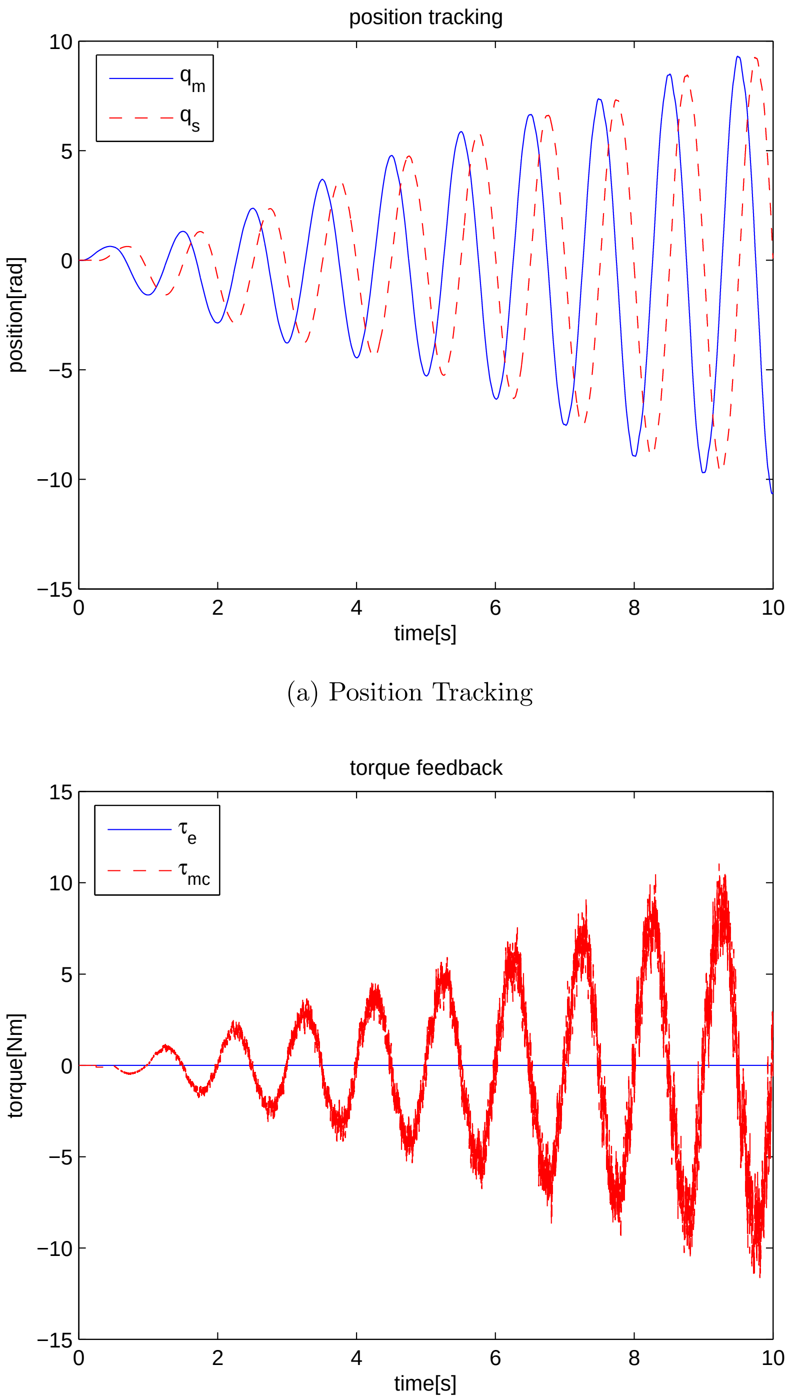
It should be noted that the oscillating forces in the force feedback graphs indicate that the human operator is able to feel the mass of the slave robot (with a time delay). To make the human operator feel this mass or not is entirely to the designer's discretion. Not incorporating this feeling indicates an attempt of making slave robot transparent to the operator, trying to make the human operator feel like he/she is interacting directly



(a) Position Tracking

(b) Torque Feedback

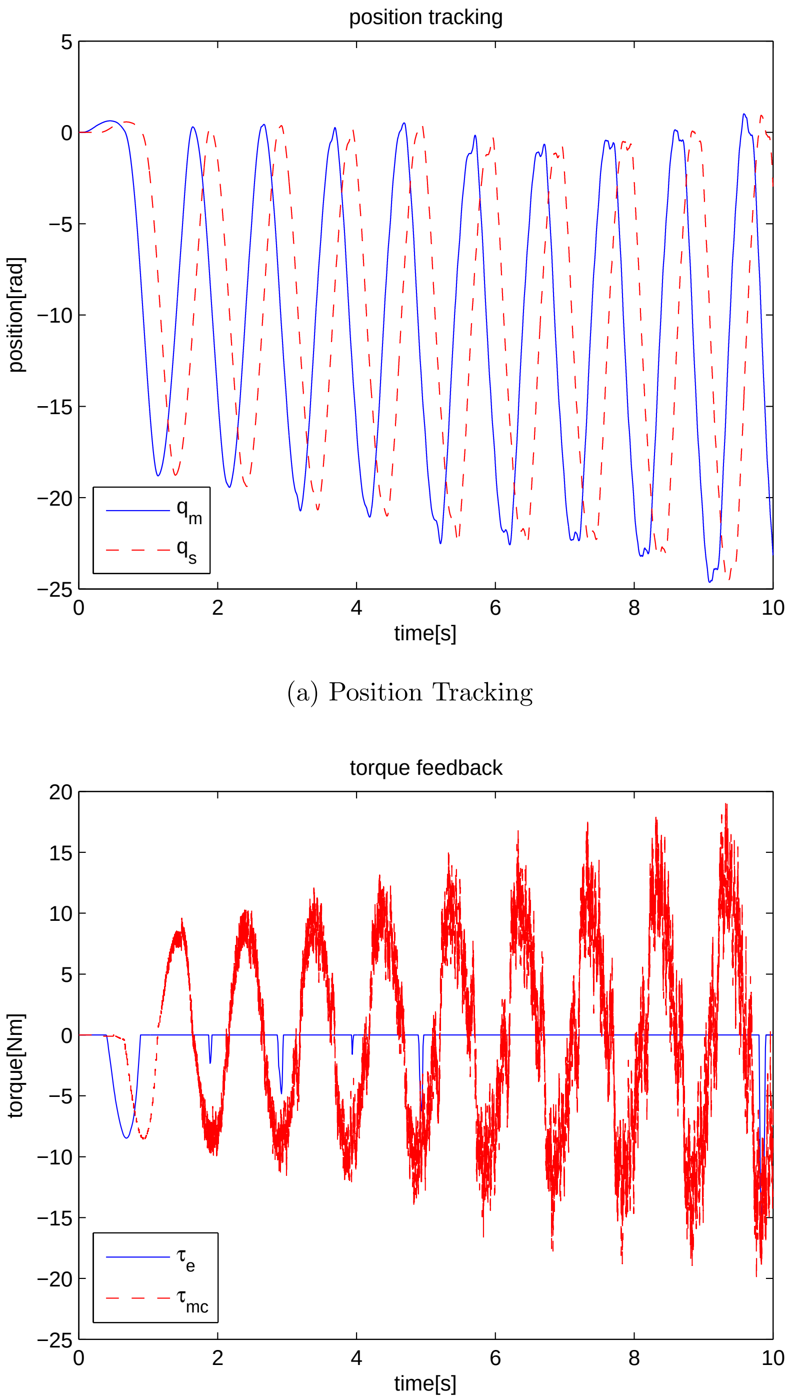
Figure 5.1: Simulation results using PD controller, no delay, no contact



(a) Position Tracking

(b) Torque Feedback

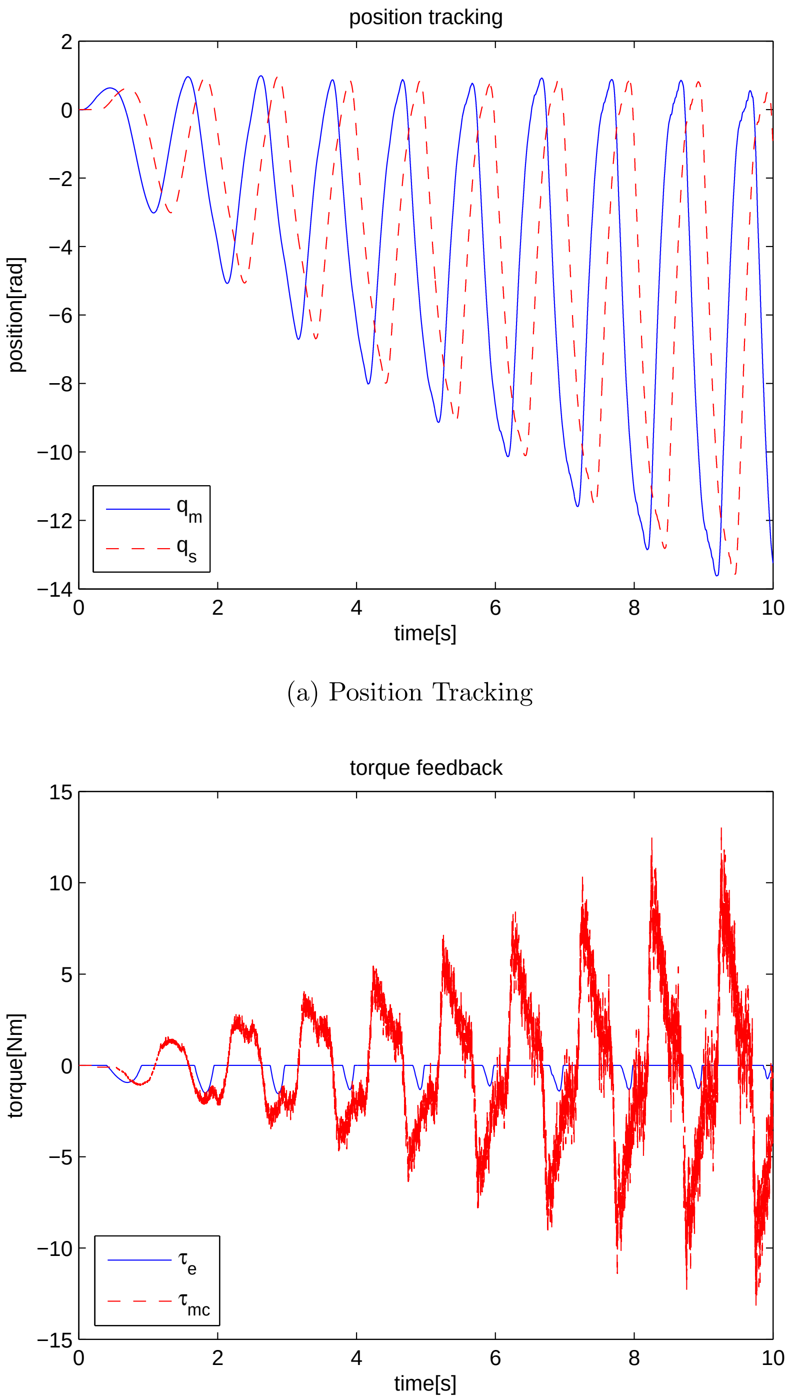
Figure 5.2: Simulation results using PD controller, free space



(a) Position Tracking

(b) Torque Feedback

Figure 5.3: Simulation results using PD controller, hard contact



(a) Position Tracking

(b) Torque Feedback

Figure 5.4: Simulation results using PD controller, soft contact

with the remote environment. Incorporating this feeling will make the operator aware of the presence of the slave robot by its mass. To make this difference on the design level, the force transferred from slave to master is usually different for those two cases. In the PD controller and the wave variable controller where slave robot is felt, the torque transferred back is τ_{sc}, which is the torque generated by the slave controller. For the sliding-mode controller and adaptive fuzzy logic controller (AFLC) where the slave robot should be transparent to the operator, the torque transferred back is τ_e which is the actual external torque felt at the remote site.

In summary, the PD controller seems to cause instability to the entire system, therefore it should be avoided at all cost for safety reasons.

5.2 Wave Variable Controller

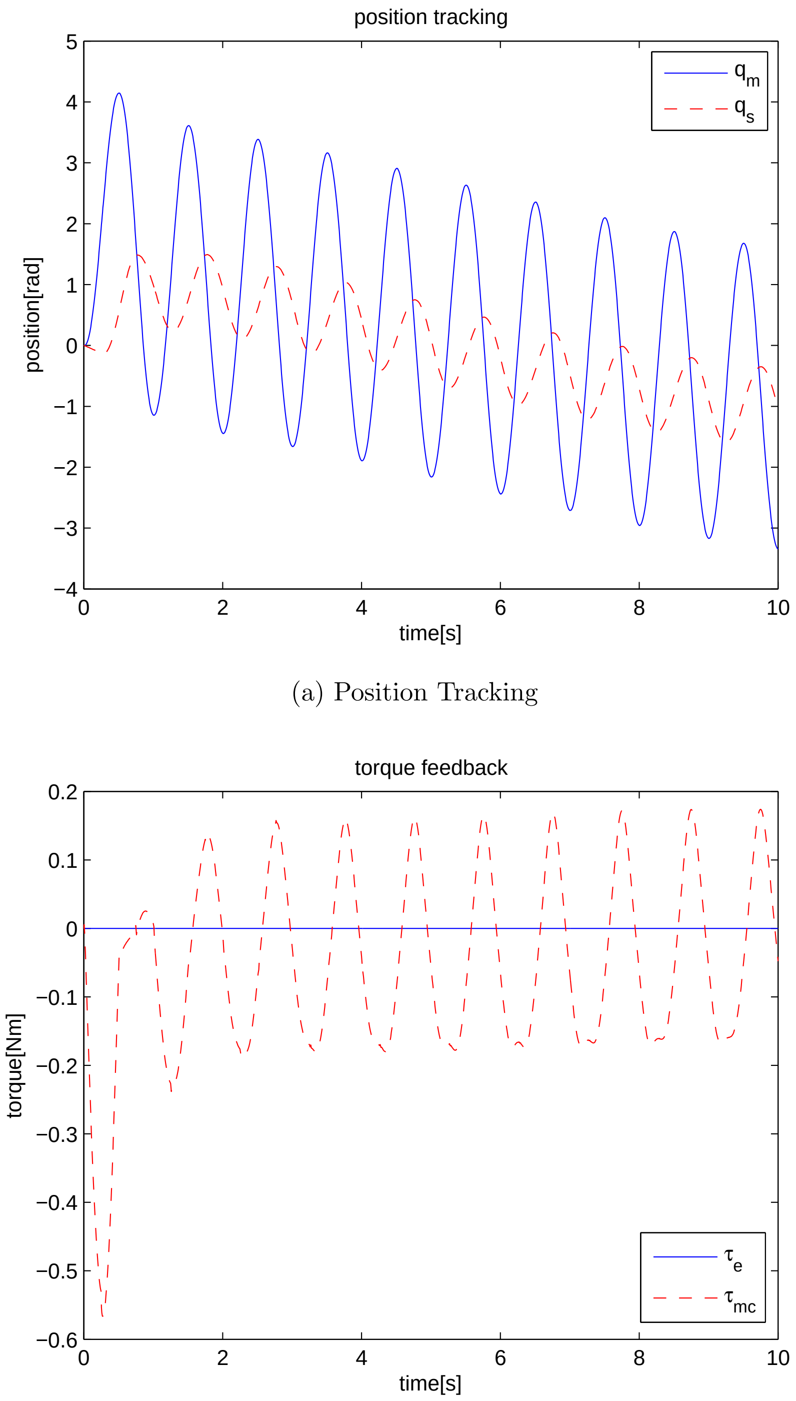
The outputted results using the wave variable controller in free space are given in figure 5.5. However as it is noticed that the slave robot does not reach the contact point of 0.1 rad in free space, the human input τ_h was slightly increased and the results tested in the same three environments are shown in figures 5.6–5.9[1]. As found in the previous works [1, 55], the controller ensures stability of the system, however the position tracking as well as the force feedback are both relatively unsatisfactory in all simulation results shown.

The use of wave variable controller very clearly eliminates the instability introduced by time delay in the bilateral teleoperation system. This is consistent for all situations.

In the no contact situation, it is true that both robots are going further away in distance once contact is made, however this phenomenon stops after a certain time period as shown in figure 5.7. This can be justified by the fact that since the weight of the robot is involved and the human input is a *sin* function, only when the robot arm is at $\pm 0.5\pi \pm 2n\pi$ rad or equivalent from its original position would it stay within two fixed joint position points (where $n = 1, 2, 3, ...$). Observing figure 5.7a, the joint position around which robot arm stopped going further is precisely at -0.5π rad. This is also the case for the other graphs.

For the position tracking, just as mentioned in the literature, this controller does not offer a very precise tracking performance. The slave robot is unable to follow master robot on the dot as can be seen from the graphs.

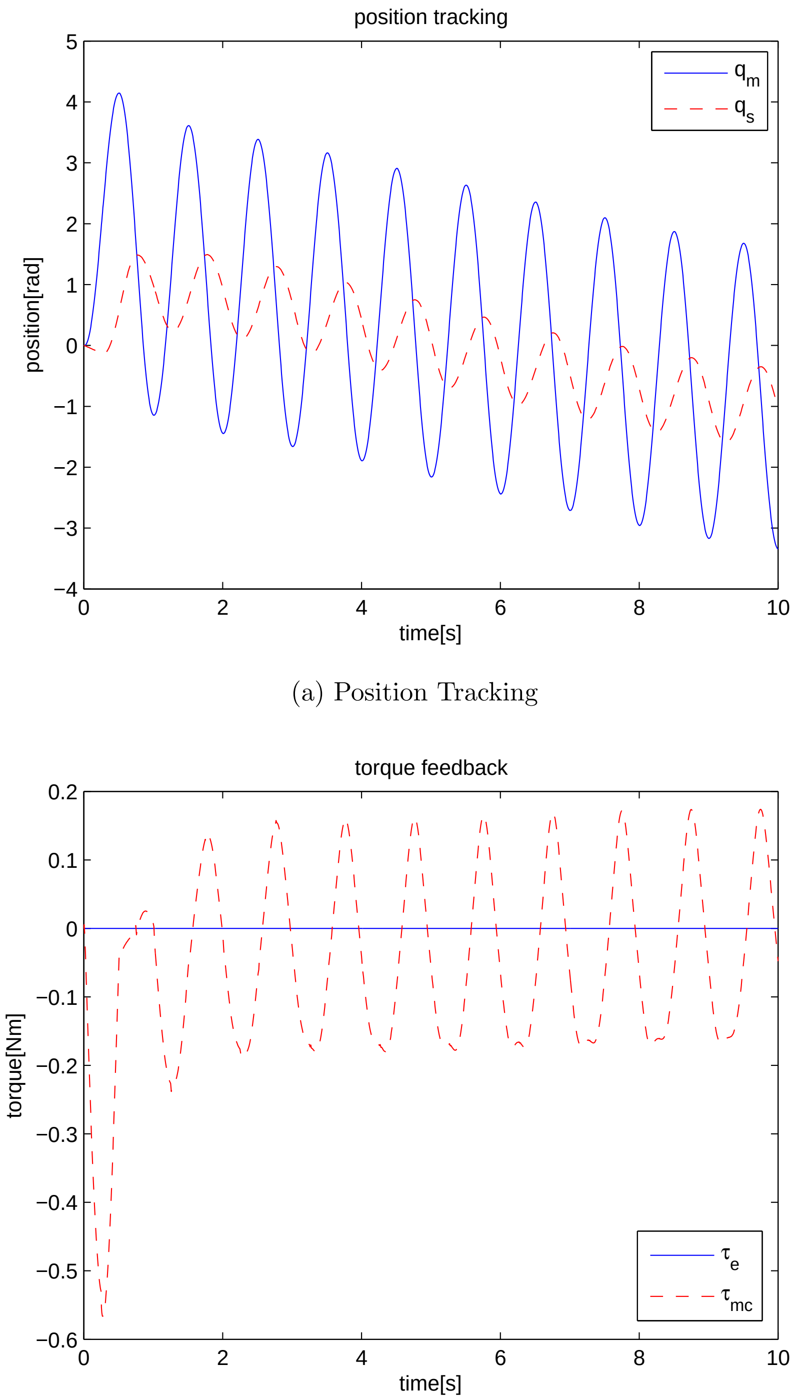
[1] The human input has been changed from the original $f(t) = 0.5sin(2\pi t)$ to $f(t) = 2.5sin(2\pi t)$, where $f(t)$ is the human input in N m and t is time in s.



(a) Position Tracking

(b) Torque Feedback

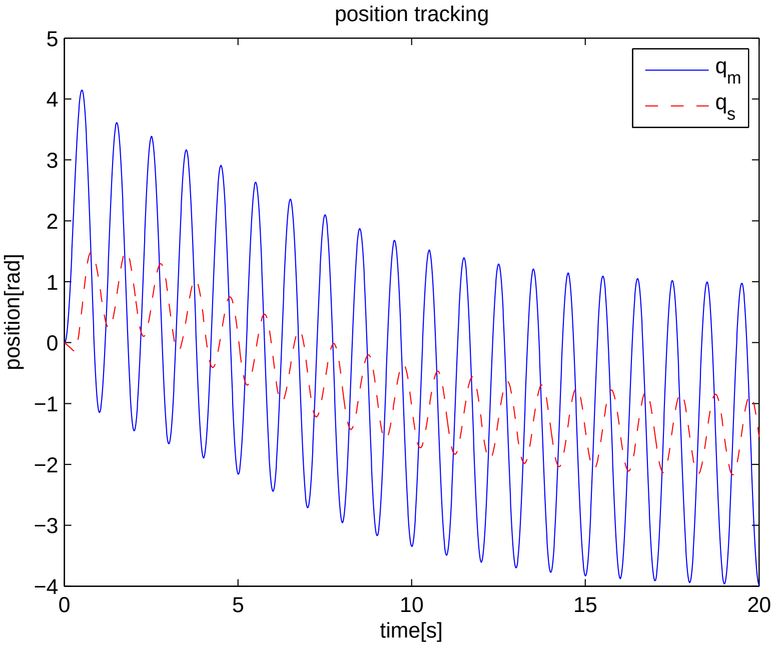
Figure 5.5: Simulation results using wave variable controller, free space



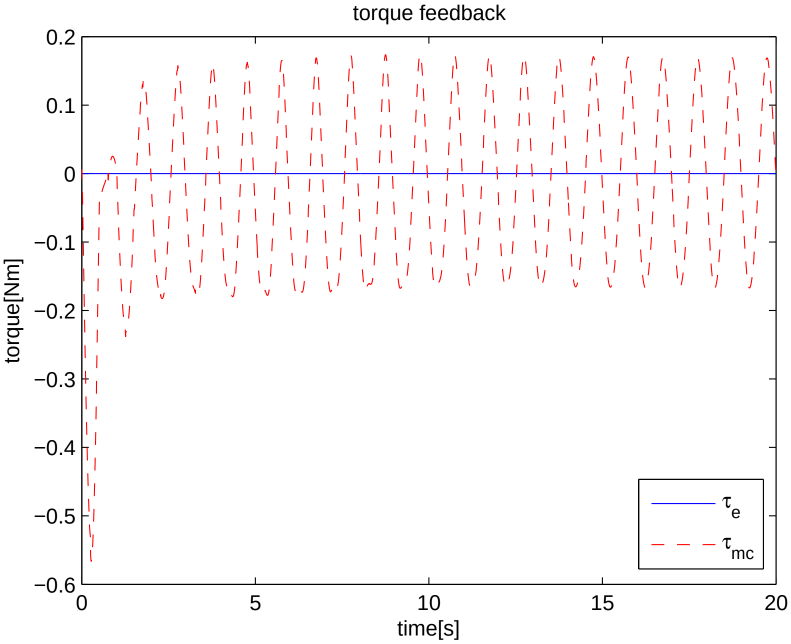
(a) Position Tracking

(b) Torque Feedback

Figure 5.6: Simulation results using WV controller, free space (enhanced τ_h)

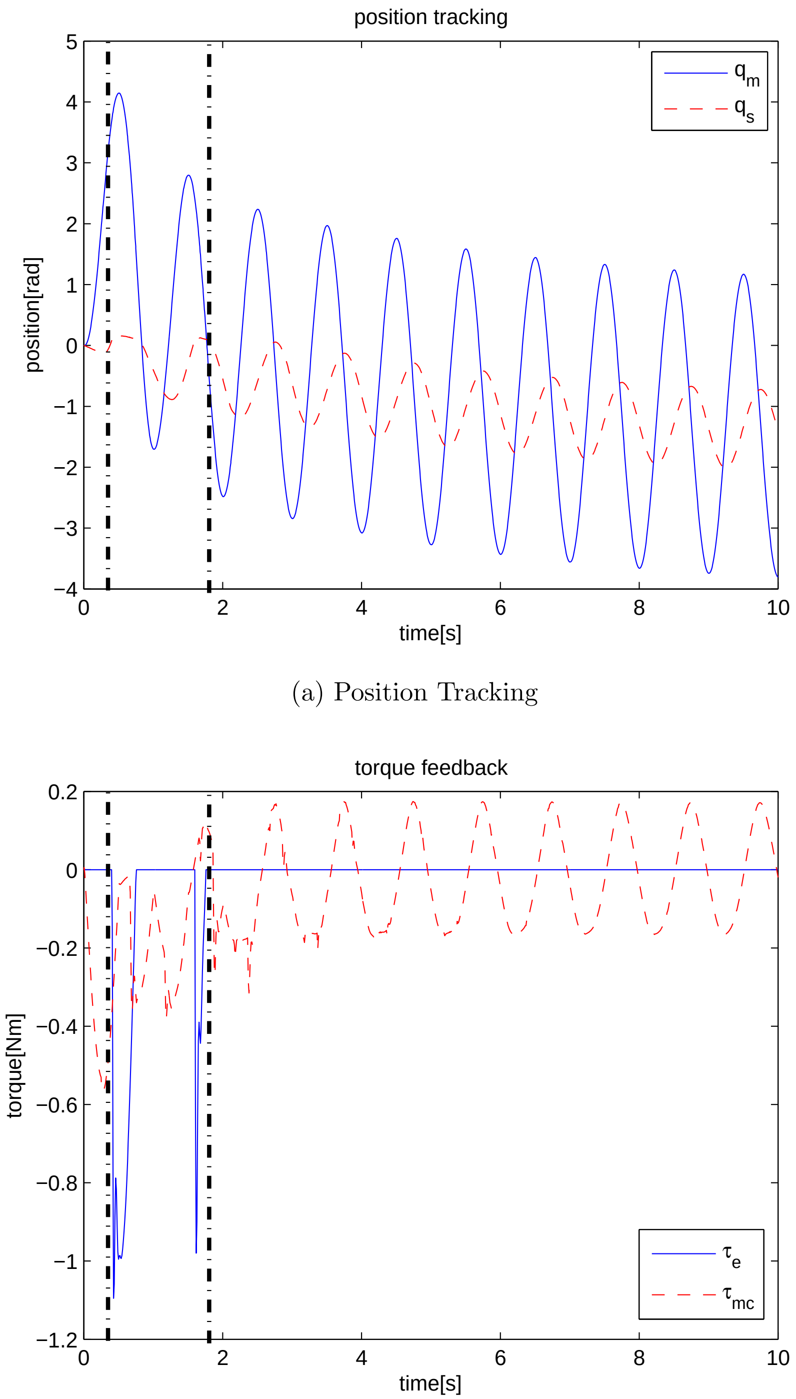


(a) Position Tracking



(b) Torque Feedback

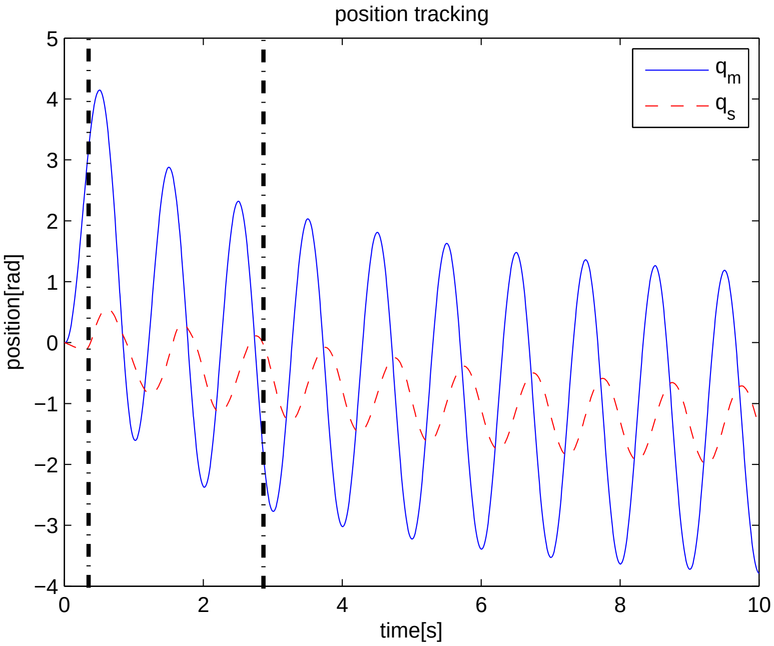
Figure 5.7: Simulation results using WV controller, free space (enhanced τ_h, longer)



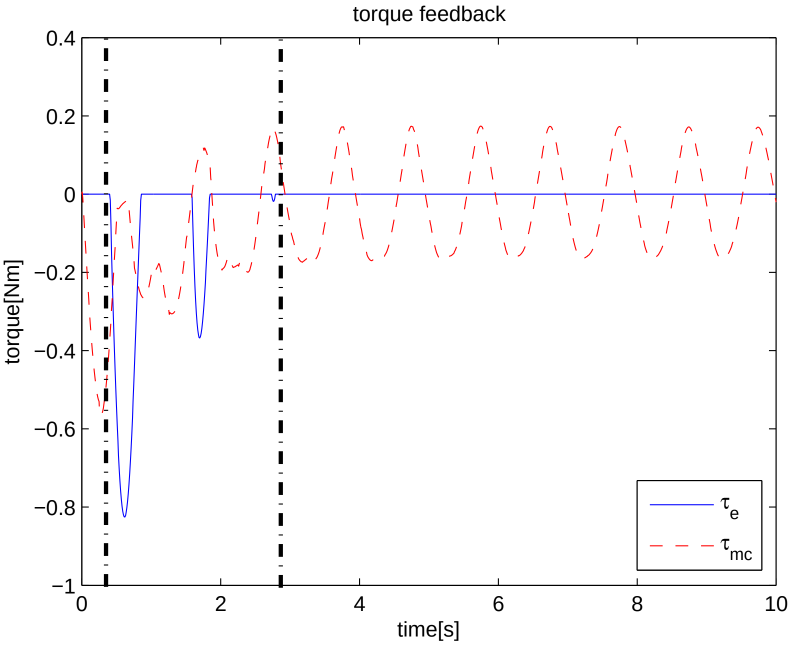
(a) Position Tracking

(b) Torque Feedback

Figure 5.8: Simulation results using WV controller, hard contact (enhanced τ_h)



(a) Position Tracking



(b) Torque Feedback

Figure 5.9: Simulation results using WV controller, soft contact (enhanced τ_h)

The force tracking performance as shown is also not very precise. Focusing on the beginning of the graphs where the contact occurred, there is always some important differences between the external force from the remote site and the force being fed back to the human operator.

In summary, the wave variable controller can guarantee system stability at the cost of system transparency. This was reflected in both position tracking and force feedback graphs.

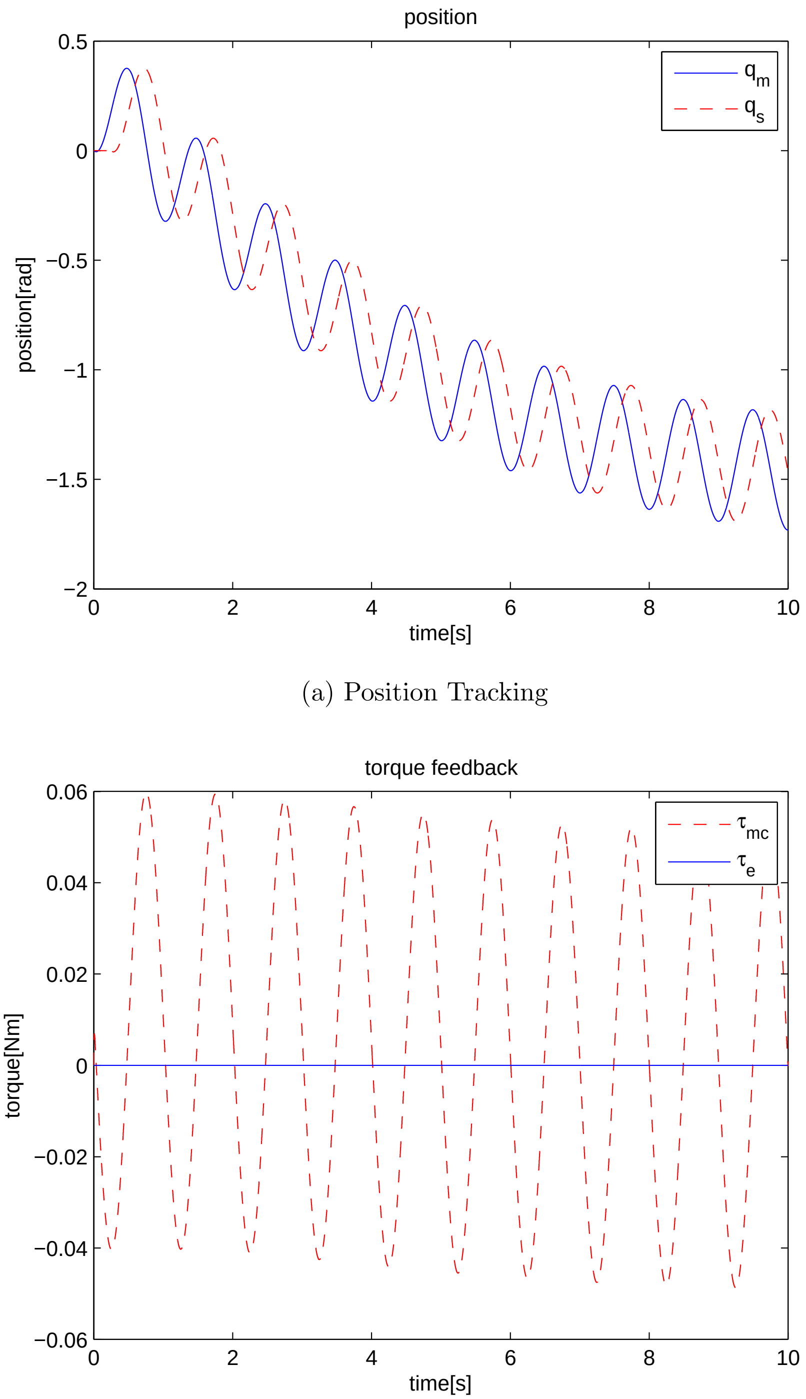
5.3 Second Order Sliding-Mode Controller

The simulation results using the second order sliding-mode controller are shown in figures 5.10–5.12. The dot-dash vertical lines in the figures indicate the area where contact occurred. It can be observed that in free space, the controller has a very good tracking performance for both position and force. But as the environment's stiffness coefficient increased, this performance decreased.

For position tracking, the second order sliding-mode controller is demonstrating an excellent performance in the no-contact situation. In the hard environment, quite some differences can be seen between master and slave robot when the contact occurred. But the difference was smaller in the soft contact environment. It should be noted that once the slave got pushed away by the force generated upon contact, it is going towards the stable joint position of 0.5π rad due to its weight as explained in the previous section. However, the difference in performance is greatly related to the fact that this controller has predefined variables for desired inertia, desired damping coefficient and desired stiffness coefficient for both master and slave robots. Those values were set for a specifc environment and need to be redefined to accommodate a different one each time.

For the force tracking performance, the results are similar. They are much better for the no contact situation while they decay as the environment got harder due to the same reason.

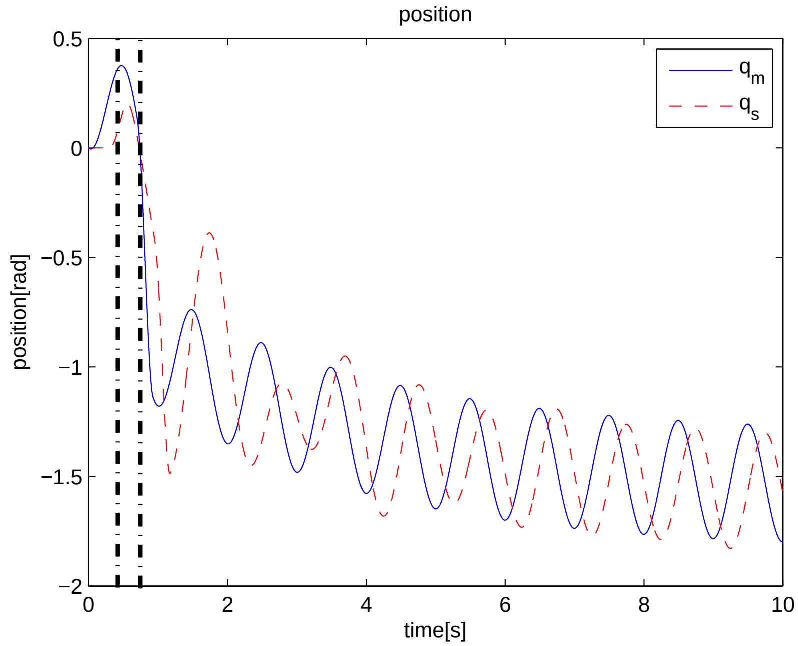
In summary, the stability and transparency of the second order sliding-mode controller seem to be heavily dependent on its pre-set parameters.



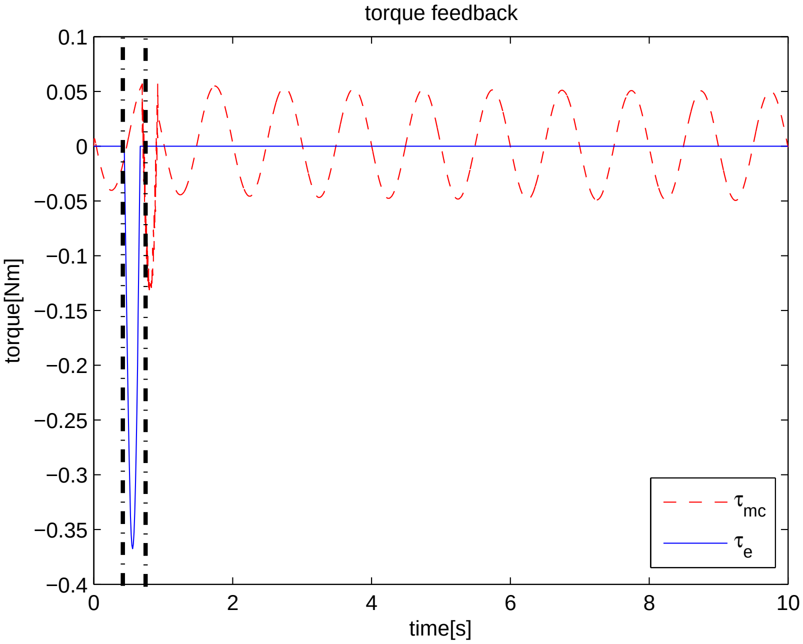
(a) Position Tracking

(b) Torque Feedback

Figure 5.10: Simulation results using sliding-mode controller, free space

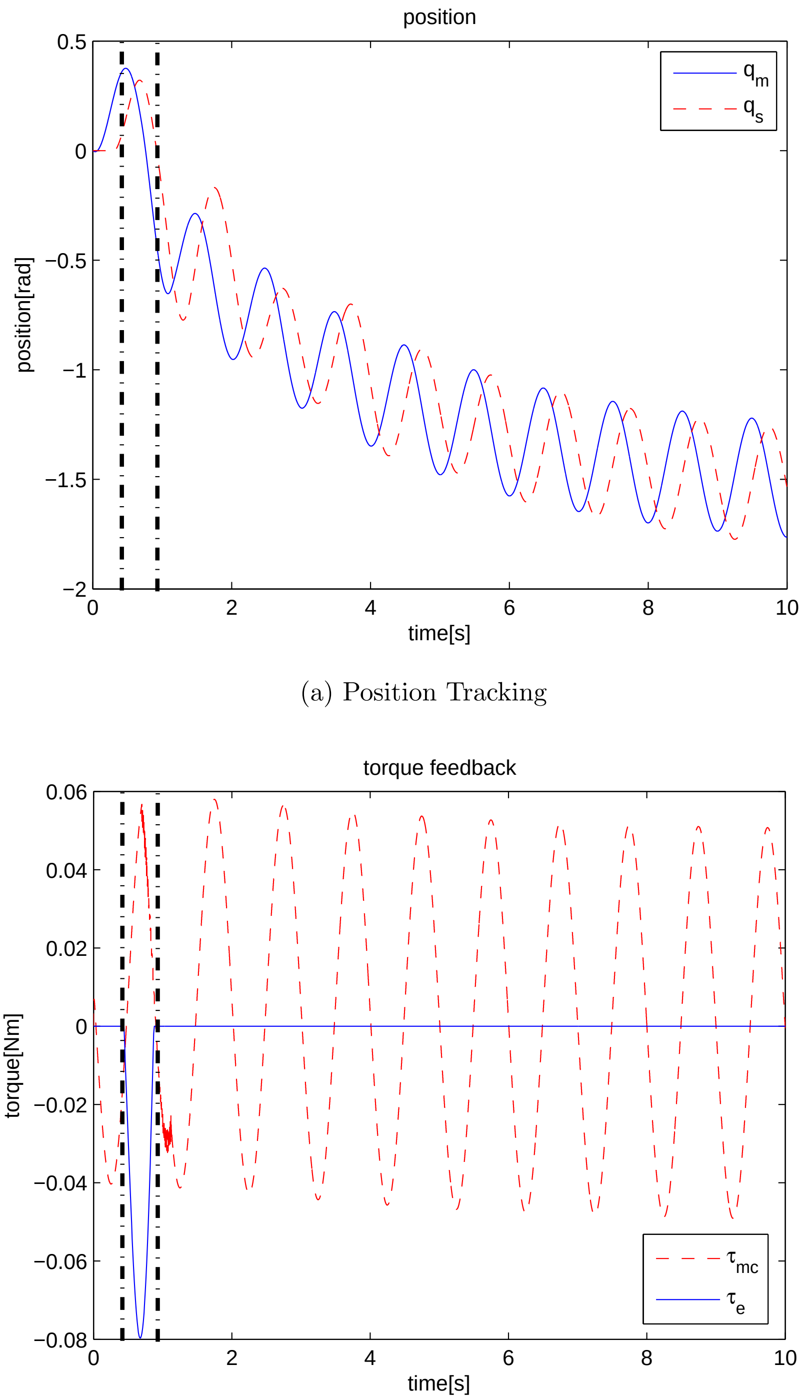


(a) Position Tracking



(b) Torque Feedback

Figure 5.11: Simulation results using sliding-mode controller, hard contact



(a) Position Tracking

(b) Torque Feedback

Figure 5.12: Simulation results using sliding-mode controller, soft contact

5.4 Adaptive Fuzzy Logic Controller

The results using the adaptive fuzzy logic controller(AFLC) are shown in figures 5.13–5.17. Again he dot-dash vertical lines in the figures indicate the area where contact occurred. The AFLC shows a fairly good response in most of the situations.

The position tracking is fairly precise throughout the different types of environments. The graphs show some slight position drift at the very beginning, but this is necessary for the training of the robot and to find the optimal parameters within the AFLC. The time it uses to accomplish this tuning is very short, therefore it is considered as acceptable. In the hard contact environment, the robot was pushed quite far by the contact force, but it eventually reached a stable point at -12.5π rad and stopped going further away. Therefore the controller is still stable.

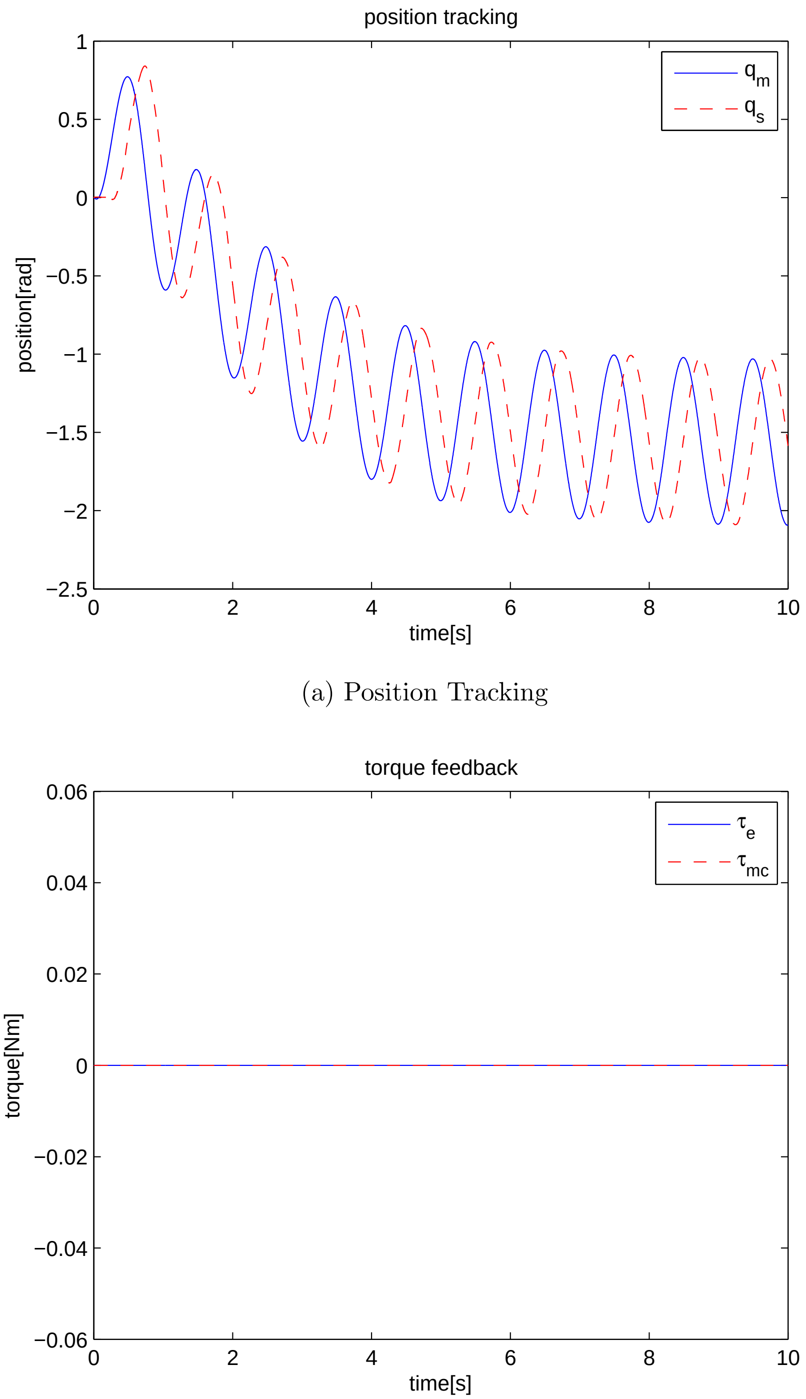
The force tracking performance is fairly satisfactory in the hard contact environment simulations as τ_e was faithfully reproduced by the AFLC. The constantly changing parameters within the controller are to be praised for being able to adapt to environments with different stiffness coefficients. In the soft contact environment, the differences between τ_e and τ_{mc} curves are expected to be ameliorated with further tuning of certain parameters specific to robot dynamics.

However, it should be noted that a precaution was used where if there was no collision detected between slave robot and environment, any output of the master side AFLC is reduced to zero. This step greatly helped for the seemingly good performance graphs. The simulation results using AFLC without this step are shown in figure 5.18 and figure 5.19, for hard contact and soft contact, respectively.

It can be seen that this step helped the elimination of instability coming from the AFLC. However since the controller was proven to be stable in chapter 3, the cause of this instability might have come from the setting of a few parameters within the controller itself which need to be fixed for specific robot parameters (such as arm length and mass). Further investigations would be needed in order to completely remove the instability even without the use of a switch block as well as the slight oscillations in the force tracking performance graph for soft environment.

The graphs show that even when both robots have been sent very far away from the original position by the contact force, the slave robot is following the master robot in a fairly satisfactory manner.

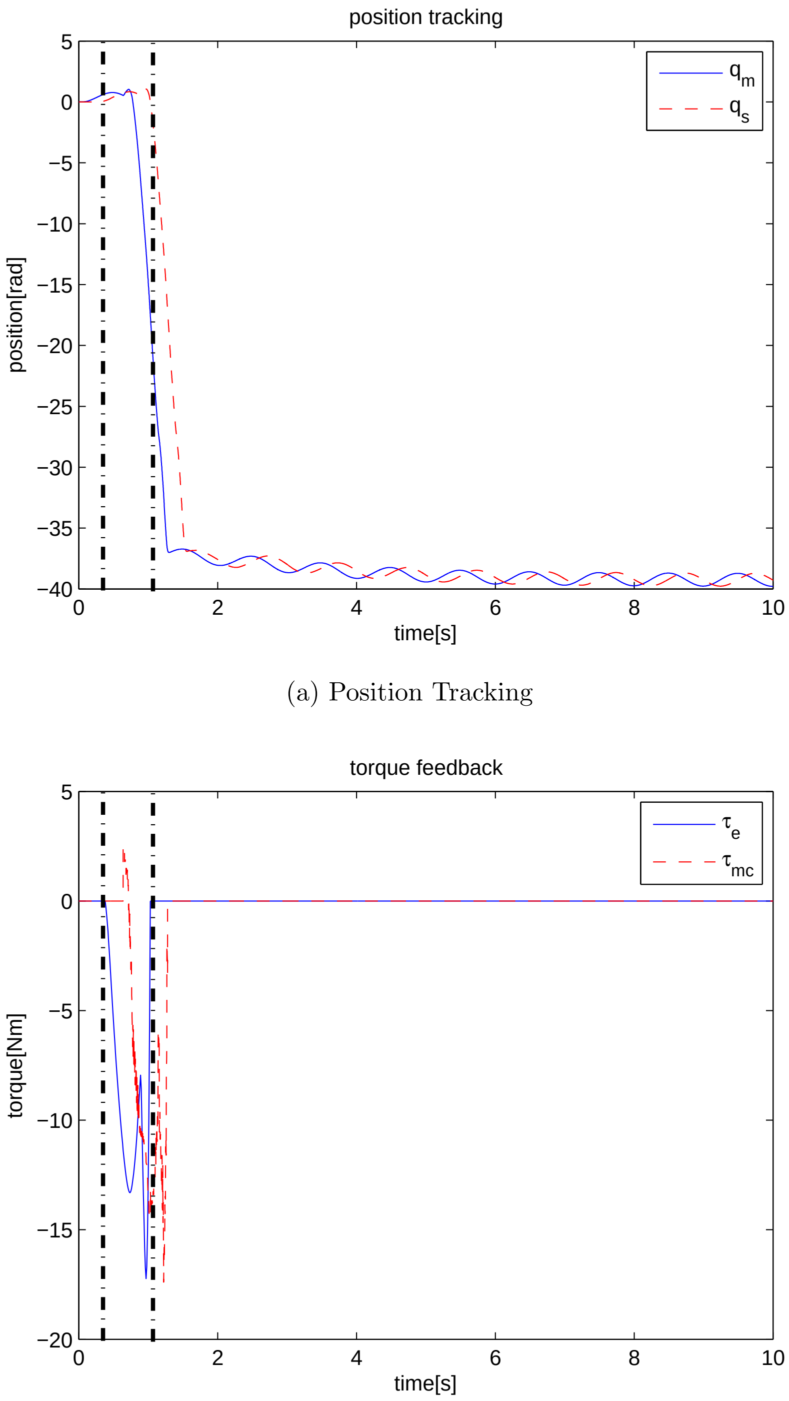
An important weakness of this controller is the extensive calculations needed. The simulations took a longer time to complete compared to the other controllers' simulation.



(a) Position Tracking

(b) Torque Feedback

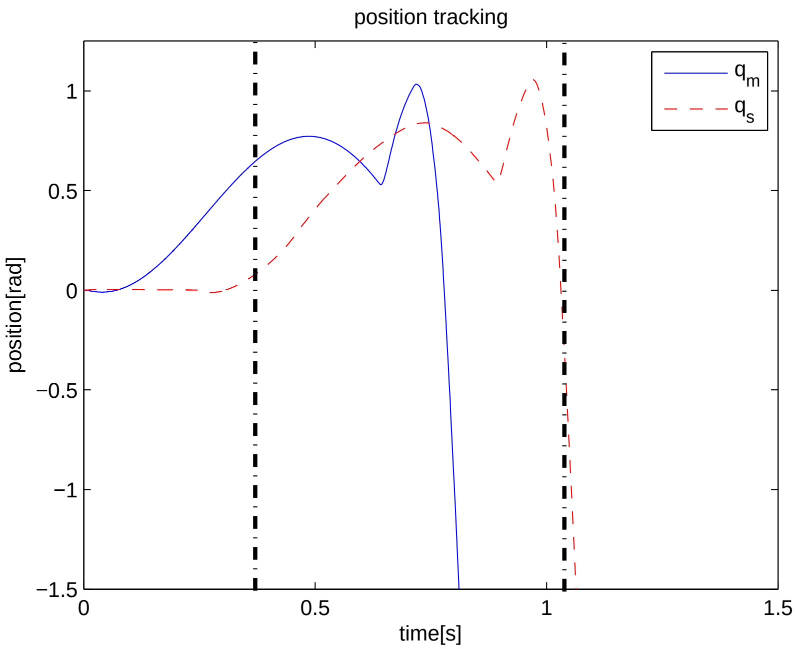
Figure 5.13: Simulation results using AFLC, free space



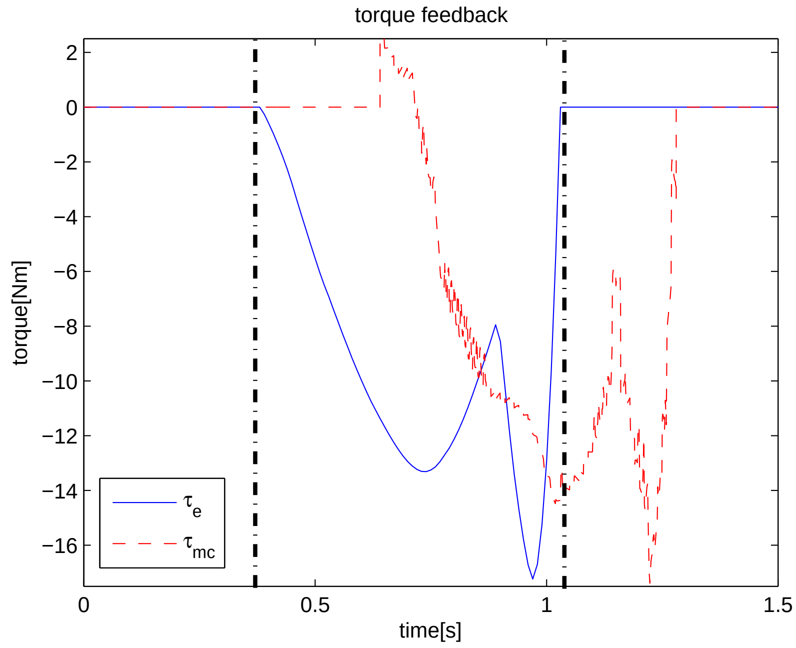
(a) Position Tracking

(b) Torque Feedback

Figure 5.14: Simulation results using AFLC, hard contact

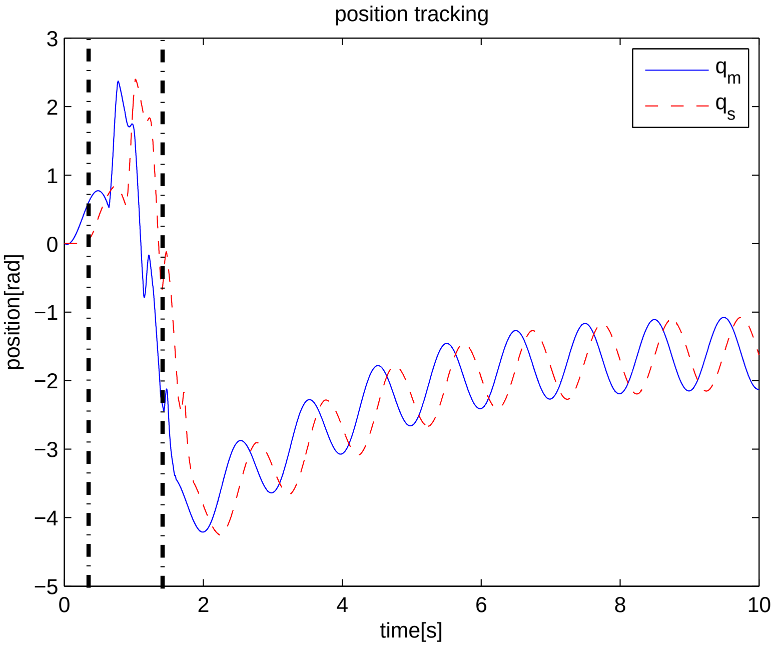


(a) Position Tracking

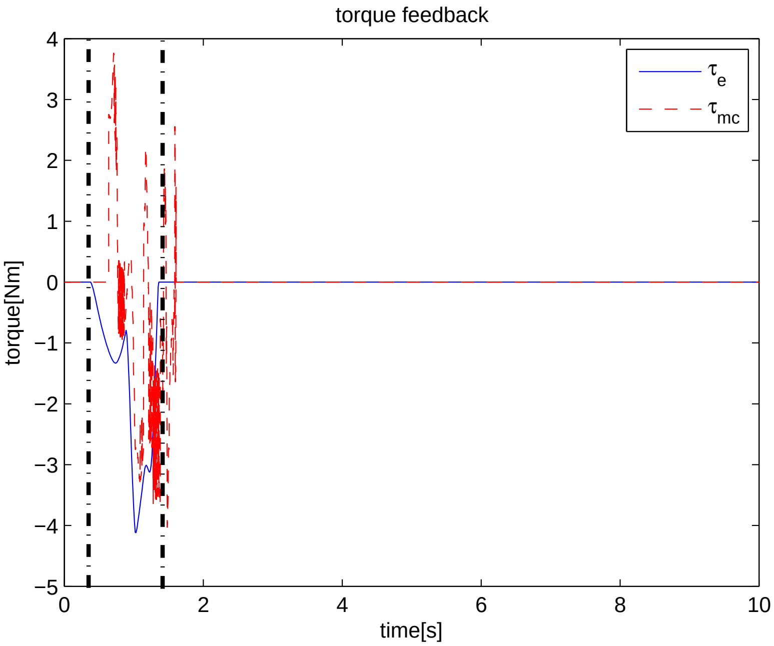


(b) Torque Feedback

Figure 5.15: Simulation results using AFLC, hard contact, zoomed view

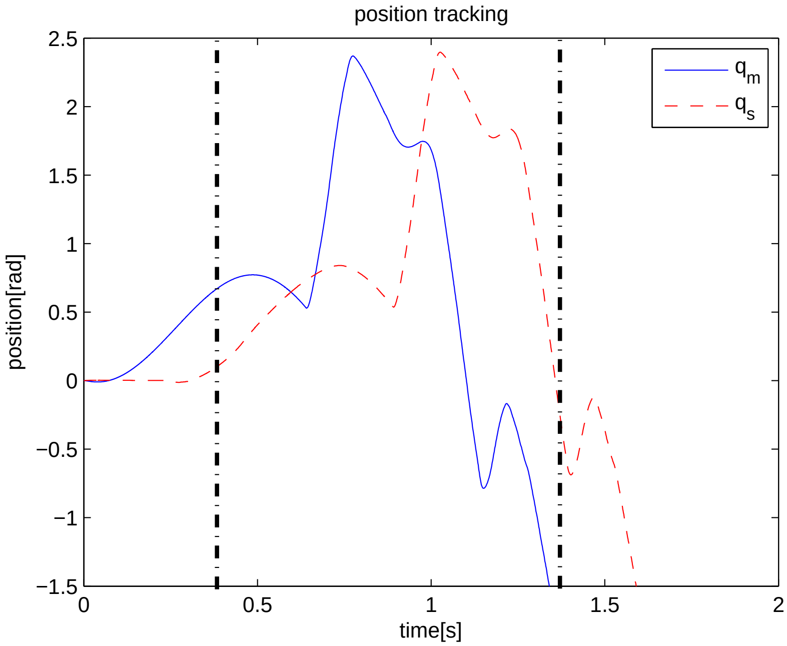


(a) Position Tracking

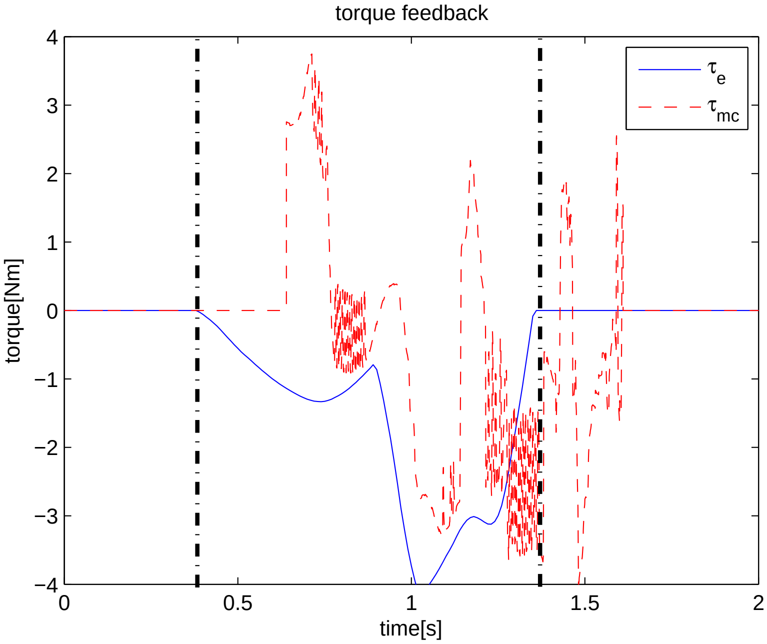


(b) Torque Feedback

Figure 5.16: Simulation results using AFLC, soft contact

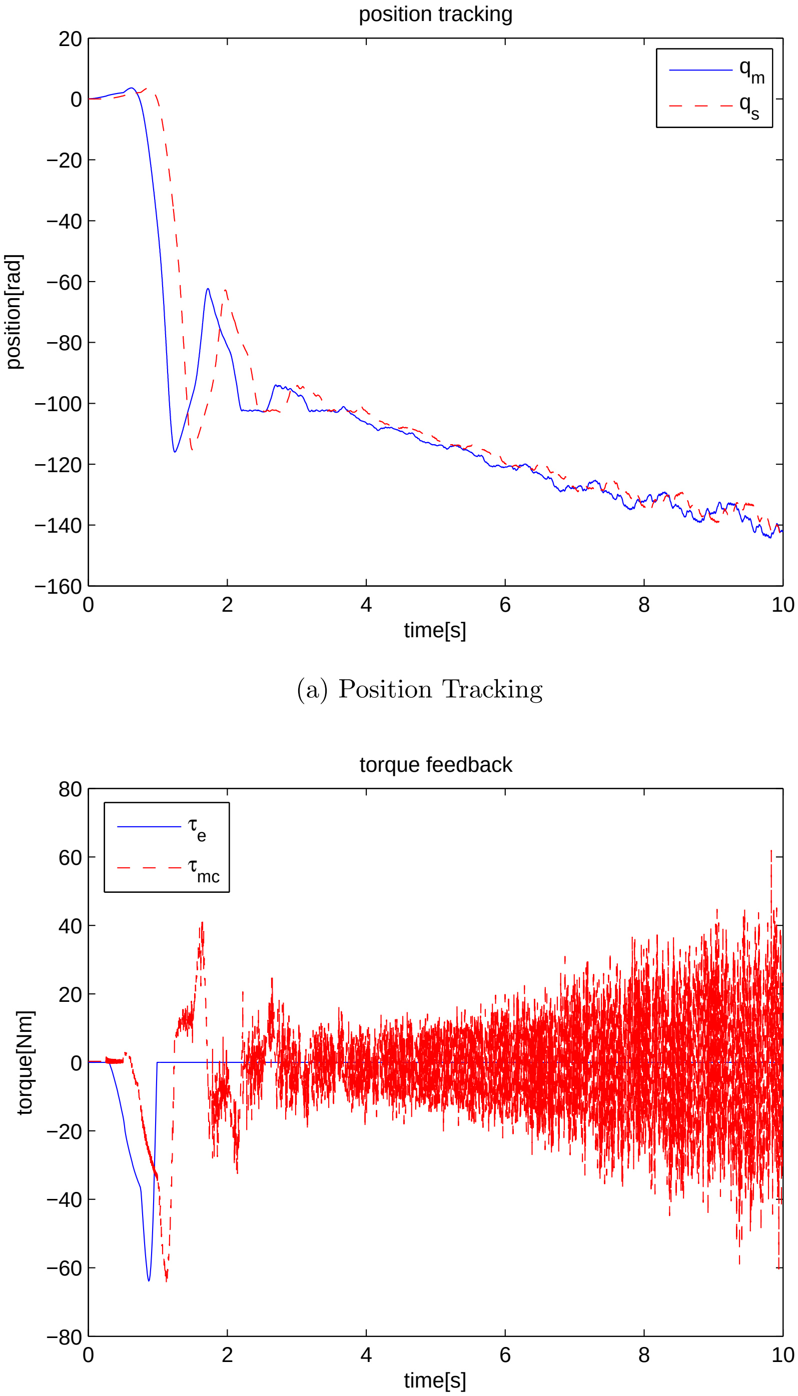


(a) Position Tracking



(b) Torque Feedback

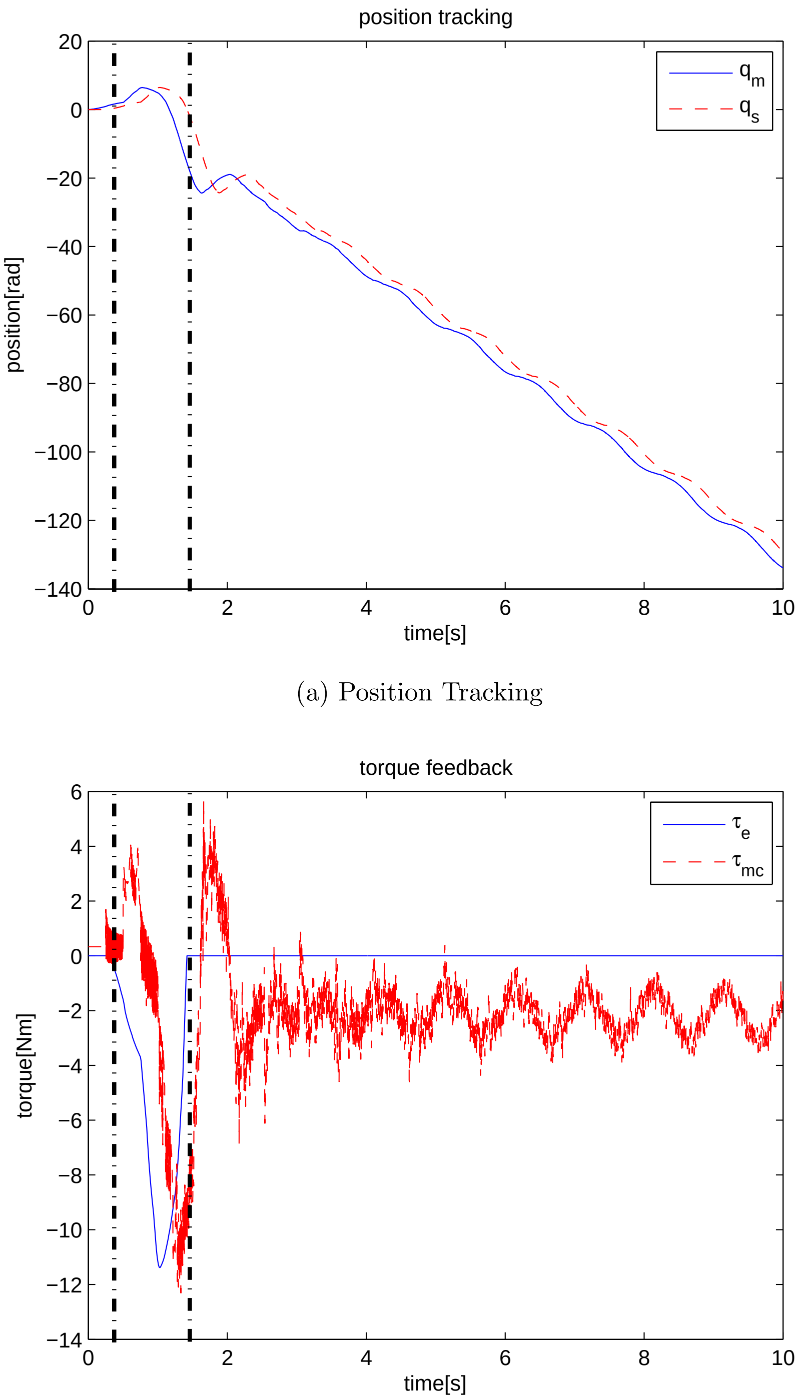
Figure 5.17: Simulation results using AFLC, soft contact, zoomed view



(a) Position Tracking

(b) Torque Feedback

Figure 5.18: Simulation results using AFLC, hard contact, without precaution step



(a) Position Tracking

(b) Torque Feedback

Figure 5.19: Simulation results using AFLC, soft contact, without precaution step

Therefore it would be desirable to find a way to simplify this controller for example by reducing the number of fuzzy membership functions for the inputs. Another disadvantage about the AFLC is that it requires the measurements of many values including velocity, acceleration and force data. Those usually have noisy signals, therefore observers should be considered when this controller will be tested in experiment.

In summary, the AFLC presented hereby shows a relatively satisfactory performance in both position tracking and force feedback. It has flaws which need to be fixed in future works, but it demonstrated some very promising results.

5.5 Summary and General Comparison

As discussed in this chapter, the simulation results for each of the four controllers unveil their behaviour under different remote environments. The comparison between all four controllers mentioned in this chapter can be seen in an overview presented in table 5.1. Each scenario is evaluated using one of six scores: unstable, very poor, poor, acceptable, good and very good. Computational simplicity is also one very important criterion included in the table. In the table, the controller AFLC1 represents AFLC with the switch and AFLC 2 represents the case when it is without.

It can be seen that overall, the PD controller is unstable and the wave variable controller performs relatively poorly in all cases. The sliding mode controller's performance varies with the changing environment (it performs very well in free space, acceptable in hard environment and good in soft environment). The AFLC with the precaution step performs quite well except for force tracking in soft environment, but the AFLC without the precaution step performs well only in free space while going unstable upon contact in most cases.

Table 5.1: Comparison of controllers based on the simulation results

Criterion/Controller	PD	WV	SMC	AFLC1	AFLC2
no-contact position tracking	unstable	poor	very good	very good	very good
no-contact force tracking	unstable	poor	very good	very good	very good
hard-contact position tracking	unstable	poor	acceptable	very good	unstable
hard-contact force tracking	unstable	poor	acceptable	good	unstable
soft-contact position tracking	unstable	poor	good	very good	unstable
soft-contact force tracking	unstable	poor	acceptable	poor	poor
computational simplicity	good	very good	acceptable	poor	poor

Chapter 6

Conclusion

6.1 Thesis Overview

In this text, the design of an adaptive fuzzy logic controller (AFLC) was attempted for bilateral teleoperation for a better integration of this technology in the field of telemedicine. Medicine is different than many other fields because the slave robot would encounter both hard and soft environments. Therefore the controller must have a good performance for both extremes and every situation in between.

Bilateral teleoperation is a fairly new field of research where scholars have already proposed the use of many passivity-based, prediction-based and model-based controllers. The controller designed in this thesis is a new type of model-based controller and is relying on the concept of fuzzy logic in order to mimic human logic to control the system. Furthermore, this controller is adaptive, which gives it the ability to accommodate to environments with different stiffness coefficients. An adaptive law was developed based on Lyapunov's stability theorem and should ensure system stability. With the simulation results, it can be seen that this controller showed a fairly satisfactory position and force tracking performance for both hard and soft environment compared to a few milestone controllers in the literature.

However, the use of an extra step was necessary to achieve this performance by ignoring the AFLC's output if there is no contact sensed on the remote site. This precaution step as well as the nature of the AFLC made the simulation quite slow. Also observers should be investigated for real experimentations. Nonetheless, the AFLC demonstrates a very good potential for the system performance upon contact with both soft and hard environments.

Having gone through these steps, this thesis contributed to the exploration of the use of adaptive fuzzy logic control in the field of bilateral teleoperation with time delay. The controller was rigorously proven stable using Lyapunov's stability theorem and tested in simulation. A comparison study was done using the outputs of the AFLC and those of three bench-marker controllers', namely the PD controller, the wave variable controller, and the second order sliding-mode controller. It was shown that by using an intelligent controller, it is possible to achieve a better transparency performance while switching between environments of different stiffness coefficients.

6.2 Future Works

In order to pursue research using AFLC in this field, one of the foremost concern would be the need to completely stabilize the controller as it demonstrates instability without the use of the precaution step in Simulink. The instability might be caused by the values of specific parameters within AFLC. More investigation will need to be done on this end.

Another very important future work is to decrease simulation time as the AFLC is demanding lots of computing resources to perform a high amount of calculations. Suggestions include the decrease of membership function number for the inputs and/or that for the outputs.

Investigation on the observers is also an important step as this controller is requiring a lot of position, velocity, acceleration and force values. Observers can help avoiding the collection of some rather noisy signals, but they need to be carefully designed in order to avoid inducing any error.

Once these problems can be solved, the controller will need to be tested in real time with hardware, in a real experiment. Once succeeded, this controller can potentially be modified and used for more advanced research in bilateral teleoperation. This includes bilateral teleoperation with variable time delay, very large time delay, cooperative robots and/or scaled operations.

Appendix A

Lyapunov's Stability Theorem

Lyapunov's Stability Theorem was introduced in 1892 by the Russian mathematician Aleksandr Lyapunov. It is a method used to determine the stability properties of an equilibrium point without having to solve the state equation. There are two methods originally suggested but whenever "Lyapunov's Stability Theorem" is being mentioned in the literature, it is sometimes specifically referring to the second method of Lyapunov. In those cases it can also be called "Lyapunov's second method for stability". It is now a widely used technique especially in the field of nonlinear systems.

Consider a dynamic system $\dot{x} = f(x,t)$, where the origin $f(x,0) = x_0$ is given. Then x_0 is stable if there exists a continuously differentiable *positive definite* Lyapunov function candidate $V(x)$ such that $\dot{V}(x)$ is *semidefinite*, and is asymptotically stable if $\dot{V}$ is *negative definite.*

First let's define the function properties. The function $V(x)$ is said to be

- **Positive definite** if $V(0) = 0$ and $V(x) \geq 0$ for all $x \neq 0$
- **Positive semidefinite** if $V(x) \geq 0$
- **Negative definite** if $-V(x)$ is positive definite
- **Negative semidefinite** if $-V(x)$ is positive semidefinite

In the context of stability at an equilibrium point x_e, the point is said to be **stable** if any point $x(t_0)$ starting close enough to it remains close enough to it. Referring to figure A.1 and using a more formal definition, the equilibrium point x_e is stable in the sense of Lyapunov if for all t_0 and for all $\epsilon > 0$, there exists a $\delta(t_0, \epsilon) > 0$ such that if $\parallel x(t_0) - x_e(t_0) \parallel < \delta$, then $\parallel x(t) - x_e(t) \parallel < \epsilon$ for all $t \geq t_0$, where t_0 denotes the starting point of time.

Taking stability a step further, an equilibrium point x_e is said to be **asymptotically**

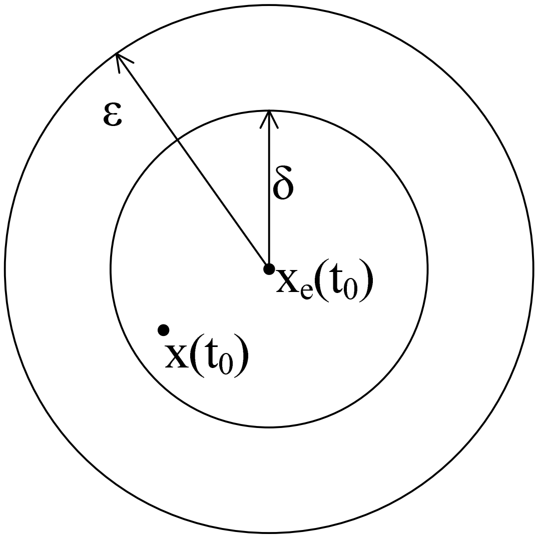


Figure A.1: An illustration of Lyapunov's stability theorem

stable if any point starting close enough to it eventually converges to it. In a more formal definition, the equilibrium point x_e is asymptotically stable if x_e is stable and ϵ can be chosen such that $\parallel x(0) \parallel < \delta$ implies that $x(t)$ approaches x_e as time goes to infinity.

In a number of situations, $V(x)$ may be taken as the potential energy of the system. This can very well explain the concept of Lyapunov's theorem. If the potential energy of the system at an equilibrium point (x_0) is decreasing or staying constant ($\dot{V} \leq 0$), then it is sensible that the system output will not grow out of bound. However if this potential energy was increasing ($\dot{V} > 0$), then it will not converge or stay to the original equilibrium point.

It should be noted however that while a system can be proven as stable or asymptotically stable by the finding of a successful Lyapunov's function candidate $V(x, t)$, nothing can be said with the finding of an unsuccessfull Lyapunov function candidate. The finding of an acceptable $V(x, t)$ remains the main difficulty in the use of Lyapunov's stability theorem. Interested readers can refer to [36, 44, 60] for this theorem's proof as well as further information.

Appendix B

Robotic Arm Modelling

B.1 Spatial Descriptions and Transformations

In the literature, robots are usually modelled as a succession of rigid links. Each link can be described by two variables: length l and position θ, where θ is the angle between the current link's position and the previous link's position. This can be illustrated as in the figures B.1. The length l is usually measured in meter and θ can be in degree or radian.

It should be noted that while fixing the axis XYZ for each link, Z-axis is always the one around which rotation is performed. In figure B.1, all the Z-axis are coming out of the page and were not drawn on the figure to keep it simple. All the coordinate systems are fixed on a specific point on the robotic arm and are numbered as 0, 1, 2, etc. Usually, coordinate system $X_0Y_0Z_0$ has its origin on the robot base and is fixed at all time; Coordinate system $X_1Y_1Z_1$ has its origin on the connecting point of link 1 to the base and rotates as link 1 rotates; The last coordinate system is usually considered as located in the tip of the furthest or last link.

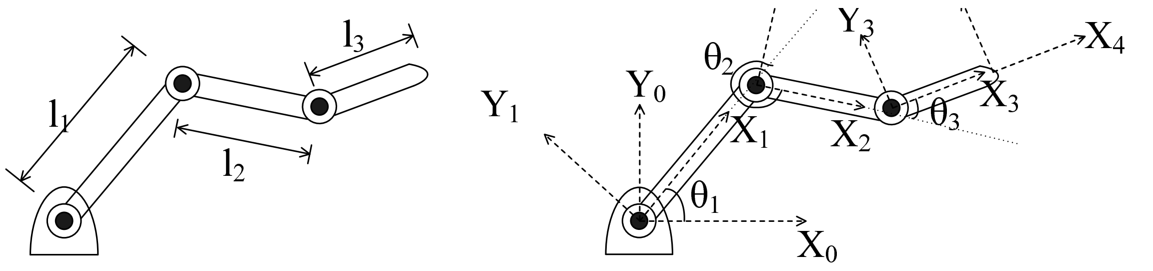


Figure B.1: Length and coordinate systems used for spatial description of robots

In this context, the Z-axis is the axis of rotation. The rotation of an angle θ around the Z-axis is denoted as

$$ {}^{0}_{1}R_{Z}(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

where ${}^{0}_{1}R$ indicates that it is a rotation from frame 0(base) to frame 1.

B.2 Dynamics

With this knowledge in mind, it now possible to introduce the iterative Newton-Euler dynamics algorithm. First, the outward iterations are presented. They are used from the inner-most link to the outer-most link

Angular velocity:

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\, {}^{i}\omega_{i} + \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} \tag{B.2}$$

where θ represents the angle between two links and $\hat{Z}$ is the vector in the direction of the axis of rotation. The notation Z_i indicates the Z-axis of link i.

Angular acceleration:

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\, {}^{i}\dot{\omega}_{i} + {}^{i+1}_{i}R\, {}^{i}\omega_{i} \times \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} \tag{B.3}$$

Linear acceleration of the base:

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_{i} \times {}^{i}P_{i+1} + {}^{i}\omega_{i} \times ({}^{i}\omega_{i} \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_{i}) \tag{B.4}$$

where ${}^{i}P_{i+1}$ is the position vector which goes from the base of link i to the base of link $i+1$

Linear acceleration of the centre of mass:

$$ {}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1} \tag{B.5}$$

where ${}^{i+1}P_{C_{i+1}}$ is the position vector which goes from the base of link $i+1$ to its centre of mass.

Force at centre of mass:

$$ {}^{i+1}F_{i+1} = m_{i+1}\, {}^{i+1}\dot{v}_{Ci+1} \tag{B.6}$$

Torque at centre of mass:

$$ {}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\, {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\, {}^{i+1}\omega_{i+1} \tag{B.7}$$

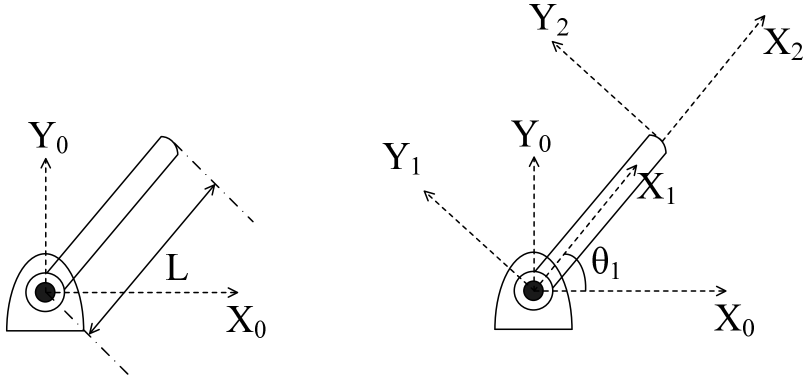


Figure B.2: Robot arm modelled

where ${}^{C_{i+1}}I_{i+1}$ is the inertia tensor written at the centre of mass for each link.

The inward iterations are shown by the equations below. They are used from the outer-most link to the inner-most link.

Force exerted on link i by link $i+1$:

$$ {}^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i \tag{B.8}$$

where the last ${}^{i+1}f_{i+1}$ is the external force directly applied on the end-effector of the robot arm.

Torque exerted on link i by link $i+1$:

$$ {}^{i}n_i = {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} \tag{B.9}$$

Linear actuator force:

$$ \tau_i = {}^{i}n_i^T\,{}^{i}\hat{Z}_i \tag{B.10}$$

B.3 Modelling

For the sake of simplicity, a robotic arm with one link is considered as shown in figure B.2. But it should be pointed out that this method is easily expendable to robots with multiple links (hence multiple degree of freedom). Its dynamics are described as in the follow equations, based on the Newton-Euler dynamics algorithm. It is important to remember that the numbers involved are referring to the corresponding frame and coordinates. (For example, ${}^{1}_{0}R$ denotes a rotation from frame 0 to frame 1.)

First, the outward iteration is examined starting with coordinate system 0 (base).

Angular velocity:

$$^{0}\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.11}$$

Angular acceleration:

$$^{0}\dot{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.12}$$

Linear acceleration of the base:

$$^{0}\dot{v}_0 = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \tag{B.13}$$

where $g = -9.81 m/s^2$ is the acceleration due to gravity, which is acting on the Y-direction.

Position Vector

$$^{0}P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.14}$$

For coordinate system 1 located on the point joining the base and link 1, the following is obtained:

Position vector to the centre of mass

$$^{1}P_{C_1} = \begin{bmatrix} L/2 \\ 0 \\ 0 \end{bmatrix} \tag{B.15}$$

where L is the length of the the link.

Angular Velocity

$$^{1}\omega_1 = {}^{1}_{0}R\,{}^{0}\omega_0 + \dot{\theta}_1\,{}^{1}\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix} \tag{B.16}$$

where θ is the angle between the X-axis or Y-axis of coordinate systems 0 and 1 (since the rotation is always done with respect to Z-axis, the rotation angle can be measured with X-axis and Y-axis). The rotation matrix is defined as in equation B.1.

Angular Acceleration

$$
{}^1\dot{\omega}_1 = {}^1_0R\,{}^0\dot{\omega}_0 + {}^1_0R\,{}^0\omega_{i0} \times \dot{\theta}_1\,{}^1\hat{Z}_1 + \ddot{\theta}_1\,{}^1\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta} \end{bmatrix} \tag{B.17}
$$

where θ is the angle between the horizontal surface and the robot link.

Linear acceleration of the base:

$$
{}^1\dot{v}_1 = {}^1_0R({}^0\dot{\omega}_0 \times {}^0P_1 + {}^0\omega_0 \times ({}^0\omega_0 \times {}^0P_1) + {}^0\dot{v}_0) = \begin{bmatrix} sin\theta * g \\ cos\theta * g \\ 0 \end{bmatrix} \tag{B.18}
$$

Linear acceleration of the centre of mass:

$$
{}^1\dot{v}_{C_1} = {}^1\dot{\omega}_1 \times {}^1P_{C_1} + {}^1\omega_1 \times ({}^1\omega_1 \times {}^1P_{C_1}) + {}^1\dot{v}_1 = \begin{bmatrix} -\frac{1}{2}\dot{\theta}^2 L + sin\theta g \\ \frac{1}{2}\ddot{\theta}L + cos\theta g \\ 0 \end{bmatrix} \tag{B.19}
$$

Force at centre of mass:

$$
{}^1F_1 = m\,{}^1\dot{v}_{C1} = \begin{bmatrix} -m(\frac{1}{2}\dot{\theta}^2 L + sin\theta g) \\ m(\frac{1}{2}\ddot{\theta}L + cos\theta g) \\ 0 \end{bmatrix} \tag{B.20}
$$

where m is the mass of the link.

Torque at centre of mass:

$$
{}^1N_1 = {}^{C_1}I_1\,{}^1\dot{\omega}_1 + {}^1\omega_1 \times {}^{C_1}I_1\,{}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{12mL^2\ddot{\theta}} \end{bmatrix} \tag{B.21}
$$

It is assumed that the link of the robotic arm is a slender rod along x-axis of length L and mass m. Hence the inertia tensor is

$$
{}^{C_1}I_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{12}mL^2 & 0 \\ 0 & 0 & \frac{1}{12}mL^2 \end{bmatrix}
$$

For the inward iterations, an external force ${}^2f_2 = [F_xF_yF_z]^T$ is considered to be acting on the tip of the robotic arm, or coordinate system 2.

First, the coordinate system 2 is considered. The term *link 2* is used to refer to an imaginary link attached on the coordinate system 2.

Torque exerted on link 2:

$$ {}^{2}n_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.22} $$

For link 1:

Position Vector

$$ {}^{1}P_2 = \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix} \tag{B.23} $$

Force exerted on origin of coordinate system 1 **by link** 2:

$$ {}^{1}f_1 = {}^{1}_{2}R\,{}^{2}f_2 + {}^{1}F_1 = \begin{bmatrix} F_x + m(-\frac{1}{2}\dot{\theta}^2 L + sin\theta g) \\ F_y + m(\frac{1}{2}\ddot{\theta}^2 L + cos\theta g) \\ 0 \end{bmatrix} \tag{B.24} $$

where the rotation matrix ${}^{1}_{2}R$ is defined as follow

$$ {}^{1}_{2}R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.25} $$

torque exerted on link 1 **by link** 2

$$ {}^{1}n_1 = {}^{1}N_1 + {}^{1}_{2}R\,{}^{2}n_2 + {}^{1}P_{C_1} \times {}^{1}F_1 + {}^{1}P_2 \times {}^{1}_{2}R\,{}^{2}f_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{12}mL^2\ddot{\theta} + \frac{1}{2}Lm(\frac{1}{2}\ddot{\theta}L + cos\theta g) + L\,F_y \end{bmatrix} \tag{B.26} $$

Extracting the Z-component gives the **Linear actuator force**:

$$ \tau_1 = {}^{1}n_1^T\,{}^{1}\hat{Z}_1 = \frac{1}{12}mL^2\ddot{\theta} + \frac{1}{2}Lm(\frac{1}{2}\ddot{\theta}L + cos\theta g) + L\,F_y \tag{B.27} $$

Regrouping the torque terms to the left hand side of the equation gives

$$ \tau_1 - L\,F_y = \frac{1}{3}mL^2\ddot{\theta} + \frac{1}{2}mL\,cos\theta\,g \tag{B.28} $$

Adding the friction term in and isolating the acceleration term gives:

$$ \tau_1 - LF_y - F_{friction} = \frac{1}{3}mL^2\ddot{\theta} \tag{B.29} $$

$$\ddot{\theta} = (\tau_1 - LF_y - F_{friction})(\frac{1}{3}mL^2)^{-1} \tag{B.30}$$

The equation B.30 was used for the robot modelling. For both master and slave robots, the term $\theta_1 - LF_y$ represents the summation of applied torques. For the master robot, it is the human operator's input force and the master robot controller's input force. For the slave robot, it is the force caused by contact with remote environment and the slave robot controller's input force. In all simulations, force was converted to torque using equation B.31

$$\tau = L \times F \tag{B.31}$$

where L is the robot link's length (0.1 m) and F is the force applied. The friction term $F_{friction}$ was chosen to be $F_{friction} = 0.15\dot{\theta}$. This is the viscous friction generated when two surfaces separated by a liquid slide against each other. It is a reasonable assumption since the rotating link should be supported by a lubricated bearing. The coefficient 0.15 is chosen based on the coefficient of sliding friction between two surfaces of hard steel [17].

Therefore, the final piece of coding used for the simulation is given as below:

```
function Qdotdot = robotR(tau, Q, Qdot) % function delaration, inputs are tau (total input toruque in N m) Q (position in rad) and Qdot(velocity in rad s^{-1}); output is Qdotdot (acceleration in rad/s^2)
r1=0.2;%link length in m
m1=0.1;%link mass in kg
D=r1*r1/3*m1;%computation of D matrix
g=-9.81;%acceleration due to gravity
G=r1/2*m1*cos(Q)*g; %torque needed due to gravity is set to zero
F=-0.15*Qdot;%torque due to friction
Qdotdot=inv(D)*(tau+F+G);%final output in rad/s^2
```

It should be noted that only the information relevant to this case of robotic arm modelling is included in this appendix. Readers who are interested to know more about robotic mechanics and control or the expansion of 1 degree-of-freedom to multiple degree-of-freedom are referred to [13].

Appendix C

Adaptive Fuzzy Logic Controller Coding

This section shows the coding used to simulate the adative fuzzy logic controller (AFLC). It is ran in an embedded Matlab editor in Simulink and is found in the “flcGauss” blocks as shown in figure 4.14. The same piece of code is used for all the places involving AFLC.

%————————————————start————————————————-
%The function name is flcGauss.
Its inputs are

- X:the array of variables to be compared, it could be position, velocity, x_r,$\dot{x}_r$, etc.
- s: $s = \dot{e} + \Lambda e$ where e is the position between desired and actual robot position.
- inputCenters: the array containing the new centre points of the membership functions in the output universe of discourse.

The function outputs are:

- fsc: the output force
- outputMfCenters1: the array containing the modified centre points of the membership functions in theoutput universe of discourse. It is fed back to the input *outputMFCenters*

function [fsc,outputMfCenters1] = flcGauss(X, s, inputCenters)
outputMfCenters=inputCenters’;
mfCenters=[-1 -0.5 0 0.5 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; -1 -0.5 0 0.5 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
%Following is the rule base. It considers two inputs X and s, each having 5 different

```
degrees of description
Rules=[1 1 1;
2 1 2;
1 2 3;
3 1 4;
1 3 5;
2 2 6;
4 1 7;
1 4 8;
3 2 9;
2 3 10;
5 1 11;
1 5 12;
4 2 13;
2 4 14;
3 3 15;
5 2 16;
2 5 17;
4 3 18;
3 4 19;
5 3 20;
3 5 21;
4 4 22;
5 4 23;
4 5 24;
5 5 25];
%parameters
n=2; %number of premises
fuzzySetNb=[5 5 25]; %number of fuzzy sets for the two inputs and the output, respectively
Lows=[-1 -1]; %numerical range for the universe of the two inputs
Highs=[1 1];

   %initialization
i=1;
```

```
num=0;
den=0;
sizeRules=size(Rules);
nbRules=sizeRules(1);
mfSigma=[0.2 0.2 0.2 0.2 0.2;0.2 0.2 0.2 0.2 0.2];
y_bar=zeros(1,nbRules); %y_bar is the center of the fuzzy members function at each combination (each combination is one rule)
for k=1:nbRules %1 to 25
    y_bar(k)=outputMfCenters(Rules(k,3));
end
%computation
mu1=0; %mu1 denotes the mu-values for X(1) – position
mu2=0; %mu2 denotes the mu-values for X(2) –¿velocity
mu=zeros(1,nbRules); %mu denotes the product of mu1 and mu2
    for k=1:nbRules %looping for each rule
    r1=Rules(k,1);
    r2=Rules(k,2);%for the 1st element in array X
    mu1=exp(-((X(1)-mfCenters(1,r1))/mfSigma(1,r1))exp2);
    switch(r1)
         case 1
              if X(1)<Lows(1)
                   mu1=1;
         end
         case fuzzySetNb(1)
              if X(1)>Highs(1)
                   mu1=1;
         end
    end%end switch, mu1 value now computed

    %for the 2nd element in array X
    mu2=exp(-((X(2)-mfCenters(2,r2))/mfSigma(2,r2))exp2);
    switch(r2)
         case 1
              if X(2)<Lows(2)
                   mu2=1;
```

```
        end
        case fuzzySetNb(2)
            if X(2)>Highs(2)
                mu2=1;
        end
    end %end switch, mu2 value now computed
    mu(k)=mu1*mu2;
%mu=2x25 multiplication, gives the probability of rule k
end %end for loop
%computing the denominator of equation 3.1
den=sum(mu);
%computing the numerator of equation 3.1
num=sum(y_bar.*mu);
%Computing the final output force fsc
fsc=num/den;
%Define the design parameter capGamma
capGamma=100; %smaller values = bigger step size
%Initializing and compute the individual Xi term as in equation 3.3 to update the centres of membership functions for the output as computed in chapter 4
Xi=zeros(1,nbRules);
Xi=mu/den;
tempOutput=zeros(fuzzySetNb(3),1); %initialize a temporary array
%updating the centres of output membership functions
for i=1:nbRules
    tempOutput(i)=y_bar(i)+(-s*Xi(i)/capGamma);
end
outputMfCenters1=tempOutput; %update the modified centres for output membership functions
end
```

Bibliography

[1] R.J. Anderson and M.W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, 1989.

[2] P. Arcara and C. Melchiorri. Control schemes for teleoperation with time delay: a comparative study. *Robotics and Autonomous Systems*, 38(1):49–64, 1 2002.

[3] H. Arioui, A. Kheddar, and S. Mammar. A predictive wave-based approach for time delayed virtual environments haptics systems. *Proceedings ofthe 2002 IEEE Int Workshop on Robot and Human Interactive Communication*, pages 134–139, Sep 2002.

[4] A. Aziminejad, M. Tavakoli, R.V. Patel, and M. Moallem. Wave-based time delay compensation in bilateral teleoperation:two-channel versus four-channel architectures. *Proceedings of the 2007 American Control Conference*, pages 1449–1454, July 2007.

[5] A. Aziminejad, M. Tavakoli, R.V. Patel, and M. Moallem. Stability and performance in delayed bilateral teleoperation: Theory and experiments. *Control Engineering Practice*, 16:1329–1343, 2008.

[6] A. Aziminejad, M. Tavakoli, R.V. Patel, and M. Moallem. Transparent time-delayed bilateral teleoperation using wave variables. *IEEE Transactions on control systems technology*, 16(3):548–515, May 2008.

[7] C. Benedetti, M. Franchini, and P. Fiorini. Stable tracking in variable time-delay teleoperation. *Proceedng of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2252–2257, Nov 2001.

[8] H. Ching and W.J. Book. Internet-based bilateral teleoperation based on wave variable with adaptive predictor and direct driff control. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 128:86–93, 2006.

[9] H. Cho and J. Park. Stable bilateral teleoperation under a time delay using a robust impedance control. *Mechatronics*, 15(5):611–625, June 2005.

[10] N. Chopra, P. Berestesky, and M.W. Spong. Bilateral teleoperation over unreliable communication network. *IEEE Transactions on control systems technology*, 16(2):304–313, March 2008.

[11] N. Chopra, M.W. Spong, S. Hirche, and M. Buss. Bilateral teleoperation over the internet. *Proceedings of the American Control Conferene*, 2003.

[12] N. Chopra, M.W. Spong, R. Ortega, and N.E. Barabanov. On tracking performance in bilateral teleoperation. *IEEE Transactions on robotics*, 22(4):861–866, August 2006.

[13] J.J. Craig. *Introduction to Robotics, Mechanics and Control, 3rd Edition*. Pearson Education Inc., 2005.

[14] J.M. Daly and D.W. Wang. Time-delayed bilateral teleoperation with force estimation for n-dof nonlinear robot manipulators. *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3911–3918, Oct 2010.

[15] R.W. Daniel and P.R. McAree. Fundamental limits of performance for force reflecting teleoperation. *The International Journal of Robotics Research*, 17(8):811–830, Aug 1998.

[16] M. Dede and S. Tosunoglu. tracking performance of an identical master-slave teleoperation system under variable time delays. *Novel Algorithms and Techniques in Telecommunications, Automation and Industrial Electronics*, pages 179–184, 2008.

[17] A.M. Sadegh E.A. Avallone, T. Baumeister III, editor. *Marks Standard Handbook for Mechanical Engineers*. McGraw Hill, 11 edition, 2007.

[18] A. Fattouh and O. Sename. H-infinity based impedance control of teleoperation systems with time delay. *5th IFAC Workshop on Time Delay Systems, France*, pages 1–5, 2003.

[19] A. Fernandez, A. Barreiro, A.Banos, and J. Carrasco. Reset control for passive teleoperation. *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, pages 2935–3940, November 2008.

[20] M. Ferre, M. Buss, R. Aracil, C. Melchiorri, and C. Balaguer. *Advances in Telerobotics (Springer Tracts in Advanced Robotics vol. 31).* Springer-Verlag Berlin Herdelberg, Germany, 2007.

[21] W. Ferrell. Delayed force feedback. *Hum. Factors*, 8(5):449–455, 1966.

[22] K. Fite, M. Goldfarb, and A. Rubio. Transparent telemanipulation in the presence of time delay. *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 1:254–259, 2003.

[23] A. Le Bot L. Laquieze G. Boyer, H. Zahouani. In vivo characterization of viscoelastic properties of human skin using dynamic micro-indentation. *Proceedings of the 29th Annual International Conference of the IEEE EMBS, Cit Internationale, Lyon, France*, pages 4584–4587, Aug 2007.

[24] S. Ganjefar, H. Momeni, and F. Janabi-Sharifi. Teleoperation systems design using augmented wave-variables and smith predictor method for reducing time-delay effect. *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, pages 333–338, 2002.

[25] L.G. Garcia-Valdovinos, V. Parra-Vega, and M.A. Arteaga. Higher-order sliding mode impedance bilateral teleoperation with robust state estimation under constant unknown time delay. *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1293–1298, July 2005.

[26] L.G. Garcia-Valdovinos, V. Parra-Vega, and M.A. Arteaga. Observer-based higher-order sliding mode impedance control of bilateral teleoperation under constant unknown time delay. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1692 –1699, October 2006.

[27] L.G. Garcia-Valdovinos, V. Parra-Vega, and M.A. Arteaga. observer-based sliding mode impedance control of bilateral teleoperation under constant unknown time delay. *Robotics and Autonomous Systems*, 55(8):609–617, August 2007.

[28] Y. Gu, C. Zhang, and K.T. Chong. Adaptive passive control with varying time delay. *Simulation Modelling Practice and Theory*, 18(1):1–8, 2010.

[29] B. Hannaford. Stability and performance tradeoffs in bi-lateral telemanipulation. *IEEE International Conference on Robotics and Automation, 1989. Proceedings.*, pages 1764–1767, May 1989.

[30] K. Hashtrudi-Zaad and S.E. Salcudean. Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation. *IEEE Transactions on robotics and automation*, 18(1):108–114, 2002.

[31] S. Hirche and M. Buss. Passive position controlled telepresence systems with time delay. *Proceedings of the 2003 American Control Conference*, 1:168–173, June 2003.

[32] S. Hirche and M. Buss. Packet loss effects in passive telepresence systems. *43rd IEEE Conference on Decision and Control*, 4:4010–4015, December 2004.

[33] P.F. Hokayem and M.W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42:2035–2057, 2006.

[34] J. Huang and F. Lewis. Neural-network predictive control for nonlinear dynamic systems with time-delay. *IEEE Trans. on Neural Networks*, 14(2):377–389, 2003.

[35] Kenji Kawashima, Kotaro Tadano, Ganesh Sankaranarayanan, and Blake Hannaford. Bilateral teleoperation with time delay using modi?ed wave variables. *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 424–429, 2008.

[36] H.K. Khalil. *Nonlinear Systems.* Prentice Hall, 3 edition, 2002.

[37] J. Kikuchi, K. Takeo, and K. Kosuge. Teleoperation system via computer network for dynamic environment. *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pages 3534–3539, May 1998.

[38] W.S. Kim, B. Hannaford, and A.K. Bejczy. force reflection and shared compliant control in operating telemanipulators with time delay. *IEEE Transactions on robotics and automation*, 8(2):176–185, April 1992.

[39] K. Kosuge, H. Murayama, and K. Takeo. bilateral feedback control of telemanipulators via computer network. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 1380–1385, 1996.

[40] D.A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–636, October 1993.

[41] D. Lee and M.W. Spong. Passive bilateral teleoperation with constant time delay. *Proceeding of the 2006 IEEE International Conference on Robotics and Automation*, 22(2):2902–2907, 2006.

[42] S. Leeraphan, T. Maneewarn, and D. Laowattana. Stable adaptive bilateral control of transparent teleoperation through time-varying delay. *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 2979–2984, Oct 2002.

[43] G.M.H. Leung, B.A. Francis, and J. Apkarian. Bilateral controller for teleoperators with time delay via mu-synthesis. *Robotics and Automation, IEEE Transactions on*, 11(1):105–116, February 1995.

[44] W.S. Levine, editor. *The Control Handbook*. CRC Press, 1999.

[45] X. Liu, W.J. Wilson, and X. Fan. Pose reflecting teleoperation using wave variables with wave prediction. *Proceedings of the IEEE International Conference on Mechatronics & Automation*, pages 1642–1647, Jul 2005.

[46] L.J. Love and W.J. Book. force reflecting teleoperation with adaptive impedance control. *IEEE Transactions on systems, man, and cyberneticsPart B: cybernetics*, 34(1):159–165, Feb 2004.

[47] R. Lozano, N. Chopra, and M.W. Spong. Passivation of force reflecting bilateral teleoperators with time varying delay. *in Proceedings of the 8. Mechatronics Forum*, pages 24–26, 2002.

[48] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S.E. Butner, and M.K. Smith. Transatlantic robot-assisted telesurgery. *Nature*, 413:379–380, September 2001.

[49] T. Mirfakhrai and S. Payandeh. A delay prediction approach for teleoperation over the internet. *Proceedings of the 2002 IEEE International Conference on Robotics &Automation*, pages 2178–2183, May 2002.

[50] M. Mulder. Stability in haptic teleoperation literature review, 2006.

[51] Espace Multimedia. “operation lindbergh”:a world first in telesurgery: The surgical act crosses the atlantic! press conference, September 2001.

[52] S. Munir and W. Book. Wave-based teleoperation with prediction. *Proceedings of the American Control Conference*, pages 4605–4611, 2001.

[53] Saghir Munir and Wayne Book. Internet-based teleoperation using wave variables with prediction. *IEEE/ASME Transactions on Mechatronics*, 7(2):124–133, June 2002.

[54] T. Namerikawa and H. Kawada. Symmetric impedance matched teleoperation with position tracking. *Proceedings of the 45th IEEE Conference on Decision & Control*, pages 4496–4501, Dec 2006.

[55] G. Niemeyer and J.J.E. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152–162, 1991.

[56] G. Niemeyer and J.J.E. Slotine. Designing force reflecting teleoperators with large time delays to appear as virtuall tools. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 2212–2218, April 1997.

[57] G. Niemeyer and J.J.E. Slotine. Towards force-reflecting teleoperation over the internet. *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pages 1909–1915, May 1998.

[58] G. Niemeyer and J.J.E. Slotine. Telemanipulation with time delays. *International Journal of Robotics Research*, 23(9):873–890, 2004.

[59] E. Nuno, L. Basanez, and R. Ortega. Passive bilateral teleoperation framework for assisted robotic tasks. *2007 IEEE International Conference on Robotics and Automation*, pages 1645–1650, Apr 2007.

[60] K. Ogata. *Modern Control Engineering.* Prentice-Hall, 3 edition, 2002.

[61] Y. Pan, J. Gu, M. Meng, and J. Jayachandran. Bilateral teleoperation of robotic systems with predictive control. *2007 IEEE International Conference on Robotics and Automation*, pages 1651–1656, Apr 2007.

[62] Y.J. Pan, C. Canudas de Wite, and O. Sename. A new predictive approach for bilateral teleoperation with applications to drive-by-wire systems. *IEEE Transactions on robotics*, 22(6):1146–1162, December 2006.

[63] J.H. Park and H.C. Cho. Sliding-mode controller for bilateral teleoperation with varying time delay. pages 311–315, September 1999.

[64] J.H. Park and H.C. Cho. Sliding-mode control of bilateral teleoperation systems with force-reflection on the internet. 2:1187–1192, 2000.

[65] J.H. Park and H.C. Cho. Sliding-mode-based impedance controller for bilateral teleoperation under varying time-delay. *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, pages 1025–1030, May 2001.

[66] K.M. Passino and S. Yurkovich. *Fuzzy Controlb.* Addison Wesley Longman Inc., 1998.

[67] I.G. Polushin, P.X. Liu, and C. Lung. A force-reflection algorithm for improved transparency in bilateral teleoperation with communication delay. *IEEE/ASME Transactions on Mechatronics*, 12(3):361–374, June 2007.

[68] E. Rodriguez-Seda. Comparative experimental study of control schemes for bilateral teleoperation systems. Master's thesis, Univ. Illinois, 2007.

[69] E.J. Rodriguez-Seda, D. Lee, and M.W. Spong. Experimental comparison study of control architectures for bilateral teleoperators. *IEEE Transactions on robotics*, 25(6):1304–1318, 2009.

[70] J. Ryu, D. Kwon, and B. Hannaford. Stable teleoperation with time-domain passivity control. *Robotics and Automation, IEEE Transactions on*, 20(2):365–373, Apr 2004.

[71] C. Sabanovic. Bilateral control with a re?ex mechanism on the slave side. *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, pages 195–200, 2005.

[72] M. Sha Sadeghi and H.R. Momeni. A new impedance and robust adaptive inverse control approach for a teleoperation system with varying time delay. *Science in China Series E: Technological Sciences*, 52(9):2629–2643, 2009.

[73] M. Sha Sadeghi, H.R. Momeni, R. Amirfar, and S. Ganjefar. Robust adaptive inverse controller design for position tracking in teleoperation systems. *16th IEEE International Conference on Control Applications, Part of IEEE Multi-conference on Systems and Control*, pages 563–568, Oct 2007.

[74] S.E. Salcudean, K. Hashtrudi-Zaad, S. Tafazoli, S.P. DiMaio, and C. Reboulet. Bilateral matched impedance teleoperation with application to excavator control. *Control Systems, IEEE*, 19:29–37, Dec 1999.

[75] M. Salter, C.A. Avizzano, A. Frisoli, P. Tripicchio, and M. Bergamasco. Bilateral teleoperation under time-varying delay using wave variables. *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4596–4602, October 2009.

[76] A. Sano, H. Fujimoto, and M. Tanaka. Gain-scheduled compensation for time delay of bilateral teleoperation systems. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 3:1916–1923, 1998.

[77] O. Sename. H-infinity control of a teleoperation drive-by-wire system with communication time-delay. *Control and Automation, MED '06. 14th Mediterranean Conference on*, 2006.

[78] A. Shahdi and S. Sirouspour. improved transparency in bilateral teleoperation with variable time delay. *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4616–4621, 2009.

[79] J. Sheng and M.W. Spong. Model predictive control for bilateral teleoperation systems with time delays. *Electrical and Computer Engineering, 2004. Canadian Conference on*, pages 1877–1880, May 2004.

[80] T.B. Sheridan. Teleoperation, telerobotics and telepresence: A progress report. *Control Eng. Pract.*, 3(3):205–214, 1995.

[81] S. Sirouspour and A. Shahdi. Model predictive control for transparent teleoperation under communication time delay. *IEEE Transactions on robotics*, 22(6):1131–1145, December 2006.

[82] A.C. Smith and K. Hashtrudi-Zaad. Smith predictor type control architectures for time delayed teleoperation. *The International Journal of Robotics Research*, 25(8):797–818, August 2006.

[83] C. Smith and H. Christensen. A minimum jerk predictor for teleoperation with variable time delay. *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5621–5627, Oct 2009.

[84] O.J.M. Smith. Closed control of loops with dead time. *Chem. Eng. Progress*, 53:217–219, 1957.

[85] Dezhen Song. *Sharing a Vision (Springer Tracts in Advanced Robotics vol. 51).* Springer-Verlag Berlin Herdelberg, Germany, 2009.

[86] S. Stramigioli, A. Schaft, B. Maschke, and C. Melchiorri. Geometric scattering in robotic telemanipulation. *IEEE transactions on robotics and automation*, 18(4):588–596, August 2002.

[87] N. Tanner and G. Niemeyer. Practical limitations of wave variable controllers in teleoperation. *Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics*, pages 25–30, Dec 2004.

[88] N.A. Tanner and G. Niemeyer. Improving perception in time delayed teleoperation. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 354–359, April 2005.

[89] A.F. Villaverde, A. Barreiro, and C. Raimundez. Passive position error correction in internet-based teleoperation. *Automatica*, 46(11):1884–1890, November 2010.

[90] A.F. Villaverde, A.B. Blas, J.Carrasco, and A.B. Torrico. Reset control for passive bilateral teleoperation. *IEEE Transactions on industrial electronics*, pages 3037–3045, July 2011.

[91] K.C. Walker, Y.J. Pan, and J. Gu. Bilateral teleoperation over networks based on stochastic switching approach. *IEEE/ASME Transactions on Mechatronics*, 14(5):539–554, October 2009.

[92] Li-Xin Wang. *A Course in Fuzzy Systems and Control.* Prentice Hall, 1997.

[93] S. Wang, B. Xu, Y. Jia, and Y. Liu. Real-time mobile robot teleoperation over ip networks based on predictive control. *2007 IEEE International Conference on Control and Automation*, pages 2091–2096, Jun 2007.

[94] X. Wang and P.X. Liu. Improvement of haptic feedback fidelity for telesurgical applications. *Electronics Letters*, 42(6):327–329, March 2006.

[95] Y. Wang, Z. Sun, and W. Chou. robust controller design for teleoperation systems with time-varying delays. *International Conference on Measuring Technology and Mechatronics Automation*, pages 266–269, 2010.

[96] N. Xi and T.J. Tarn. Action synchronization and control of internet based telerobotic systems. *Robotics and Automation, IEEE International Conference on*, 1:219–224, 1999.

[97] N. Xi and T.J. Tarn. Stability analysis of non-time referenced internet-based telerobotic systems. *Robotics and Autonomous Systems*, 32(2-3):173–178, 2000.

[98] J. Yan and S.E. Salcudean. Teleoperation controller design using h-infinity-optimization with application to motion-scaling. *IEEE Transactions on control systems technology*, 4(3):244–258, May 1996.

[99] Y. Ye and P.X. Liu. Improving force feedback fidelity in wave-variable-based teleoperation. *IEEE International Conference on Robotics and Automation*, pages 194–199, 2008.

[100] Y. Yokokohji, T. Imaida, and T. Yoshikawa. Bilateral teleoperation under time-varying communication delay. *Proceedings of the 1999 lEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1854–1859, 1999.

[101] Y. Yokokohji and T. Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling. *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 849–857, May 1992.

[102] K. Yoshida and T. Namerikawa. Stability and tracking properties in predictive control with adaptation for bilateral teleoperation. *2009 American Control Conference*, pages 1323–1328, 2009.

[103] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[104] T. Zhang and Y. Li. A control scheme for bilateral teleoperation systems based on time-varying communication delay identification. *1st International Symposium on Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006.*, pages 273–278, 2006.

[105] J. Zhu, X. He, and W. Gueaieb. Trends in the control schemes for bilateral teleoperation with time delay. *Second International Conference on Autonomous and Intelligent Systems, AIS2011*, pages 146–155, 2011.

[106] W. Zhu and S. Salcudean. Stability guaranteed teleoperation: an adaptive motion/force control approach. *IEEE Transactions on Automatic Control*, 45(11):1951–1969, 2000.