INTERNET-BASED BILATERAL TELEOPERATION

A Thesis
Presented to
The Academic Faculty

by

Ho Ching

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2006

INTERNET-BASED BILATERAL TELEOPERATION

Approved by:

Dr. Wayne J. Book, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Nader Sadegh
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Harvey Lipkin
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Bonnie S. Heck
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Julie A. Jacko
School of Biomedical Engineering
Georgia Institute of Technology

Date Approved by Chairman

ACKNOWLEDGEMENTS

I would first like to thank my parents and Priscilla for their moral support and patience regarding my graduate studies at Georgia Tech. I owe much of the success in my research activities to my advisor Prof. Wayne Book who provided invaluable guidance and advice. I greatly appreciate the technical support offered by my research group, namely J. D. Huggins for machining the contour and maze blocks and Matt Kontz for providing the codes for the Hydraulically Actuated Lifter (HAL). Without them this research would have taken substantially longer to complete. I would also like to thank my thesis committee members for lending their expertise to my study and proof-reading my thesis. Finally I would like to thank the numerous faculties and friends who make my stay with Georgia Tech a very pleasant and memorable learning experience.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

SUMMARY

In conventional bilateral teleoperation, transmission delay over the Internet can potentially cause instability. The wave variable algorithm guarantees stability under varying transmission delay at the cost of poor transient performance. Adding a predictor on the master side can reduce this undesirable side-effect, but that would require a slave model. An inaccurate slave model used in the predictor as well as variations in transmission delay, both of which are likely under realistic situations, can result in steady state errors. A direct drift control algorithm is used to drive this error to zero regardless of the source of error. A semi-adaptive predictor that can distinguish between free space and rigid contact environment is used to provide more accurate force feedback on the master side. A full adaptive predictor is also used that estimates the slave environment parameters using recursive least squares with a forgetting factor. This research presents the experimental results and evaluations of the wave variable based methods under a realistic operation environment using a real master and slave.

The effectiveness of this algorithm is fully evaluated using human subjects with no previous experience in haptics. Three algorithms are tested using PHANTOM brand haptic devices as master and slave: conventional bilateral teleoperation with no transmission delay as control, wave variable teleoperation with approximately 200 ms transmission delay one way, and wave variables with adaptive predictor and direct drift control with approximately 200 ms transmission delay one way. For each algorithm the human subjects are asked to perform three simple tasks: use the master to force the slave to track a reference trajectory in free space with the least amount of error, identify a contour surface on the slave side as accurately as possible using only haptic information from the master, and navigate a simple maze on the slave side in the least amount of time using haptic information from the master.

CHAPTER 1

INTRODUCTION

Telerobotics is a field of robotics that is concerned with the control of robots from a distance. In the most technical sense it involves both teleoperation and telepresence. Teleoperation means doing work from a distance while telepresence means feeling as if you are at the remote site. Teleoperation may involve the remote control of any sort of machine: robotic manipulator, remotely operated vehicle (ROV), unmanned aerial vehicle (UAV), etc. It can also involve controlling a device over a physical distance as well as over a difference in scale between the user and remote environment. Telepresence can be described as the feedback loop of telerobotics where the states of the remote environment are fed back to the user in various ways such as visual, audio, or force feedback. A system that involves teleoperation with force feedback is sometimes described as bilateral teleoperation because the information flows both ways producing a closed-loop system. On the other hand, a system that involves teleoperation with no force feedback can be described as unilateral teleoperation since the information only flows one way, and the system is open-loop. In this thesis, the word teleoperation automatically implies bilateral teleoperation. The human user side of the teleoperation system is termed as the master side, where the remote environment side is termed as the slave side. A generic bilateral teleoperation setup is shown in Fig. 1.1.

Teleoperation is desirable in many situations where it is not practical for the human to perform the task onsite. An example is nuclear waste disposal which can put an onsite human operator at risk. It may also be impractical to have a human directly operate onsite due to difference in scale. An example is micro-surgery where the surgeon has to operate at a cellular level. However, teleoperation is not the only way to perform a task in a remote environment. Another area of active robotics research involves a

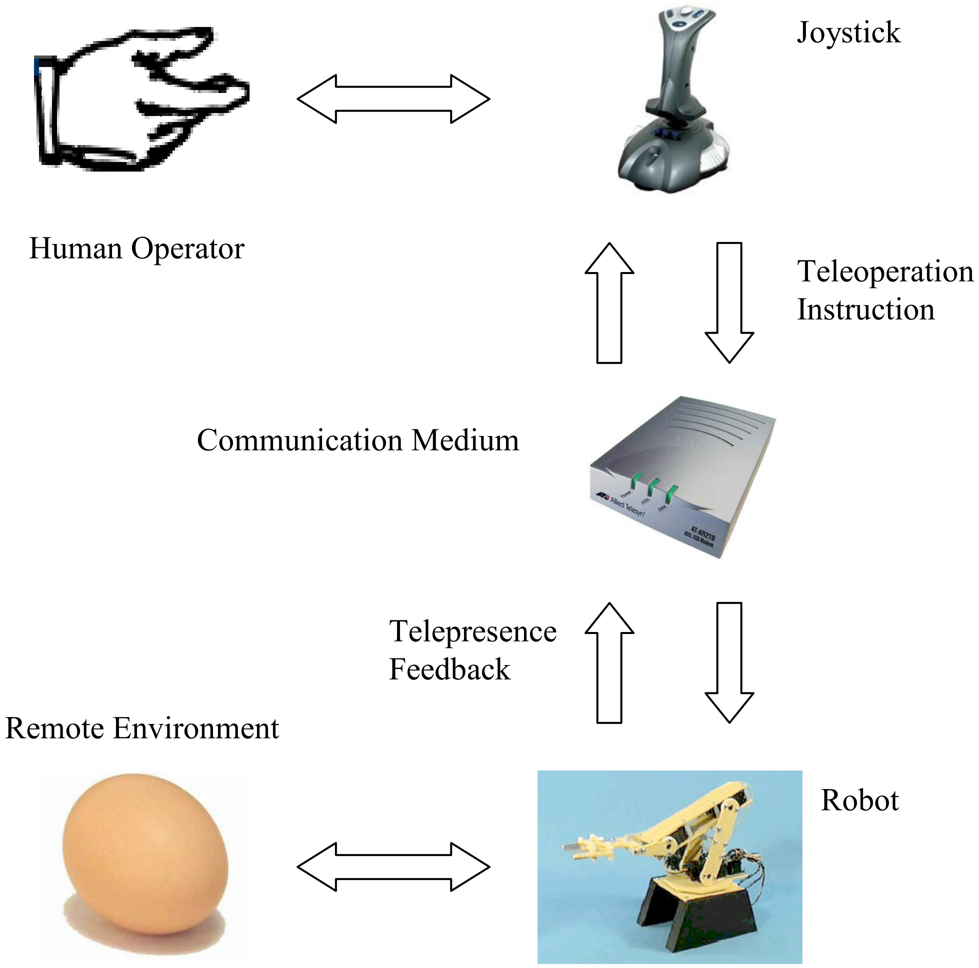


Figure 1.1: Typical Bilateral Teleoperation Setup

completely autonomous robot in the remote site. This method eliminates many problems involving data transmission between the master and slave, including transmission delay which is the focus of this thesis. In some cases one does not have a choice. For example, robots in an interplanetary mission must be autonomous to a certain degree because signals at the speed of light would take several minutes and even hours to reach the remote site. Teleoperation for tasks that require a reaction time of a few seconds is simply not feasible for such missions. Nevertheless at the current state of technology, full automation may not be reliable enough for many tasks that a human may take for granted. In tasks such as surgery, artificial intelligence still can not match human hands in dexterity and split-second decision making ability. Until automation has reached this level of sophistication, it is desirable to maintain the human in the control loop for more complicated and delicate tasks.

In a teleoperation system, a communication setup is necessary so the master can give instructions to the slave while the slave can send telepresence data back to the master. The communication line could be radio waves through space, sonar signals through water, data packets over the Internet, etc. The quality of this communication line is crucial to insure the teleoperation system operates smoothly. A common fault with communication line is transmission delay. The transmission delay in some communication lines can also be time-varying due to the unpredictability of the communication infrastructure. On top of that, data transmission may not be 100% reliable, meaning some data may not reach the intended destination at all.

This thesis focuses on bilateral teleoperation with telepresence via force feedback over the Internet. A possible basic bilateral teleoperation setup without transmission delay may include the master and slave plants as well as a local PD control on the remote device. A simple block diagram is shown in Fig. 1.2. Here x_m, x_s, F_h, F_{pd}, and F_e are

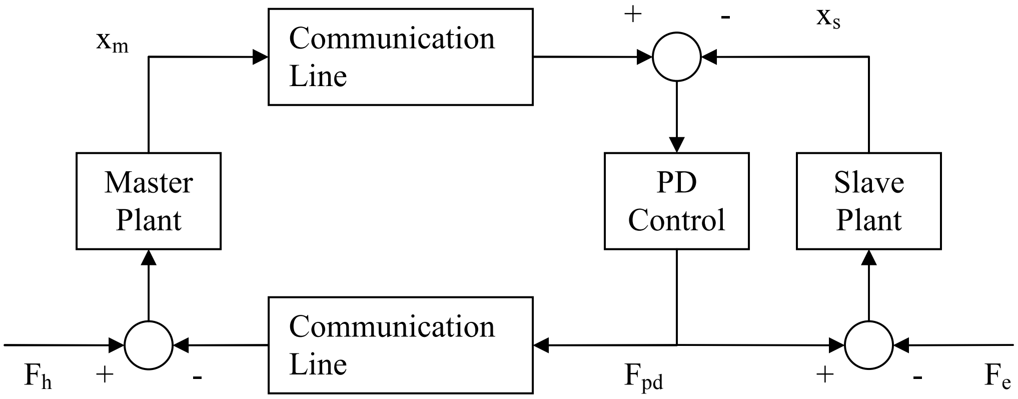


Figure 1.2: Basic Bilateral Teleoperation

master position, slave position, human input force, PD control force, and environmental disturbance force respectively. The communication line may have its own unique dynamics. In case of the Internet, the communication line may include variable transmission delays and lost data packets. These are serious problems in teleoperation because it amounts to essentially a pure variable delay in a closed-loop system. It is a well-known problem in system dynamics and control that a pure delay in a closed-loop system is very destabilizing. The goal of this research is to provide a solution to this stability problem in time delayed bilateral teleoperation without severe deterioration in system performance for both free space and rigid contact environments.

1.1 Haptics

The word *haptic* comes from the Greek word *haphe* meaning pertaining to the sense of touch. Teleoperation with haptic telepresence interacts with the user via force feedback in the joystick, providing a higher and unique degree of telepresence not easily achievable by visual or audio feedback. This is an emerging technology that promises a wide range of applications.

Haptic telepresence is not as common as visual or audio feedback, but it can prove extremely useful for certain teleoperation tasks. In fact, some seemingly simple tasks are very hard to execute without force feedback. For example, a human hand can hold an egg without crushing it because the hand can provide just the right amount of pressure based on its ability to sense feedback force. If the applied force is too small, the egg will slip out of the hand. If the applied force is too large, the hand will crush the egg shell. A teleoperation system without haptic telepresence does not allow a human user to achieve this simple task even with detailed visual and audio feedback.

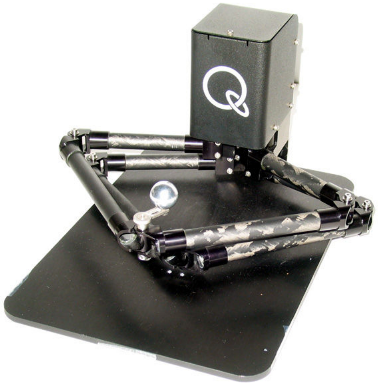
Haptic feedback can also provide better telepresence through taction. By tapping or moving a human hand along a surface, force feedback in the form of tiny vibrations allows the user to identify the texture of the surface and thus the type of surface material.

Again if this type of telepresence can be fed back to the user in teleoperation, he or she may be better able to accomplish the task in question.

One of the earliest applications of haptics is in aircraft servo-control [1]. In the early history of flight, airplanes are relatively small such that control surfaces are connected to the pilot control stick via mechanical linkages. Aerodynamic forces on the control surfaces are hence carried over to the pilot via the control stick which proves to be invaluable feedback to the pilot. For example, when an aircraft approaches the stall point, aerodynamic forces produce violent vibration on the control surfaces. A pilot can feel those vibrations on his control stick and use it as a warning as he or she is approaching a dangerous flight condition. However, as airplanes become large, mechanical linkages can not be efficiently designed to allow a pilot to exert large enough forces on the control surfaces. Hence servo-systems are introduced to control the aircraft where control forces are applied hydraulically or electrically. These servo-systems are normally one-way without feedback from the control surface. For the pilot to feel the force feedback from the control surfaces, haptic feedback is used to artificially reproduce it. With the advances in flight-by-wire technology over the past few decades, haptic feedback in flight control became ever more prevalent.

Another early pioneering application in haptic is nuclear engineering [2]. Some of the nuclear waste from nuclear reactors is so radioactive that it must be sealed inside thick lead containers at all times. In fact some of the first teleoperators are used during the development of the atomic bomb by the United States where radioactive materials are handled via robot arms that are mechanically connected to the human operator located in a separate room. Electrically actuated teleoperators are then developed to accomplish similar tasks.

Recently one can find haptics in advanced virtual reality applications which several commercially available haptic devices that include the PHANTOMs manufactured by Sensable Technologies. Figure 1.3 shows some of the commercially

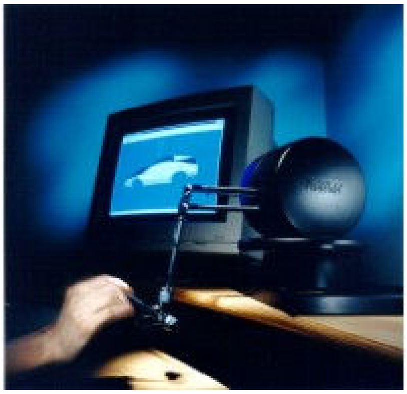


Quanser, Inc.
3-DOF Planar Pantograph

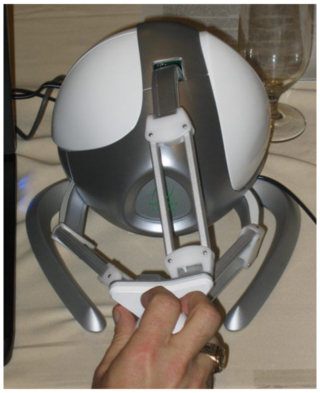
MFB Technologies
Freedom 6S Haptic Device

Immersion Corporation
Impulse Engine 2000

Sensable Technologies
PHANTOM Omni



Sensable Technologies
PHANTOM 1.5



Novint Technologies
Falcon

Figure 1.3: Sample of Commercially Available Haptic Devices

available haptic devices. Some computer aided design (CAD) use haptics, adding the sense of touch to previously visual solutions. An example is the FreeForm Modeling system developed by Sensable for its PHANTOM haptic devices [3]. This helps engineers to assemble a virtual machine more intuitively. Haptics has also been used extensively in the field of medicine where it may prove especially useful in minimally intrusive surgical procedures. Intuitive Surgical is a company that specializes in this application [4]. The surgeon can use haptic teleoperation to operate on a much smaller area using a surgical robot designed to operate on a nearly microscopic level. Furthermore, telesurgery allows the surgeon to operate from a central workstation while performing surgery at various locations via the Internet. This allows him or her to perform on many more similar surgeries with less fatigue, resulting in better outcome for the patients.

The price of today's haptic device is too high to allow wide spread usage in everyday home electronics outside of research and industrial institutions. The least expensive model of haptic device available from Sensable Technologies, the PHANTOM Omni, costs over US$5000. Some gaming joysticks and joypads are currently available with crude force feedback for a modest price. For example, PlayStation 2 joypads contain rumble packs which are simple attachments that vibrate upon command from the game console. Some haptic steering wheels are also available specifically designed for race car simulation games. Novint Technologies is currently trying to break into the home electronics consumer market for haptic devices via the gaming industry [4]. The company is planning to produce a haptic device called the Falcon as a substitute for the mouse in gaming (Falcon preys on the mouse). It is comparable in performance to a lower end haptic device from Sensable Technologies, but the planned retail price is under US$100. This shows how rapidly high precision haptic technology is expanding and becoming more affordable to the masses.

1.2 Transmission Delay

Transmission delay, sometimes also termed as latency, of a communication line is the time from the start of data packet transmission at the source to the start of data packet reception at the destination. The source of the latency can vary from the speed of the signal to how the signal is relayed among various gateways. In many instances this delay can become significant enough to become noticeable by a human.

The algorithms proposed in this thesis are specially designed for and tested with the Internet as the communication medium, although in principle they are applicable to any sort of transmission delay. The Internet is a complex network of servers and clients where data transmission is not direct but is forwarded over many links via many gateways. This can produce significant latency especially at certain times of the day with heavy network congestion and in areas with poor network infrastructure. The unpredictability of the Internet can result in variations in latency as well as lost data packets. To perform bilateral teleoperation over the Internet, the system must solve the problems posed by these pure delays in what is effectively a closed-loop system.

Before we continue, it is important to discuss the difference between bandwidth and latency. Many Internet service providers imply higher bandwidth means low latency. This is not entirely true. Bandwidth is the measure of how many data packets can travel through the network over a given period of time while latency is the amount of time the data packets takes to travel from the source to the destination. These two network properties are not usually related to each other, but together they determine the perceived speed of a connection. A good way to understand the difference between latency and bandwidth is by comparing the postal service and the Internet. Even though the Internet has much lower latency than the postal service with transmission time measured in seconds rather than days, the postal service potentially has much higher bandwidth if the package mailed is a box full of DVDs. What it means is that bilateral teleoperation problem related to transmission delay over the Internet can not be solved simply by increasing the bandwidth of the connection because this only increase the number of data packets one can receive in a given time, not the speed of the data packets.

The primary protocol for Internet data transmission is the transmission control protocol (TCP). This is a confirmation-based protocol where the transmitting computer sends a data packet and then requests a confirmation of reception from the receiving computer. If the confirmation is not received within a reasonable amount of time, the transmitting computer resends the data packet in question. This insures 100% of the data is transmitted with high reliability, making this protocol very popular in web browser programs. However, this protocol is also unpredictable in transmitting data in a timely manner because a lot of time may be spent waiting for confirmation. Figure 1.4 shows a typical TCP round trip transmission delay as tested by Munir and Book [21]. Each data packet sent includes a time stamp and a bit of a sine wave. The signal is sent from Atlanta and reflected off a server in France back to the source. As the figure shows, by the time the sine wave is reflected back to the source, the shape is virtually unrecognizable even though all the data packets do eventually arrive. The transmission delay is also very unpredictable with large variations.

In real time operations such as online gaming, some programmers would prefer to use user datagram protocol (UDP). This protocol eliminates the need for confirmation where the transmitting computer keeps sending the data packets with no regard as to whether the receiving computer has received the data. This means that all the data are sent in a timely fashion, an important feature for real time operations. But the lack of confirmation also means that it is less reliable. Figure 1.5 shows a typical UDP round trip transmission delay as tested by Munir and Book [21]. The experiment was performed in a similar fashion as the TCP case. As we can see, even though not all the data packets made it through the Internet, the basic shape of the sine wave is still recognizable. The transmission delay is also more stable than TCP.

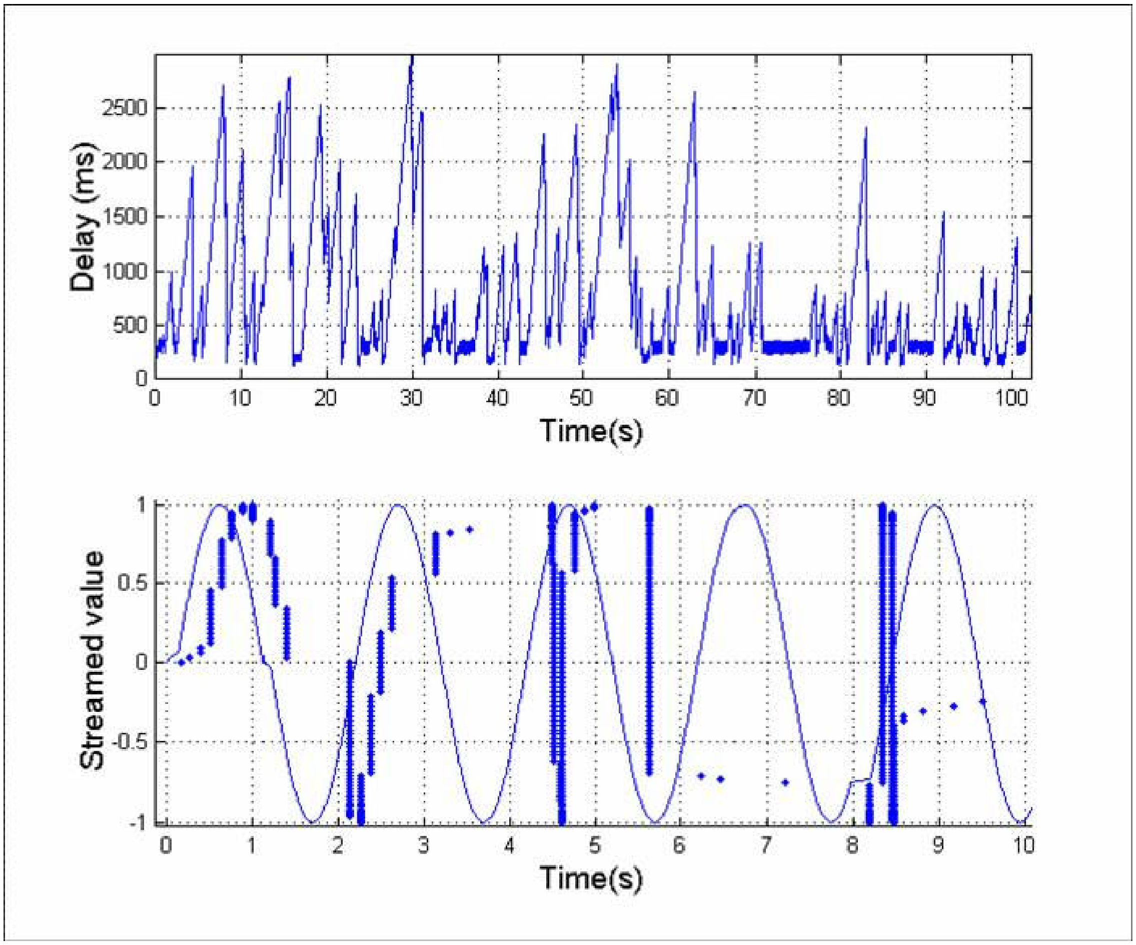


Figure 1.4: TCP Transmission Delay

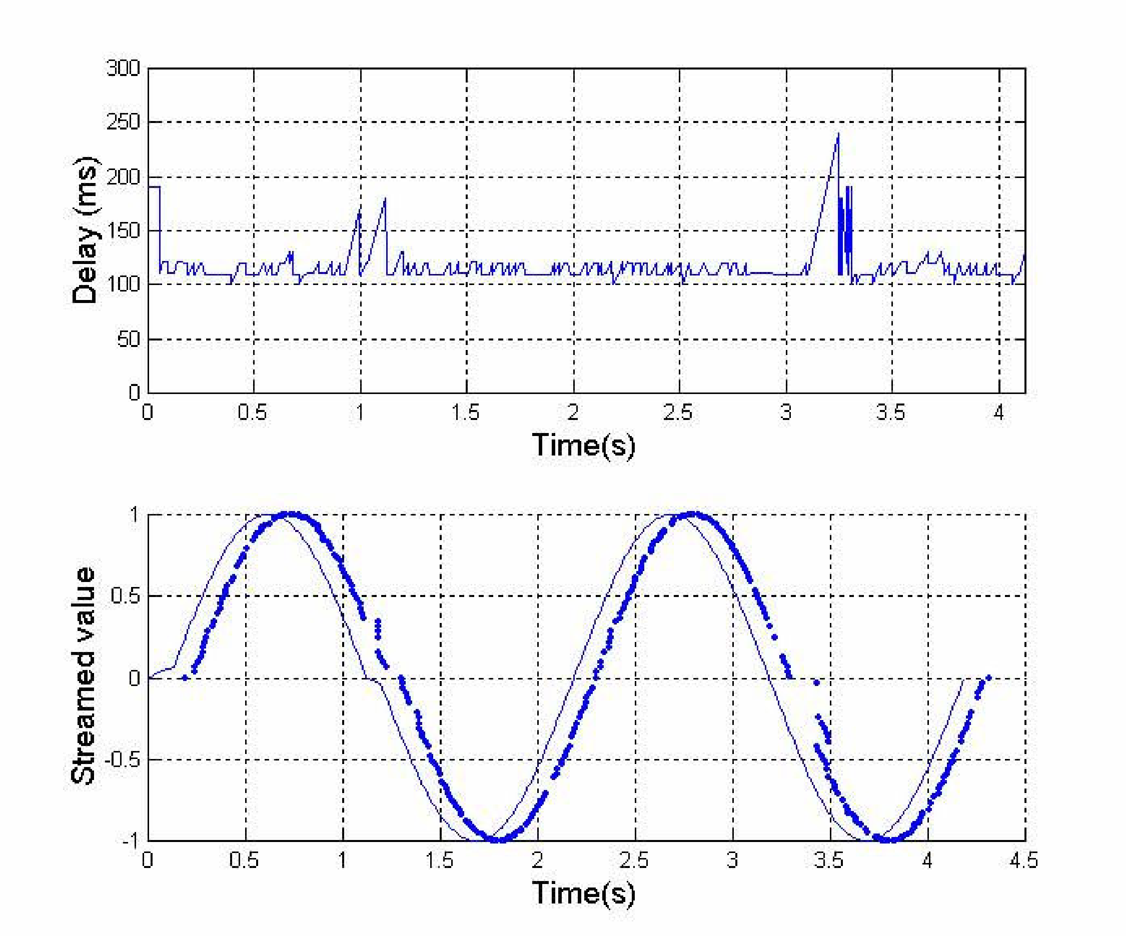


Figure 1.5: UDP Transmission Delay

1.3 Research Summary

The original research covered in this thesis can be categorized into four sections: improvement of drift control for existing algorithms, addition of adaptive filtering to existing algorithms, testing and evaluation on an asymmetric bilateral teleoperation setup, and evaluation of algorithm effectiveness via human subject testing.

The algorithm to compensate for transmission delay is based on previous work by Munir and Book [21]. They used a real master device to control a virtual slave. This research attempts to test out their algorithm on real master and slave devices, and in the process discovered flaws that made the system less robust outside virtual systems. One flaw is the use of a drift control algorithm that is sensitive to slave modeling error. A more robust drift control algorithm is used that eliminates steady state error regardless of the source of drift. The algorithm is also modified to allow the system to adapt to the slave plant-environment, hence allowing wider range of operation. These improvements on the algorithm provide for a more practical implementation on realistic devices.

The algorithm is tested on a roughly symmetric bilateral teleoperation setup where the master and slave devices are both electromechanical with similar dynamics and workspace. In this research the symmetric devices used are PHANTOM haptic devices manufactured by Sensable Technologies. To assess the versatility of the algorithm, it is implemented on a more asymmetric bilateral teleoperation where the master is electromechanical, and the slave is electrohydraulic. In this research the master used is still the PHANTOM, but the slave is a custom made Hydraulically Actuated Lifter (HAL) with a workspace of roughly five times that of the PHANTOMs. This requires additional scaling rules to be applied to insure that no actuator becomes overloaded, and no device extends beyond its workspace. Success in this setup shows that the algorithm can be modified to allow compatibility among various devices as master or slave.

Given that the success of the delay compensation algorithm may allow online application by the general public, its performance is evaluated based on human subjects with little experience in haptic teleoperation. A number of subjects were chosen from the Georgia Tech undergraduate population to provide data with statistical significance. The subjects were asked to perform several simple tasks covering various slave environments using different algorithms as well as the conventional bilateral teleoperation with no transmission delay. In addition to assessing the effectiveness of the algorithms, the experiment also provides data with which other delay compensating algorithms can be compared.

1.4 Thesis Outline

The thesis is organized into 10 chapters. The first chapter provides an introduction to the research topic while the rest of the chapters cover the following.

- **Chapter 2 (Background Research):** This chapter covers past work in transmission delay compensation in bilateral teleoperation. The majority of this chapter is used to cover wave based algorithms because the proposed algorithm in this thesis is also based on wave variables. However, representative work based on other methods is also included.
- **Chapter 3 (Wave Variable Algorithm):** The wave variable formulation forms the basis of the proposed algorithm in this research; hence a comprehensive understanding of this method is important. In this chapter, wave variable algorithm is derived starting from the telegrapher's equations. A proof of passivity of wave variables is also given.
- **Chapter 4 (Wave Based Algorithms):** One of the shortcomings of the wave variable approach is poor transient response. This chapter shows modifications to the wave variable algorithm that can overcome its shortcoming. Discussion focuses on the addition of a predictor to improve transient, drift correction to improve steady state, energy-based regulators, and a few forms of adaptive algorithms to compensate for changing environment.

- **Chapter 5 (Experimental Setup and Results):** This chapter introduces the experimental setup and the results for both symmetric and asymmetric bilateral teleoperation. Experiments were performed for free space and rigid contact environments. The significance of the results are highlighted
- **Chapter 6 (Performance Evaluation Using Human Subjects):** To evaluate the effectiveness of the algorithms, human subjects were asked to perform some simple tasks in a bilateral teleoperation setup. This chapter explains in detail the experimental procedures and results of the tests.
- **Chapter 7 (Conclusion):** This section concludes the thesis with some final remarks as well as a summary of the research contributions. Recommendations on follow-up research in Internet-based teleoperation are also given.

CHAPTER 2

BACKGROUND RESEARCH

Pure time delay in closed-loop control system had been recognized to cause stability problem since the development of classical control. This could be easily proved via the root locus method and was taught in most undergraduate level textbooks in system dynamics and control, namely Ogata [6]. In the 1960s, Ferrell published some of the first papers that explore the effect of time delay in a remote control setup with human operators in the system. In [7], he reported on the effect of inserting transmission delay between master and slave manipulators. Because the study was based on an open-loop system, stability was never an issue. In a later study [8], he reported on the effect of a similar setup except in a closed-loop system where stability was an issue. There were no algorithms implemented in his work that specifically compensated for the transmission delay. The papers noted that human operators tended to adopt a wait-and-move strategy to cope with the delay, although such a strategy severely limited the performance for large delays.

The study of the effect of transmission delay and compensation methods in teleoperation setup really took off with the proliferation of the Internet. The Internet allowed for the quick transfer of vast amount of information over large distances accessible to the general public. People began to use the Internet for multimedia transfer, gaming, e-commerce, and other applications. It was inevitable that someone would consider using it as a transmission medium for remote control tasks in real time. At first the Internet-based teleoperation was mostly used in a supervisory setup with no direct feedback. This was of course easier and safer to implement because the system was open loop where transmission delay could deteriorate performance but not system stability.

However, the study of more direct feedback control over the Internet had been carried out and in some cases implemented in existing Internet infrastructure.

2.1 Supervisory Teleoperation

This section listed some examples of actual implementation of supervisory teleoperation over the Internet. Taylor and Dalton [9] had a robot in University of Western Australia that was remote controlled via the Internet in 1994 using a JAVA based applet running on a web browser. The human operator could program reference points and send them to the robot. The robot then ran its programmed trajectory via a local control loop without further human intervention and feedback signal to the human operator.

At Georgia Tech, the Intelligent Machine Design Laboratory (IMDL) had the Hydraulically Actuated Lifter (HAL) that could be remotely operated from the Internet [10]. HAL could be considered as a two-degree-of-freedom robot similar to a forklift but without steering. Again the involvement of the human operator was supervisory in nature where he or she input control parameters and run experiments such as step input and frequency response on HAL. The operator observed the experimental process visually via a webcam. After the experiment was complete, the operator could download the data to his or her computer. This setup was designed for educational purpose where students had the option of running experiments on HAL at any time they chose from their home computer.

Jet Propulsion Laboratory had also used Internet as a communication medium in the remote control of robotic vehicles on Mars [11]. They used the Web-Interface for Telescience (WITS) as an Internet-based tool that the Mars Pathfinder for both mission operation at JPL and public outreach. Mission scientists used WITS to access data received from the Mars rover as well as plan the mission via the Internet. This was particularly important in this mission because Mars explorations via rovers were expected to last for months and even years. The mission scientists were not expected to remain in mission control center for that long, and access to control from various points around the world became vital. This tool also allowed the public to become more involved as they had similar access to the data such as videos. Of course the general public was not allowed to participate in the actual mission planning, but they were able to use the received data on Mars to plan their own mission in a simulation. For example, based on photographs taken by the rover of its surrounding, the public might plan a trajectory for the rover to take in simulation.

The research works mentioned above were all supervisory teleoperation without direct operator intervention and feedback where the slave robot had a relatively high degree of autonomy. With more direct feedback signal from the slave to generate haptic sensation for the operator, the transmission delay in the closed-loop would require some type of delay-compensation algorithm. The following sections discussed some of the algorithms being researched with particular emphasis on wave variable based algorithms.

2.2 Algorithms Based on Wave Variables

Wave variable algorithm was one of the several algorithms designed to solve the instability problem caused by transmission delay. It was a highly robust algorithm that could guarantee stability under any transmission delay but at the cost of very poor transient response. The detailed background on this algorithm was provided here since the proposed algorithm in this thesis was based on wave variables.

Most current developers of the wave variable algorithm traced the origin of the method to a paper by Spong and Anderson [12, 13]. They took a passivity approach to deal with transmission delay inspired by the telegrapher's equations that described the dynamics of the power transmission line. The ideal power transmission line model consisted of a cascade of inductors in series and capacitors in parallel. Since all the electrical elements were just energy storage elements, the system was passive and lossless. The paper attempted to model a teleoperation transmission line after the ideal power transmission line and hence inherit its passive characteristics. The transmission line was assumed to be a two-port network. The papers then proceeded to prove system stability using the scattering theory.

This idea was carried further by Niemeyer and Slotine who introduced the term wave variable [14, 15]. The wave variable was defined as the intermediate variable that is transmitted across the Internet as opposed to more conventional variables such as position, velocity, and force data. The algorithm was still based on that developed by Spong and Anderson, but they used an approach that is more intuitive to mechanical engineers. The reason for using the term wave variable was not explicitly explained in the paper, but we deduced that the term came from the fact that information was transmitted across the power transmission line in the form of energy waves. In fact, the square of the wave variable had dimensions of power. The only parameter to be adjusted in this algorithm was the wave impedance that represented the characteristic impedance of the modeled power transmission line.

Niemeyer and Slotine's wave variable algorithm was applicable to a single degree of freedom system where the wave impedance was a scalar. Munir and Book [16] then proceeded to extend this algorithm to be applicable to robots with multiple degrees of freedom where the wave impedance was a matrix. The degrees of freedom might even be coupled to each other, but the algorithm was restricted to symmetric teleoperation where the wave transforms on the master and slave sides were identical. Alise, Roberts, and Repperger [17] further extended the wave variable algorithm to include multiple-degree-of-freedom coupled system with scaling such that it could operate on asymmetric teleoperation.

2.2.1 Wave Impedance Matching Research

Although the wave variable algorithm was very robust in stability, it had very poor transient qualities. In electrical power transmission, energy transmitted from the power station to the household appliances could get reflected if the effective impedance of the power transmission line was different from that of the appliance impedance. A mechanical analogy was a transverse pulse traveling down a string that got reflected due to an abrupt change in string density. Unfortunately, the wave variable algorithm also inherited that part of the power transmission line characteristic. Depending on the slave environment, there might be significant transient oscillation due to the wave variable reflections. This transient contributed to longer task completion time for wave variables. A study by Lawn and Hannaford asked human users to perform a number of simple tasks, and it suggested that task completion time was approximately 50% greater for passivity based methods like wave variables when compared to other methods such as PD-type control [18].

Since Niemeyer and Slotine, several researchers proceeded to improve on the wave variable algorithm to take advantage of its robustness while eliminating its shortcomings. One of the active areas of research in wave variables was a method called wave impedance matching. This principle had been recognized by Niemeyer and Slotine in their original paper on wave variables [14]. It was a method to reduce the wave reflection at the teleoperation terminals by artificially matching either the master or slave impedance to the wave impedance. While this significantly improved the transient response, it severely distorted the transmitted data. Depending on whether the master or slave was position or force controlled, the velocity or force signals might be distorted. Force distortion affected teleoperation transparency; velocity distortion produced steady state error in position. Benedetti et al. proposed that using only impedance matching on the slave side could sufficiently reduce transient oscillation while reducing the steady state error in position [19]. They also proposed a tuning algorithm for the wave impedance that was dependent on the perceived transmission delay as a way to compensate for time-varying transmission delay. Later Lew and Repperger explored the use of impedance matching in teleoperation as part of a program to design maintenance robot to clean Space-Based Laser Mirror [20]. An alternative impedance matching algorithm was used on the slave side that also reduced transient oscillation with a first-order filtering effect. Steady state error in position was still nevertheless apparent in the simulation results for transmission delays of up to one second.

2.2.2 Wave Predictor Research

An alternative method to improve wave variable transient response was the use of a predictor. A predictor using wave variables was similar to a Smith predictor. It tried to compensate for transmission delay by predicting what the returning signal would be without the delay based on a model of the remote plant. The predictor method actually could work effectively on its own right without wave variables. But unfortunately this required an accurate model to insure stability. Working with wave variables, a poor predictor model was less likely to cause instability, but it could still cause steady state errors and poor transparency.

Different versions of the wave predictor algorithm had been proposed in the past. Munir and Book [21] introduced a wave predictor where the predictor was located on the master side using a model of the slave plant. The predictor worked in the wave variable domain. A Kalman filter estimated the delayed slave states and sends the data to the predictor which marched the state to the present. A correction signal was then injected into the returning wave variable. To insure passivity, a regulator based on an energy reservoir gradually incorporated the correction. The algorithm had been successfully tested for transmission delays of up to 400 ms on a real master and a virtual slave. Ganjetar et al. [22] also proposed a similar setup for the wave predictor algorithm. However their method was simpler as it did not involve a regulator to guarantee passivity. Arioui et al. [23] proposed a wave predictor algorithm with a different setup where a master model was used. They termed the prediction as "somehow prediction" because the model only included that of the master and not that of the human operator. The prediction block was placed outside of the wave variable domain, although the paper also discussed the feasibility of the predictor working in the wave variable domain. This algorithm had been tested successfully for up to one second transmission delay.

2.2.3 Wave Variable Drift Control Research

Another problem with the wave variable algorithm was that it could potentially produce steady state error. As noted in the next chapter, the wave variables consisted of a state transformation of the force and velocity. Hence it was the velocity data and not the position data that was directly encoded into the wave variable which was then transmitted across the Internet. Position data must be integrated from the velocity data transformed back from wave variables. Under such conditions like varying transmission delay where the velocity data might not be reconstructed in an orderly fashion, the position data could not be preserved causing gradual disagreement between the master and slave states.

There were several research papers that focus on solving this problem regarding wave variables. One of the first groups to address this issue was Kosuge et al. [24] who also proposed the use of a virtual time delay. Essentially all incoming data passed through a buffer that temporarily stored the data based on the maximum transmission delay duration at the moment. The size of the buffer was constantly adjusted based on the network traffic condition. This method worked but might unnecessarily deteriorate teleoperation performance because the buffer was always based on maximum delay. Munir and Book [21] included a drift control algorithm along with the wave predictor. The drift control was placed on the master side and compared the expected and actual error between the master and slave positions. If the two values were different, the drift control generated a correction to drive this difference to zero. The correction was incorporated passively via an energy-based regulator. However it was found that with an inaccurate predictor model, the expected error between master and slave might not be correct. Yokokohji et al. [25] chose to introduce a correction signal based on wave integrals. The expected wave integral was computed by integrating using time stamps. This value was compared with the actual wave integral which was computed by integrating using received time. Recall that the position data was encoded in the wave integral. If the two values were different, then a correction signal was directly injected that was directly proportional to the difference. However, this algorithm did not guarantee passivity. Yokokohji et al. [26] then proposed a modified algorithm that is similar to [25] but included the computation of the energy stored in the network. If it became apparent the drift control was generating net energy, the algorithm was switched off abruptly. Mirfakhrai and Payandeh [27] suggested using an autoregressive model as a predictor to forecast future values of the transmission delay. The predictions were used with a look-up table to tune the gain with which the wave integrals were to be fed into the system. They claimed this gain scheduling and tuning algorithm could reduce the mismatch between the master and slave positions at steady state. Zhang and Li [28] also used the wave integral setup similar to that proposed by Yokokohji et al. but included a time delay identification algorithm which was used to tune the wave integral gain.

2.2.4 Miscellaneous Wave Variable Research

There were also several research papers regarding wave variables to further extend and improve the algorithm. Carignan and Olsson [29] designed a form of wave variable algorithm that allowed two masters to control one slave. This had particularly useful applications in user training tasks where a teacher on one master could guide the student on the other master via the slave. Since the slave force was fed back to both masters, the student got the same haptic feedback during the learning process.

An unusual research into wave variables was its relationship to human motor control. Massaquoi and Slotine [30] proposed that the human cerebellum might function as a wave variable processor. A biological arm control model was formulated based on wave variables called the Wave Variable Intermediate Cerebellar Arm Control Model (WVICACM). In a later paper by Massaquoi et al. [31], the group produced a simplified version of the WVICACM.

The wave variables not only could compensate for transmission delay, it could also adjust for sampling period delays within the computation structure. Lee et al. [32] noticed that during implementation of the wave variable algorithm on a sampled-data system, a unit time delay arised due to the causality of the reflected wave. They proposed a redesign of the wave impedance such that the wave variable algorithm guaranteed stability for large delayed reflection in a sampled-time system. This algorithm was tested under free space and rigid contact environments.

Kim et al. [33] proposed a hybrid wave variable system with a passivity observer/controller. A passivity observer kept track of the energy flow into the system, and the result was used to adjust a passivity controller which was simply a damping block on the slave side. The passivity observer/controller was primarily used to insure passivity of the master and slave plants where the wave variable algorithm was used to insure passivity of the communication block. The algorithm was successfully tested for up to 200 ms transmission delay.

Niemeyer had remained active in the wave variable related research. Recently Niemeyer and his student Tanner had been researching the application of wave variables in sensing texture via high-frequency force feedback [34, 35, and 36]. They proposed injecting the environmental force into the returning wave variable after passing it through a high-pass filter while the signal sent to the slave remained unchanged. Furthermore they proved that while the communication channel with delay alone was no longer passive in this case, the overall communication channel and slave plant was passive and hence stable. During the implementation of this algorithm on Toolhandle haptic devices, they also identified some practical limitations of wave variables. One of them was the limitation of conventional low-pass filters to reduce velocity noise due to instability caused by the filter phase lag. However, they solved this problem by placing the low-pass filters in the wave domain where the filter phase lag was interpreted as another delay [37]. Another problem encountered was that constant wave impedance was tuned and optimized for only a certain degree of slave impedance. This became a problem if the slave transitioned between free space and rigid contact. To solve this problem, they developed a self-tuning algorithm that adjusted the wave impedance online depending on the slave impedance encountered [38]. Niemeyer also worked with Diolaiti to create a better implementation method of wave variables in electrical systems. They noticed that in traditional haptic rendering, amplifier bandwidth, quantization, and discretization all contributed to performance limit. Meanwhile the amplifier tried to cancel out the electric motor's dynamics even though they were beneficial to haptic rendering. Diolaiti and Niemeyer hence proposed an algorithm that incorporated the electrical system's dynamics as part of the wave variable algorithm [39]. The resultant system produced better haptic performance with simpler implementation.

2.3 Algorithms Based on Smith Predictors

There was a family of time delay compensating algorithms based on the Smith predictor, first developed by Smith as a method to compensate for transmission delay in 1957 [40]. Smith proposed the algorithm originally as a solution to chemical processes where the plant had considerable lag. A thermal system was another type of plant with considerable amount of lag. Such an algorithm was also applicable to pure time delays. The wave predictor type algorithms could be interpreted as predictor working in wave domain. This did not have to be the case; a predictor could operate on its own right. Since after the wave variables, the predictor was the most significant part of the proposed algorithm in this thesis, this section was dedicated to background work on predictor type algorithms.

The Smith predictor compensated for time delays by anticipating what the feedback signal would be without the delay. This method could be very effective but required a very accurate model to make the correction prediction. Hence much effort had been spent on developing a Smith predictor where the model could adapt to the environment. Alternatively, a predictor could be modified to be more robust and insensitive to modeling error.

Bahill [41] proposed constantly adjusting the predictor model parameters to minimize the square of the error between the model and plant outputs via a gradient method for minimization. He also suggested marching the signal sent to the plant ahead of the transmission delay to achieve zero latency tracking. He et al. [42] proposed a more indirect adaptive Smith predictor using the recursive least square algorithm to estimate the plant parameters. However, instead of using the parameters to predict the feedback signals, these parameters were used to adjust PID control gains via Ziegler-Nichols tuning method. Huang and DeBra [43] developed a set of tuning rules for Smith predictor where the model parameters to be tuned were divided into poles, gains, and delay time. However, the tuning could only be accomplished beforehand in the training phase; tuning online was impossible. Wu et al. [44], instead of trying to estimate the plant parameters, employed an approximate Smith predictor where a Pade approximation of the pure time delay was used. Although it did not guarantee stability at all times, the approximation did make the algorithm more robust. The sufficient condition for stability for this method was demonstrated by applying the Nyquist diagram.

Hashtrudi-Zaad and Salcudean [45] applied a composite adaptive control strategy to estimate the plant parameters for a Smith predictor in a bilateral teleoperation setup. The algorithm required master and slave positions, velocities, and accelerations as inputs for the parameter estimator. The same parameter estimator was also used to tune the slave controller. Gao et al. [46] included a fuzzy controller with an adaptive Smith predictor that used a lookup table to adjust the controller gains based on the error between model and plant output. Ivanova and Hadjiski [47] also used an adaptive Smith predictor based on a fuzzy estimator to adjust the model parameters. Recently there was extensive research into using neural networks in an adaptive Smith predictor such as Huang and Lewis [48], Wang et al. [49], and Smith and Hashtrudi-Zaad [50]. In most cases the proposed schematics involved neural networks on the slave side where the network weights were adjusted via back-propagation. The predictor on the master side consisted of identical neural networks whose weights were constantly updated by the slave networks. The operator input was then passed through the predictor networks to generate a model output that was used to compensate for the transmission delay. The adaptive Smith predictor based on neural networks was more robust than adaptive Smith predictors based on other methods such as recursive least squares because it could compensate for nonlinear plants. Huang and Lewis [48] also included the conditions for stability of such algorithms using Lyapunov analysis.

2.4 Teleoperation with Virtual Slave

One group of time delay compensating algorithms was based on a virtual slave to the master that was based on the actual slave plant. In this respect, they were similar to the Smith predictor. However unlike the Smith predictor, there was no direct feedback from the slave. Instead the virtual slave on the master side provided all the feedback to the human operator. To provide good transparency an accurate model was required to build the virtual slave. But unlike the Smith predictor, as long as the virtual slave model was stable, any inaccuracy in the model would not cause any overall system instability because there was no direct feedback.

Bejczy et al. [51] used a high fidelity graphics "phantom robot" that was controlled by the operator in real time without delay against a static task image. The image of the real robot was also overlaid on top of the static task image. The real robot was programmed to follow the trajectory of the "phantom robot." The algorithm was tested using a PUMA robot where the predictive display enhanced the teleoperation performance, although the paper noted that stereoscopic predictive display was necessary for general three-dimensional tasks. No haptic feedback was used in this study.

Kotoku [52] designed a teleoperation system based on a virtual polyhedral slave environment modeled after the real slave environment. When the operator pushed into a contact, the virtual slave simulated a frictionless contact feedback force without delay. Funda and Paul [53] developed a similar system except they included a procedure to create the virtual slave environment based on cameras and range scanners with the help of the human operator. This allowed the system to operate without a priori knowledge of the slave environment. They also tested their algorithms out using a PUMA robot.

Tsumaki et al. [54] included force feedback in their virtual reality based teleoperation. The teleoperation system combined velocity-level commands with some slave autonomy. With the force feedback computed based on velocity data, the system was less sensitive to geometric modeling errors. The problem of position drift was solved by using a simple correction algorithm that drove the virtual slave to the real slave position. In free space environment, the virtual slave was set to high gain mode. When near contact, the virtual slave was set to a slower optimal velocity mode along with a transition velocity zone. Furthermore, contact sensors mounted on the real slave allowed the system to switch between non-contact and contact modes. The effectiveness of the algorithm was evaluated by performing two simple tasks. One involved picking up and moving a weight with a robot. The other involved opening a model door with a robot.

Penin et al. [55] performed a rare experiment using Experimental Test Satellite 7 (ETS-VII) as the remote slave where the human operator was based on the ground station. A predictor algorithm was used based on a virtual slave on the master side to compensate for round trip transmission delays of five to seven seconds. Although ETS-VII was located at low Earth orbit where radio wave took only a fraction of a second to reach from Earth, the signal must be relayed among several ground station which took much more time. Three algorithms were tested using this setup. First a potential field of virtual forces was used to guide the operator via force feedback around the remote environment. Second a virtual slave was used to generate feedback force. The operator had the option of taking a "snapshot" of the remote slave which was used to update the states of the virtual slave to prevent drift. This made the teleoperation system more robust to virtual slave modeling error. Finally the feedback force generated by the virtual slave was displayed to the human operator. The algorithms were tested by performing three tasks: the grasping of a grapple fixture, the assembly of a truss joint, and the deployment of the truss. These tasks were practical and could be expected in a typical space construction mission. The human operator was able to accomplish all tasks successfully using all the algorithms.

2.5 PD-Type Controller with Llewellyn Stability Criterion

In 1952 Llewellyn developed the passivity-based Llewellyn stability criterion for two-port network [56]. This criterion had been used in bilateral teleoperation algorithms. Oboe and Fiorini [57] used PD-type controller on both master and slave plants. The master and slave positions were transmitted. On the master end, the difference between the master and slave positions was fed into the PD controller to create a feedback force for the operator. On the slave end, the difference in positions was fed to the controller to create the controller force for the slave plant. An impedance matrix was derived from the teleoperation system. Llewellyn's stability criterion was then imposed on the matrix to derive a set of conditions for the PD control parameters that insured stability.

Imaida et al. [58] tested this algorithm in another rare experiment using the ETS-VII as the slave robot. The round trip transmission delay was also found to be about seven seconds. Three operators, all with a background in teleoperation, were asked to perform some simple tasks. The first task was a pushing task where the operator was asked to push into a contact and maintain constant force. The second task was a slopetracing task where the operator attempted to identify a contour of the surface traced by the slave robot. The third task was a peg-in-hole task where the operator used the robot to place a peg into a hole. The final task was a slide-handle task where the operator used the robot to slide a handle in a guide rail. In all three tasks, the operator used modes of operation. First was the bilateral mode with force telemetry display where the master provided force feedback along with a display of force telemetry. Second was the bilateral mode only without force telemetry display. The final mode was the unilateral mode with force telemetry display where the master did not provide force feedback. The results indicated the stability of the algorithm. They also showed that even with such severe transmission delay, force feedback from the master could greatly enhance the operator's teleoperation performance.

2.6 Compliant Control

About a decade ago there were a few papers proposing the use of shared compliant control to compensate for transmission delay. Shared compliant control emulated a mechanical spring and damper connected virtually to the slave side robot. This was implemented by first low pass filtering the slave environmental force and then feeding it back to the position command signal coming from the master side. While originally developed to soften impacts from the robot, this algorithm had been modified to compensate for transmission delay by Kim [59]. Although no rigorous proof of stability was given, Kim was able to compute a stable range of the compliant control spring and damping constants via frequency analysis.

Kim et al. [60, 61] proceeded to test this algorithm on six human subjects with varying degree of teleoperation experience for transmission delay of up to four seconds. Comparison was made with conventional force feedback teleoperation, although only up to 500 ms delay because by then the conventional teleoperation became unstable. The subjects performed two tasks. The first task required the subject to push against a contact and maintain constant force. The second task required the subject to put a peg into a hole. The results showed the superiority of the shared compliant control over simple force feedback. Kim [62] also tested the algorithm with a predictive display that better aided the operator in time delayed teleoperation. Again the algorithm showed its superiority of simple force feedback.

More recently there was a passivity observer/controller algorithm in development that appeared to be a descendent of the compliant control. Hannaford and Ryu [63] wrote about the use of this algorithm to compensate for time delay. The algorithm assumed the bilateral teleoperator is a network where a passivity observer kept track of the net energy flow into the network. If at any moment the network started to generate energy, the passivity controller would be activated. The passivity controller was basically an adjustable damper similar to a compliant control without the spring. The damping constant was constantly adjusted to only bleed away the energy that was generated. This was called the series passivity controller with impedance causality. Alternatively one could employ the parallel passivity controller with admittance causality. The algorithm had been proven to compensate for transmission delay of 15 ms as well as for other potentially unstable situations such as hard contact.

2.7 Other Research on Time Delay Teleoperation

This section included a few other works in this field to conclude the review on background research. Koshkouei and Zinober [64] used a sliding mode algorithm to compensate for transmission delay. Two forms of sliding modes were considered: delay-independent and delay-dependent sliding modes. The former algorithm included a sliding surface that was independent of the delay information, while the later was dependent. The stability of both algorithms was shown by both rigorous mathematical proof using Lyapunov analysis and numerical simulation. Park and Cho [65] expanded on the sliding mode algorithm to better compensate for time-varying transmission delay.

This was accomplished by using a nonlinear gain that could be set independently of transmission delay. The algorithm was tested successfully via simulation and shown to be robust against varying transmission. Although also based on a model of the remote slave, the sliding mode algorithm, unlike the Smith predictor, had no predictor property, and the master feedback force consistently lagged behind the slave environmental force.

Leung et al. [66] introduced the use of H-infinity optimal control and μ-synthesis design framework to design a controller for the teleoperator that achieved stability for a prescribed time delay margin while optimizing performance specifications. The algorithm was successfully tested using simulation for up to two second transmission delay under free space and contact environment. The slave model used was that of the Shuttle Remote Manipulator System shoulder joint. The simulation showed that the controller could be tuned to stabilize the teleoperator without becoming over-conservative.

Over the past few years, Handshake VR Inc. [67] became a company that specialized in haptic software. Its product line included the Time Delay Compensation (TiDeC) toolkit based on a proprietary algorithm designed to allow stable time delayed teleoperation. The operator had the option to choose from several latency models as well as TiDeC's aggressiveness in compensating for time delay. Handshake VR Inc. had designed the TiDeC toolkit to be used for haptic devices manufactured by such companies as Sensable Technologies. Recently the toolkit had been tested as part of NASA Extreme Environment Mission Operation (NEEMO) where aquanauts lived in NASA's research station Aquarius, located 20 m underwater off Florida coast, for several days to simulate the isolation and confinement of a spaceship. The study investigated the feasibility of surgery by proxy allowing lay people to act as surgeons via telementoring with a real surgeon. This concept might become useful in long-term space missions where injured or sick astronauts might not be brought down to Earth immediately. Crew members were guided by a doctor in Florida to perform a simulated surgical operation through a virtual model. The signal was transmitted both wirelessly and via sonar underwater with an undisclosed amount of transmission delay that was compensated by TiDeC. The experiment was judged to be a success. To the best of our knowledge, this was the only commercially available toolkit designed for time delayed teleoperation.

There are many more works in the area of time delay teleoperation. This chapter covers some background research to give the reader a glimpse of the variety of methods used to solve this control problem. Special emphasis is placed on methods based on wave variables because the algorithm proposed in this thesis is also based on this method. However other algorithms also offer advantages such as drift control and the use of various adaptive algorithms to adjust predictor model parameters. Some of the tasks used by past researchers to test various time delay compensation algorithms provide ideas for the tasks used in our human experiments.

CHAPTER 3

WAVE VARIABLE ALGORITHM

The wave variable algorithm is an extremely useful passivity-based algorithm that can guarantee stability of bilateral teleoperation with varying transmission delay. This powerful method forms the background of all the work presented in this thesis. Hence it is important for the reader to have a thorough understanding of this algorithm. The chapter first introduces the telegrapher's equation, a model for power transmission lines, from which the wave variable algorithm is derived. Then it proceeds with the derivation and a discussion of the wave impedance. Finally a rigorous proof of stability is provided using energy balance methods.

3.1 Telegrapher's Equations

The ultimate inspiration for wave variable method is the telegrapher's equations that describe electrical power transmission across distance. It is obvious that a power transmission line alone does not generate energy, otherwise we would not have an energy crisis! Hence the wave variable algorithm is designed to model the power transmission line. A good understanding of the telegrapher's equations provides a good understanding of the wave variable method.

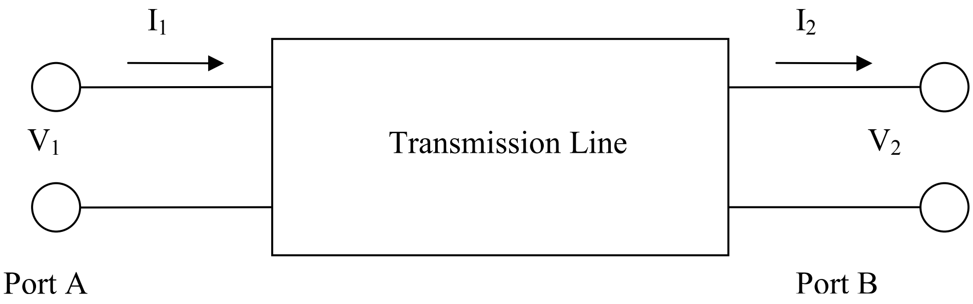


Figure 3.1: Two-Port Transmission Line Network

For the purposes of analysis, an electrical transmission line can be modeled as a two-port network as in Fig 3.1. Here the current is represented by *I*, and the voltage is represented by *V*. In the simplest case, the network is assumed to be linear, and the two ports are assumed to be interchangeable. If the transmission line is uniform along its length, then its behavior is largely described by a single parameter called the characteristic impedance. This is the ratio of the complex voltage to the complex current at any point on the line.

An approximate linear model for the transmission line was described by Oliver Heaviside [70]. The telegrapher's equations as they are called, describe the relationship between voltage and current along a conductor. Although the model consists of linear electrical elements of resistors, capacitors, and inductors, the characteristic impedance is evenly distributed along the entire transmission line. The block diagram of an infinitesimally short length of transmission line along the *x* direction is given in Fig. 3.2. In this schematic, the various electrical component representations are as follows.

- The resistivity of the transmission line along the conductor is given by the resistor

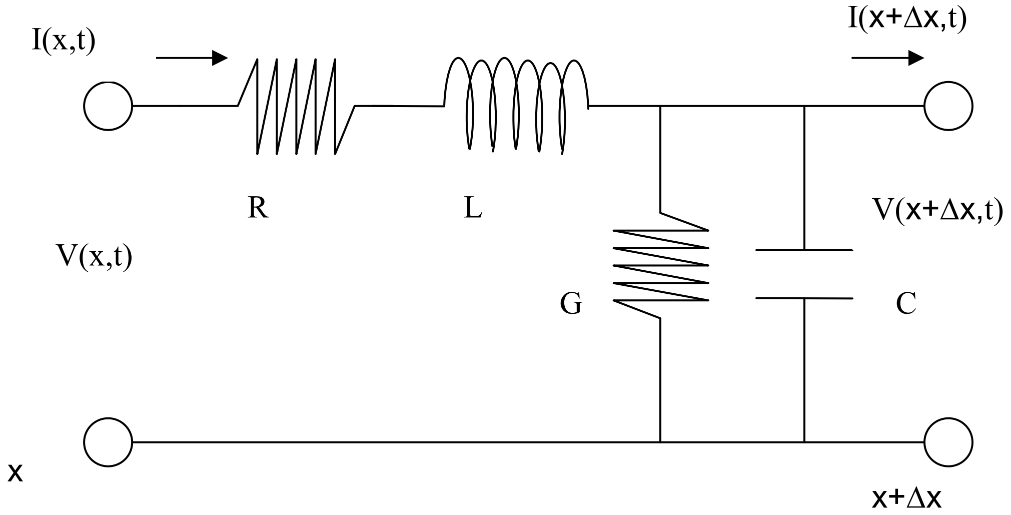


Figure 3.2: Schematic Representation of Transmission Line Elementary Component

in series *R*, with units of ohms/meter.

- The distributed inductance of the transmission line produced by magnetic field around the wire and such is given by the inductor in series *L*, with units of henries/meter.
- The distributed capacitance of the transmission line insulator between the signal and return wire is given by the capacitor in parallel *C*, with units of farads/meter.
- The resistivity of the transmission line insulator between the signal and return wire is given the resistor in parallel *G*, with units of ohms/meter.

Voltage, current, and distance along the line are *V*, *I*, and *x* respectively. If we assume an ideal lossless transmission line where no energy is dissipated, we can eliminate the effects of *R* and *G*. The telegrapher's equation then becomes

$$\frac{\partial V(x,t)}{\partial x} = -L\frac{\partial I(x,t)}{\partial t} \tag{3.1}$$

$$\frac{\partial I(x,t)}{\partial x} = -C\frac{\partial V(x,t)}{\partial t} \tag{3.2}$$

For the entire length of transmission line, the ideal lossless version of the model represented by Fig. 3.2 becomes a cascade of inductors in series and capacitors in parallel as shown in Fig. 3.3.

The goal now is to establish a relationship between V_1, I_1, V_2, and I_2. This can be

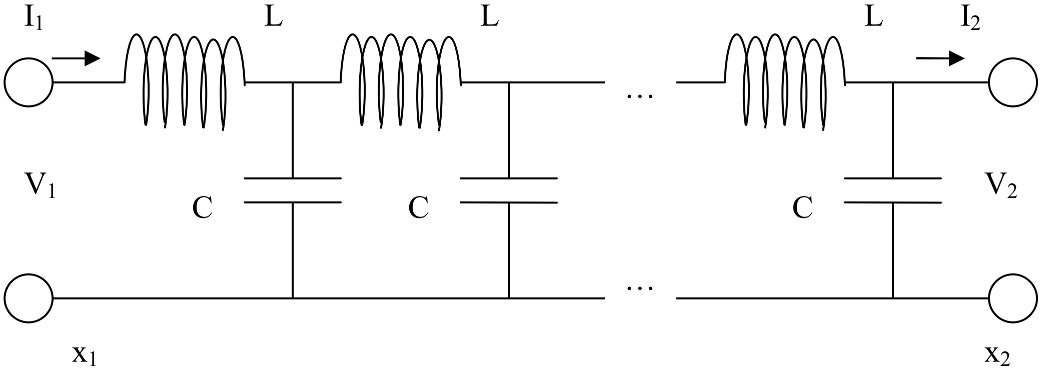


Figure 3.3: Ideal Lossless Transmission Line Circuit

done by applying the Laplace transform and imposing the port voltages and currents. The relationship between the voltages and currents at the two ports can be best described by using an impedance matrix in Laplace domain with the voltages on one side and the currents on the other. The impedance matrix is derived in [70] as

$$\begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix} = \begin{bmatrix} Z_0 \coth \dfrac{sl}{v_0} & Z_0 \operatorname{cosech} \dfrac{sl}{v_0} \\ Z_0 \operatorname{cosech} \dfrac{sl}{v_0} & Z_0 \coth \dfrac{sl}{v_0} \end{bmatrix} \begin{bmatrix} I_1(s) \\ -I_2(s) \end{bmatrix} \tag{3.3}$$

The length of the transmission line is given by *l*. *Z₀* is the characteristic impedance of the transmission line while v_0 is the speed of transmission. Both values are related to the distributed capacitance and inductance by

$$Z_0 = \sqrt{\frac{L}{C}} \tag{3.4}$$

$$v_0 = \frac{1}{\sqrt{LC}} \tag{3.5}$$

Notice that the transmission delay is simply the transmission distance divided by the speed of the transmission.

As energy travels down a power transmission line and hits one of the ports, some of the power will get transmitted while some will get reflected. We can easily see this by using a mechanical analogy using transverse pulses traveling down a string in tension. In such case, the string tension force in N is analogous to the inverse of transmission capacitance, and the string density in kg/m is analogous to transmission inductance. We notice that increasing and decreasing the string density would effectively increase and decrease the analogous characteristic impedance. Consider a string that is fixed on one end which is analogous to an open circuit at one of the ports. When an incident pulse reaches the end, the pulse is completely reflected and inverted. Consider another case where a string end is free to move vertically which is analogous to a short circuit at one of the ports. When an incident pulse reaches the end, the pulse is also completely

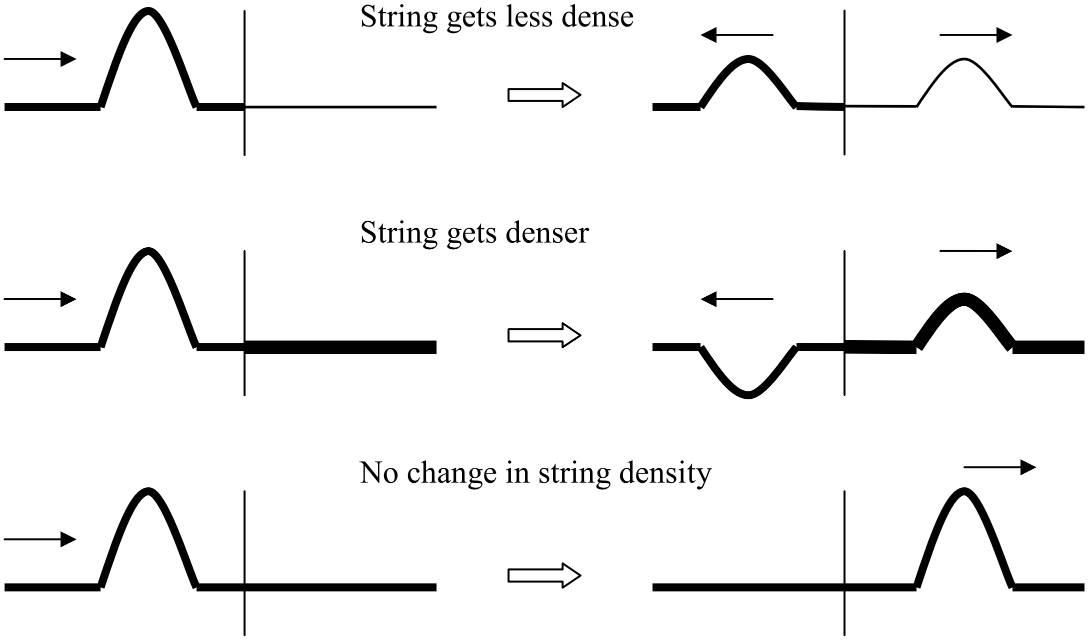


Figure 3.4: Mechanical Analogy of Transmission Line Boundary

reflected, but without the inversion. In intermediate cases where the pulse meets a string of different density of the original string, part of the pulse is transmitted, and part is reflected. If string gets denser, the reflected pulse is inverted; otherwise the reflected pulse is not inverted. The only way to insure total transmission is if the string density does not change. Figure 3.4 provides a summary of the situation.

In relationship to the power transmission line, this analogy suggests that an incident energy pulse traveling along the line is bound to get reflected at a port unless the effective impedance connecting the port terminals is exactly the same as the characteristic impedance of the transmission line. This property contributes to the poor transient response of the wave variable method, whose derivation from the transmission line impedance matrix will be shown in the next section.

3.2 Wave Variable Derivation

The goal of the wave variable method is to model the teleoperation transmission line after the power transmission line to acquire its lossless and passive characteristics.

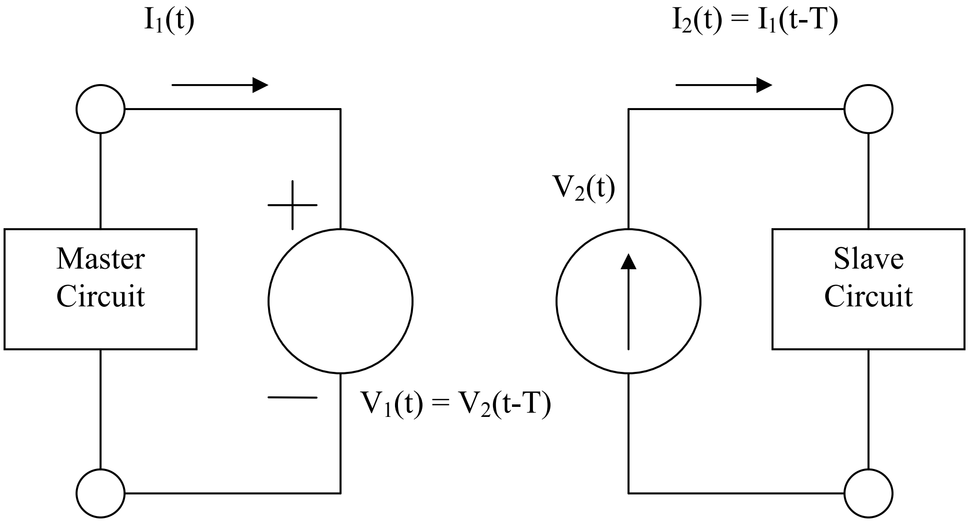


Figure 3.5: Circuit Diagram for Teleoperation Transmission Line

However this is not straight-forward because the properties of the teleoperation transmission line are different from that of the power transmission line.

Figure 3.5 shows the circuit of a typical teleoperation transmission line that can be compared to a power transmission line circuit in Fig. 3.3. For consistency's sake, the signals sent represent voltage and current data. Of course this is analogous mechanically to force and velocity data respectively. Here the diagram really consists of two circuits coupled to each other. On the right hand side there is a current source that generates a current equal to that of the current on the left hand side after a delay of time *T*. On the left hand side there is a voltage source that generates a voltage equal to that of the voltage

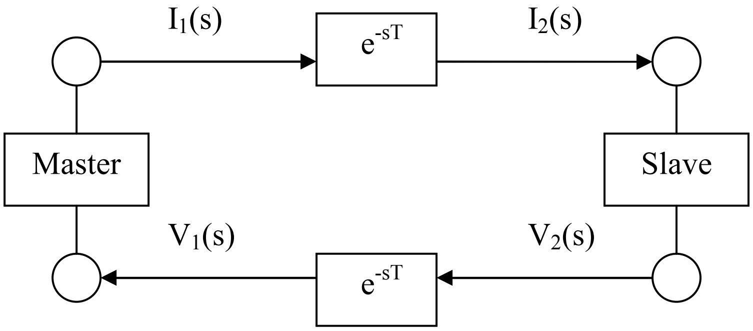


Figure 3.6: Block Diagram of Teleoperation Transmission Delay

on the right hand side after a delay of time *T*. Each terminal is also connected to a plant, closing the loop. We can easily see that the relations between the voltages and currents on both ports are

$$V_1(t) = V_2(t-T) \tag{3.6}$$

$$I_2(t) = I_1(t-T) \tag{3.7}$$

We can interpret this circuit as a flow of voltage and current information. A representative block diagram is shown in Fig. 3.6.

A teleoperation transmission line consists nothing more than pure delays for signals traveling between the operator and remote sites. From a transmission delay's perspective, the most significant difference between a teleoperation and power transmission line is that in the former case, the delay is lumped into a single pure delay term. In the later case, the delay is evenly distributed throughout the transmission line via the cascade of capacitors and inductors. The pure delay terms are inherent parts of the teleoperation transmission line and can not be arbitrarily eliminated. In other words, we can not model after a power transmission line simply by deleting the pure delay blocks and substituting the capacitor-inductor cascade of Fig 3.3. Instead we must include the pure delay elements in a model that is mathematically identical to that of the power transmission line.

To model after the power transmission line, we begin with the impedance matrix in Eq. (3.3). The equations are reshuffled such that the voltage and current corresponding to port one is on left hand side and those of port two is on the right hand side. The result is a type of hybrid matrix because the matrix elements contain both impedance and admittance terms. The hybrid matrix is

$$\begin{bmatrix} V_1(s) \\ I_1(s) \end{bmatrix} = \begin{bmatrix} \cosh \dfrac{sl}{v_0} & Z_0 \sinh \dfrac{sl}{v_0} \\ \dfrac{1}{Z_0} \sinh \dfrac{sl}{v_0} & \cosh \dfrac{sl}{v_0} \end{bmatrix} \begin{bmatrix} V_2(s) \\ I_2(s) \end{bmatrix} \tag{3.8}$$

Notice that the transmission line distance *l* divided by the transmission speed v_0 is equal to the transmission delay *T*. Since a pure delay term is inherent in a teleoperation transmission delay, one must isolate the pure delay terms from the hybrid matrix in Eq. (3.8). These terms are embedded in the hyperbolic functions. Fortunately, it turns out that the pure delay terms are the eigenvalues of the hybrid matrix. An eigen-decomposition yields

$$\begin{bmatrix} \cosh\frac{sl}{v_0} & Z_0\sinh\frac{sl}{v_0} \\ \frac{1}{Z_0}\sinh\frac{sl}{v_0} & \cosh\frac{sl}{v_0} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2Z_0}}{2} & -\frac{\sqrt{2Z_0}}{2} \\ \frac{1}{\sqrt{2Z_0}} & \frac{1}{\sqrt{2Z_0}} \end{bmatrix} \begin{bmatrix} e^{sT} & 0 \\ 0 & e^{-sT} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2Z_0}}{2} & -\frac{\sqrt{2Z_0}}{2} \\ \frac{1}{\sqrt{2Z_0}} & \frac{1}{\sqrt{2Z_0}} \end{bmatrix}^{-1} \tag{3.9}$$

In addition to isolating the pure delay terms, the eigenvectors are independent of time. Now it is possible to introduce intermediate variables such that only these values undergo pure delay. These intermediate variables are defined as

$$\begin{bmatrix} V_1(s) \\ I_1(s) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2Z_0}}{2} & -\frac{\sqrt{2Z_0}}{2} \\ \frac{1}{\sqrt{2Z_0}} & \frac{1}{\sqrt{2Z_0}} \end{bmatrix} \begin{bmatrix} U_1(s) \\ W_1(s) \end{bmatrix} \tag{3.10}$$

$$\begin{bmatrix} V_2(s) \\ I_2(s) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2Z_0}}{2} & -\frac{\sqrt{2Z_0}}{2} \\ \frac{1}{\sqrt{2Z_0}} & \frac{1}{\sqrt{2Z_0}} \end{bmatrix} \begin{bmatrix} U_2(s) \\ W_2(s) \end{bmatrix} \tag{3.11}$$

$$\begin{bmatrix} U_1(s) \\ W_1(s) \end{bmatrix} = \begin{bmatrix} e^{sT} & 0 \\ 0 & e^{-sT} \end{bmatrix} \begin{bmatrix} U_2(s) \\ W_2(s) \end{bmatrix} \tag{3.12}$$

The values Z_0 and *T* are the characteristic impedance and transmission delay respectively of the modeled power transmission line. We can not arbitrarily change the transmission delay, but the characteristic impedance is an adjustable value. It is usually desirable to match the characteristic impedance to those at the port terminals. This will be discussed later.

The intermediate terms *U* and *W* are called wave variables, and the eigenvector matrices in Eq. (3.10) and (3.11) are the wave transform matrices. The *U* wave flows from left to right, and the *W* wave flows from right to left. The subscript *1* indicates the left side, and the subscript *2* indicates the right side. They are simply a state transformation of the voltage and current variables. Since the wave transform matrix is invertible, one can easily transform back and forth between the two sets of variables. It is also possible to reshuffle the equations such that the one of the inputs is a wave variable while the other is either current or voltage. In fact for reasons of causality this is often the type of transformation performed. Furthermore, we can multiply the wave transform matrix by any scalar as long as it is done on both sides. In this sense, we can say that the wave variables are not unique. However the values in Eq. (3.10) and (3.11) are used because they generate a tidy power flow variable that is equal to the square of the wave variable. This is shown later.

Suppose we desire to send current data from left to right, while the right responds by sending voltage data back to the left as in Fig. 3.6. Consider that the wave transform is employed such that instead of current and voltage data, wave variables are passed through the transmission line. In this case, the wave transform on the left side must transform the current I_1 and inbound wave variable W_1 into voltage V_1 and outbound wave variable U_1. The wave transform on the right side must transform the voltage V_2 and inbound wave variable U_2 into current I_2 and outbound wave variable W_2. The wave transformation matrices then becomes

$$\begin{bmatrix} U_1(s) \\ V_1(s) \end{bmatrix} = \begin{bmatrix} \sqrt{2Z_0} & -1 \\ Z_0 & -\sqrt{2Z_0} \end{bmatrix} \begin{bmatrix} I_1(s) \\ W_1(s) \end{bmatrix} \tag{3.13}$$

$$\begin{bmatrix} I_2(s) \\ W_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{\sqrt{2Z_0}}{Z_0} & -\dfrac{1}{Z_0} \\ 1 & -\dfrac{2}{\sqrt{2Z_0}} \end{bmatrix} \begin{bmatrix} U_2(s) \\ V_2(s) \end{bmatrix} \tag{3.14}$$

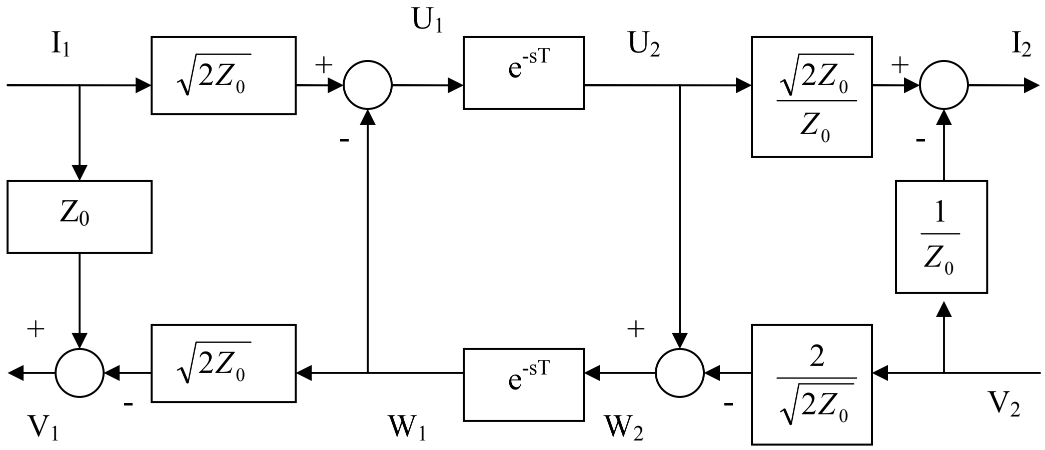


Figure 3.7: Wave Variables with Symmetric Transmission Delay

The above equations can be organized into a block diagram form that yields Fig. 3.7 which is a representation of the wave variable algorithm. We must stress that Fig. 3.7 is mathematically identical to the power transmission line circuit in Fig. 3.3 with the voltage on the right side and current on the left side as inputs. Solving for Fig. 3.7 with I_1 and V_2 as inputs and I_2 and V_1 as outputs, we get

$$\begin{bmatrix} V_1(s) \\ -I_2(s) \end{bmatrix} = \begin{bmatrix} Z_0 \tanh(sT) & \operatorname{sech}(sT) \\ -\operatorname{sech}(sT) & \frac{1}{Z_0}\tanh(sT) \end{bmatrix} \begin{bmatrix} I_1(s) \\ V_2(s) \end{bmatrix} \qquad (3.15)$$

Reshuffling the power transmission line equation of Eq. (3.3) yields an identical equation. Notice that if we assume a steady state where the input current and voltage are constant, then the transmission delay does not matter anymore. In this case the output current at port one is the same as the input current at port two, and the same applies to the voltage transmission. This shows that the wave variable method ideally produces no steady state error. However, if we use the mechanical analogy where the velocity is used instead of current, we notice that there may be position drift over time. Wave variable only encodes the velocity while the integral of wave variable encodes the position data. This implies that position may not be preserved using wave variables where any small error in the wave variable transmission will accumulate over time. The error in wave variable transmission may have many sources such as variable transmission delay and imperfections in the implementation of the algorithm and variations of the algorithm.

Figure 3.7 shows current and voltage input on the left and right port terminals respectively. Notice it is possible to modify the ports to accept current or voltage at both terminals and produce a symmetric bilateral teleoperation system. If both port terminals accept current and output voltage, then the equivalent equation is the impedance matrix as shown in Eq. (3.3). If both port terminals accept voltage and output current, then the equivalent equation is the admittance matrix or the inverse of the impedance matrix.

This concludes the derivation of the wave variable algorithm which can be summarized as an attempt to turn the teleoperation transmission line into the power transmission line by mathematical means to inherit its passivity characteristics. The original paper on wave variables by Slotine and Niemeyer [14] set the notation convention where the wave variables are represented by *u* and *v*, and the wave impedance is represented by *b*. Previously the letter *w* was used to avoid confusion with *v* for voltage. From this point onwards, this paper will switch from electrical to mechanical system variables where force is used instead of voltage, and velocity is used instead of current. The notation on wave variables and wave impedance will also be changed to follow Slotine and Niemeyer's notation.

3.3 Stability Analysis Using Scattering Operator

Spong and Anderson [12] originally used the scattering operator to prove stability of the wave variable algorithm. This section provides a reproduction of their proof.

First a definition of passivity is given for a typical two-port transmission line network as shown in Fig 3.1, using force instead of voltage and velocity instead of current. We assume that at port one, if work is done on the network then force and velocity have the same sign. At port two, force and velocity have the same sign if work is done by the network. The difference in notation for different ports is reflected by the negative sign on velocity at port two. Assuming that there is no energy initially stored in the two-port network, the net energy flow into the network is

$$\underline{F} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{3.16}$$

$$\underline{\dot{x}} = \begin{bmatrix} \dot{x}_1 \\ -\dot{x}_2 \end{bmatrix} \tag{3.17}$$

$$E_{net}(t) = \int_0^t \left(\underline{F}^T \underline{\dot{x}}\right) d\tau = \int_0^t \left(F_1\dot{x}_1 - F_2\dot{x}_2\right) d\tau \tag{3.18}$$

For the network to be passive, the energy going into the network must be greater than or equal to the energy coming out of the network as time goes to infinity. In other words Eq. (3.18) must be greater than or equal to zero as time goes to infinity, or

$$\int_0^\infty \left(\underline{F}^T \underline{\dot{x}}\right) d\tau \geq 0 \tag{3.19}$$

If this statement is false, then we can assume that energy is generated by the network since more energy is coming out than going in over time. If the integral equals to zero, then the network is lossless because all the energy going into the network comes out at the end.

The passivity condition in Eq. (3.19) can also be applied to networks with more than two ports or just one port. There is no limit on the size of the force and velocity vectors defined by Eq. (3.16) and Eq. (3.17) respectively. We simply need to keep the velocity signs consistent with the power flow direction at each port. In addition if we assume there is initial energy stored in the network, then the left side of the passivity condition in Eq. (3.19) will include an additional initial energy storage term added to it. This modification states that for passivity the total energy flow out of the network has to be less than or equal to the total energy flow into the network and the energy initially stored in the network.

To prove stability, Spong and Anderson [12] used the scattering operator *S(s)* that is defined as

$$\underline{F}(s)-\underline{\dot{x}}(s)=S(s)[\underline{F}(s)+\underline{\dot{x}}(s)] \tag{3.20}$$

The scattering operator can be readily derived from the two-port relationship of the network in the form of an impedance, admittance, or hybrid matrix.

We can show that if the norm of the scattering operator *S(s)* is less than one for all *s*, then the network satisfies the condition for passivity. By definition, if the value is less than one, then

$$\left\|\underline{F}(s)-\underline{\dot{x}}(s)\right\| \leq \left\|\underline{F}(s)+\underline{\dot{x}}(s)\right\| \tag{3.21}$$

Squaring both sides of the inequality yields

$$\left\|\underline{F}(s)+\underline{\dot{x}}(s)\right\|^2-\left\|\underline{F}(s)-\underline{\dot{x}}(s)\right\|^2 \geq 0 \tag{3.22}$$

Expanding and then integrating Eq. (3.22) over time leads to the condition for passivity given as

$$\begin{aligned}&\int_0^\infty\left[(\underline{F}+\underline{\dot{x}})^T(\underline{F}+\underline{\dot{x}})-(\underline{F}-\underline{\dot{x}})^T(\underline{F}-\underline{\dot{x}})\right]\\&=4\int_0^\infty\left(\underline{F}^T\,\underline{\dot{x}}\right)dt \geq 0\end{aligned} \tag{3.23}$$

If the norm of the scattering operator *S(s)* is one, then the system is lossless.

Using the scattering operator, we can show that pure transmission delay as shown in Fig. 3.6 is unstable. With pure transmission delay in both directions, the scattering matrix can be computed as

$$S(s)=\begin{bmatrix}-\tanh(sT) & \operatorname{sech}(sT)\\ \operatorname{sech}(sT) & \tanh(sT)\end{bmatrix} \tag{3.24}$$

To compute the norm of the scattering operator, we find the square-root of the maximum eigenvalue of S^HS where S^H is the transpose conjugate of *S*,

$$\lambda_{\max}\left(S^H S\right) \\ = \left[\tan^2(\omega T) + \sec^2(\omega T)\right] + \frac{1}{2}\sqrt{3\tan^4(\omega T) + 10\tan^2(\omega T)\sec^2(\omega T) + 3\sec^2(\omega T)} \tag{3.25}$$

The norm is greater than one for any value of ω greater than zero, with the maximum eigenvalue going to infinity when *ω* approaches *π/2*. This proves that a bilateral teleoperation with pure delays in the transmission line produces a non-passive system and is therefore unstable.

We can also use the scattering operator to show the passivity resulting from the wave variables based on Fig 3.7. For simplicity, the wave impedance is taken to be one. The scattering operator can then be solved as

$$S(s) = \begin{bmatrix} 0 & e^{-sT} \\ e^{-sT} & 0 \end{bmatrix} \tag{3.26}$$

The norm of the scattering operator can be computed to be one. This suggests that the wave transform method is not only passive but also lossless. This agrees with the power transmission line model on which the wave variable method is based. Recall that the power transmission is modeled by a cascade of capacitors and inductors, both of which are energy storage elements. Without any dissipative electrical elements such as resistor, the power transmission line and therefore the wave transform method is lossless.

Although the proof via scattering operator produces the correct results, we must note that the method is not flawless from an engineering perspective. From Eq. (3.20), we can see that the definition of the scattering operator requires addition and subtraction of force and velocity. This produces a problem where the dimensions do not match since we can not add force and velocity and also calls the dimensions of the scattering operator into question. The only reason the proof still works is because when deriving the passivity condition from the scattering operator in Eq. (3.23), the terms with inconsistent units such as force squared and velocity squared would cancel each other. This leaves only the power terms that are then integrated over time to produce the energy balance.

Hence although the proof is mathematically sound, it makes little intuitive sense. In the following section, proof of passivity of the wave variable method will be shown without the use of the scattering operator.

3.4 Stability Analysis Using Energy Balance

It is also possible to perform a stability analysis without using the scattering operator. We simply need to directly substitute into the net energy flow equation of Eq. (3.19) and observe the sign of the integral.

First we will prove the potential for instability of a conventional teleoperation transmission line as shown in Fig. 3.6 via the passivity condition. Making the corresponding substitution into Eq. (3.19), we get the net energy flow into the network defined as

$$E_{net}(t)=\int_0^t \left[F_2(\tau-T)\dot{x}_1(\tau)-F_2(\tau)\dot{x}_1(\tau-T)\right]d\tau \tag{3.27}$$

Again we are assuming that there is no initial energy stored in the network. It is easy to see from Eq. (3.27) that there is nothing to guarantee the integral will be positive as time goes to infinity. In fact we can always phase the force input at port two and velocity input at port one such that the network is always generating power. Here we will provide an example of an input set such that the network is not stable.

Consider the set of inputs for force at port two and velocity at port one to be

$$F_2(t)=\sin\left[\frac{\pi}{4T}(t-T)\right] \tag{3.28}$$

$$\dot{x}_1(t)=\sin\left(\frac{\pi}{4T}t\right) \tag{3.29}$$

Recall that *T* is the transmission delay. Force and velocity inputs are sine waves with a period that is eight times that of the transmission delay. Of course these values are all delayed by the transmission delay after crossing the transmission line. The force input on port two is delayed by the transmission delay such that it is in phase with the velocity data from port one after it crossed the transmission line to port two. Substituting Eq. (3.28) and (3.29) into Eq. (3.27), we get

$$\begin{aligned} E_{net}(t) &= \int_0^t \left[\sin\left(\frac{\pi}{4T}\tau\right)\sin\left(\frac{\pi}{4T}\tau - \frac{\pi}{2}\right) - \sin^2\left(\frac{\pi}{4T}\tau - \frac{\pi}{4}\right)\right] d\tau \\ &= -\int_0^t \frac{1}{2} d\tau = -\frac{t}{2} < 0 \end{aligned} \tag{3.30}$$

At this point it is apparent that the net energy flow into the network is always negative for any time. Equation (3.30) demonstrates how it is possible intentionally drive a conventional network unstable using only bounded inputs at the two ports.

We must stress that having pure transmission delay in the teleoperation transmission line does not immediately imply overall system instability. The result mentioned above only shows how it is always possible to find controllers and plants that can produce sets of inputs to the transmission line to cause instability. The controllers and plants themselves may be stable with bounded inputs and bounded outputs. It is possible to carefully design controllers that can compensate for the potential instability caused by a conventional transmission line. Work has been done towards this goal, such as using PD-type controllers and Llewelyn's stability criterion. However, it may be desirable to modify the transmission line so that it guarantees stability under any passive controllers and plants, and this is what the wave variable method can accomplish.

We can prove that the wave variable algorithm guarantees passivity by doing the same stability analysis and keeping track of the power flow into the network. This time the network is defined by Fig. 3.7 and the equations defined by Eq. (3.10), (3.11), and (3.12). In convention with wave variable tradition and in the mechanical engineering analogy, the current is replaced by velocity, voltage by force, wave variables with *u* and *v*, and wave impedance with *b*. The power flow into the network from port one in terms of wave variables is

$$\begin{bmatrix} F_1 \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2b}}{2} & -\frac{\sqrt{2b}}{2} \\ \frac{1}{\sqrt{2b}} & \frac{1}{\sqrt{2b}} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \tag{3.31}$$

$$F_1 \dot{x}_1 = \frac{1}{2}\left(u_1^2 - v_1^2\right) \tag{3.32}$$

We can also do the same with the power flow from the network from port two and get

$$\begin{bmatrix} F_2 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2b}}{2} & -\frac{\sqrt{2b}}{2} \\ \frac{1}{\sqrt{2b}} & \frac{1}{\sqrt{2b}} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \tag{3.33}$$

$$F_2 \dot{x}_2 = \frac{1}{2}\left(u_2^2 - v_2^2\right) \tag{3.34}$$

Substituting Eq. (3.31) and (3.33) into the energy balance equation of Eq. (3.18), and we get the net energy flow into the network in terms of wave variables only, defined as

$$E_{net}(t) = \frac{1}{2}\int_0^t \left[u_1^2(\tau) - v_1^2(\tau) - u_2^2(\tau) + v_2^2(\tau)\right] d\tau \tag{3.35}$$

Recall that the only difference between the wave variables at different sides is the transmission delay as indicated by Eq. (3.12). Making the substitutions yields

$$\begin{aligned} E_{net}(t) &= \frac{1}{2}\int_0^t \left[u_1^2(\tau) - v_2^2(\tau - T) - u_1^2(\tau - T) + v_2^2(\tau)\right] d\tau \\ &= \frac{1}{2}\int_0^t \left[u_1^2(\tau) - u_1^2(\tau - T)\right] d\tau + \frac{1}{2}\int_0^t \left[v_2^2(\tau) - v_2^2(\tau - T)\right] d\tau \end{aligned} \tag{3.36}$$

We can combine the limits of the integral in Eq. (3.36) to get

$$E_{net}(t) = \frac{1}{2}\int_{t-T}^t u_1^2(\tau) d\tau + \frac{1}{2}\int_{t-T}^t v_2^2(\tau) d\tau \geq 0 \tag{3.37}$$

At this point we can conclusively show that the term in Eq. (3.37) is always greater than or equal to zero because the integral terms are always positive regardless of the signs of the wave variables. Since Eq. (3.37) describes the net energy flow into the network, it

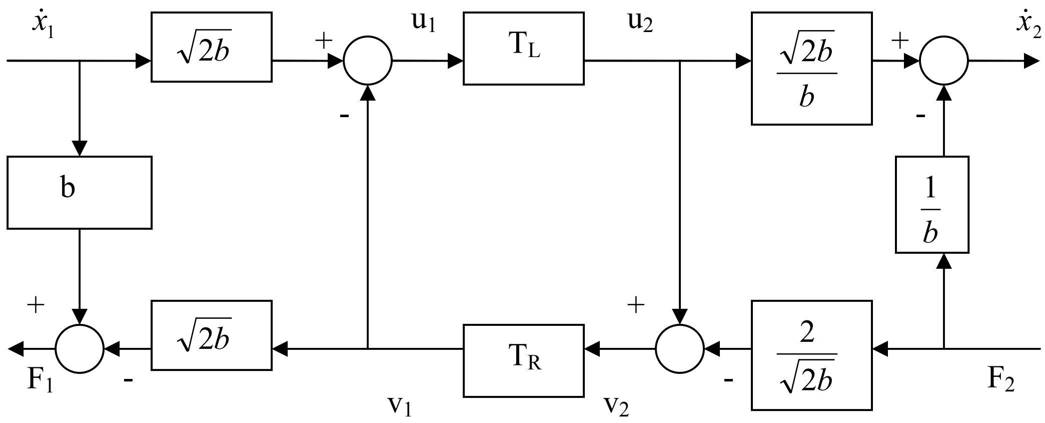


Figure 3.8: Wave Variable with Asymmetric Transmission Delay

suggests according to the definition of passivity in Eq. (3.19) that the network is now always a passive element.

The proof above assumes the same transmission delay in both directions. Consider a case now where the transmission delay is different in different directions. The block diagram of Fig. 3.8 is similar to that of Fig 3.7, except that the transmission delay is now asymmetric. If we perform a similar energy balance in terms of wave variables, we get

$$E_{net}(t) = \frac{1}{2}\int_{t-T_L}^{t} u_1^2(\tau)d\tau + \frac{1}{2}\int_{t-T_R}^{t} v_2^2(\tau)d\tau \geq 0 \tag{3.37}$$

Notice that the only thing that is different between the above equation and Eq. (3.36) are the differences in the lower limits of the integrals. Nevertheless this does not change the sign of the net energy flow, indicating that asymmetric transmission has no effect on passivity in wave variables. In fact even variable transmission delay has no effect on passivity in wave variables.

Notice that wave variables have the units of square root of power. Hence we can view them as power flow across the transmission line, and the pure transmission delay becomes temporary energy storage elements. The key word here is *temporary* because as seen from Eq. (3.37), the power is only integrated over the duration of the transmission delay. If the power flow, represented by square of wave variable, becomes zero for longer than the duration of the transmission delay, all the energy that is stored in the transmission delay is delivered to the output. This also shows that the wave variable algorithm is a lossless algorithm.

We have mentioned before that the wave variables are not unique since the wave transform matrix can be multiplied by a scalar as long as this is done on both sides of the transmission line. Suppose the wave transform matrices of Eq. (3.31) and (3.33) are scaled by a constant c. The energy balance equation of Eq. (3.37) becomes

$$E_{net}(t)=\frac{c^2}{2}\int_{t-T_L}^{t}u_1^2(\tau)d\tau+\frac{c^2}{2}\int_{t-T_R}^{t}v_2^2(\tau)d\tau\geq 0 \tag{3.37}$$

We can see from Eq. (3.37) that simply scaling the wave transform matrix does not change the passivity guarantee. However, the units of the wave variable may not be the square root of power anymore. Instead the product of the scaling constant and wave variable has the units of square root of power.

3.5 Wave Impedance Matching

Recall from the discussion on the power transmission line that power transmitted along the line is prone to be reflected at the ports on either end. Since the wave variable algorithm is mathematically identical to that of the power transmission line, we can expect to observe a similar effect using this method. If we look at Eq. (3.13) and (3.14), we can see that the outbound wave variable computed by the wave transform matrix is also a function of the inbound wave variable in addition to being a function of either force or velocity. In other words, part of the incoming wave is bounced back in the outgoing wave. This is true at both ports, regardless whether the wave transform matrix accepts force or velocity. In a teleoperation setup, this is highly undesirable because it corrupts useful data flow by producing oscillations. Similar to the analogy of transverse wave traveling on a string, it is possible to eliminate this problem by adding terminal

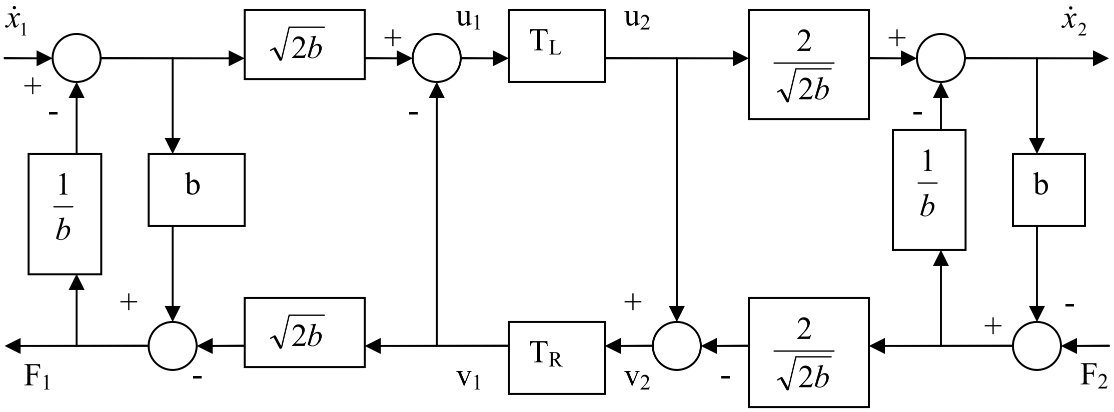


Figure 3.9: Wave Impedance Matching

elements at the ports such that the impedances at the terminals match the wave impedance.

Consider Fig. 3.9 that is a modified version of the block diagram of Fig. 3.8. Here the force data from the wave transform on the left-hand side is fed into an admittance element, and its output is subtracted from the velocity data flow into the wave transform. A similar modification is made on the right-hand side where the velocity data is fed into an impedance element, and its output is added into the force data flow into the wave transform. The value of admittance and impedance are chosen to match that of the wave impedance *b*. If we solve for the outgoing wave variable u_1 on the left port and its counterpart v_2 on the right port, we get

$$u_1 = \sqrt{\frac{b}{2}}\dot{x}_1 \tag{3.38}$$

$$v_2 = \frac{1}{\sqrt{2b}}F_2 \tag{3.39}$$

We can see now that the outgoing wave variable consists of only input velocity and force terms and no longer include the incoming wave variable. In other words, the wave variables are no longer reflected at the ports.

While the transient response with matched wave impedance is now significantly improved, we are left with tracking errors. We can see this by solving for the output force F_1 at the left port as a function of the input force F_2 at the right port, and the output velocity $\dot{x}_2$ at the right port as a function of the input velocity $\dot{x}_1$ at the left port. The results can be shown as

$$F_1(t) = \frac{1}{2} F_2(t - T_R) + \frac{b}{2} \dot{x}_1(t) \tag{3.40}$$

$$\dot{x}_2(t) = \frac{1}{2} \dot{x}_1(t - T_L) - \frac{1}{2b} F_2(t) \tag{3.41}$$

We can see how with matched wave impedance, the output velocity and force are scaled by half and have offsets that are functions of input velocity and force. Even at steady state where input velocity and force are constant, the output velocity and force would no longer match their corresponding input. In case of force transmission, this implies the perceived momentum between terminal one and two will be different. The same applies to velocity transmission. However since an offset is also introduced to the velocity, when we integrate velocity to obtain position, we also experience position drift. In most teleoperation tasks where kinematic accuracy is vital, position drift between the two ports of a teleoperation setup is highly undesirable.

CHAPTER 4

MODIFIED VERSION OF WAVE VARIABLE ALGORITHM

The wave variable method is an extremely useful passivity-based algorithm that can guarantee stability of bilateral teleoperation with varying transmission delay, but the algorithm suffers from poor transient response due to wave reflections at the terminals of the transmission line. Matching wave impedances at the terminals using passive elements can eliminate these reflections and improve transient response, but this modification distorts the transmitted information that can result in position drift and force misperception.

This thesis focuses on an alternative method to improve transient response while taking advantage of the stability guarantee of the wave variable algorithm. The scheme incorporates a predictor operating in wave variable domain, a drift control algorithm, and an adaptive filter that anticipates the remote environment. Wave prediction can improve performance by anticipating the value of the returning wave variable. However, it requires a model from which to base the prediction, and inaccurate model generates incorrect prediction that results in errors that accumulates to position drifts. A drift control algorithm can eliminate drift caused by poor predictor models and other sources of steady state error such as variable transmission delay. Furthermore, given a dynamic remote environment the predictor would need to adapt accordingly. Hence mode-switching and recursive least square (RLS) adaptive algorithms are used to estimate the states of the remote environment and send the necessary adjustments to the predictor.

4.1 Smith Predictor

Before describing the wave predictor algorithm, it is necessary to explain the setup and nature of the Smith predictor on which the wave predictor is based. The Smith

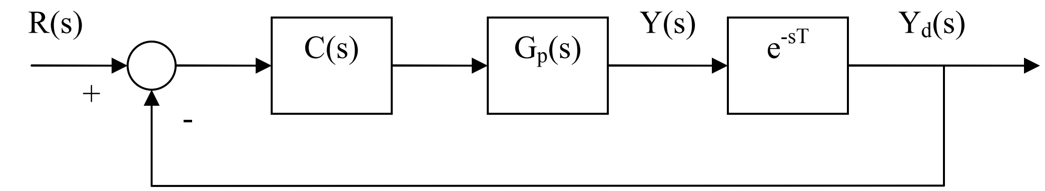


Figure 4.1: Feedback Control with Delay

predictor algorithm can be described as an attempt to cancel out the transmission delay in the control. The method by its own right is an effective algorithm to compensate for transmission delay given an accurate model of the slave is provided. Getting a Smith predictor to operate in wave variable domain simply makes it more robust to modeling errors.

To better understand the development of the Smith predictor, we will first show a block diagram of a simple feedback control where the transmission delay is represented by a separate block from the rest of the plant, as shown in Fig. 4.1. Here *C(s)* is the control block, $G_p(s)$ the plant block without the delay, and T the transmission delay. We can clearly see that the output delay $Y_d(s)$ is used for the negative feedback, and the pure delay term is clearly in the characteristic equation to potentially destabilize the closed-loop system. If we could use the fictitious un-delayed output *Y(s)* for the feedback, then we can remove the pure delay term from the control loop and significantly improve performance. Of course this is not physically feasible because it is usually not possible to isolate the pure delay from the rest of the plant. The delay may be distributed not

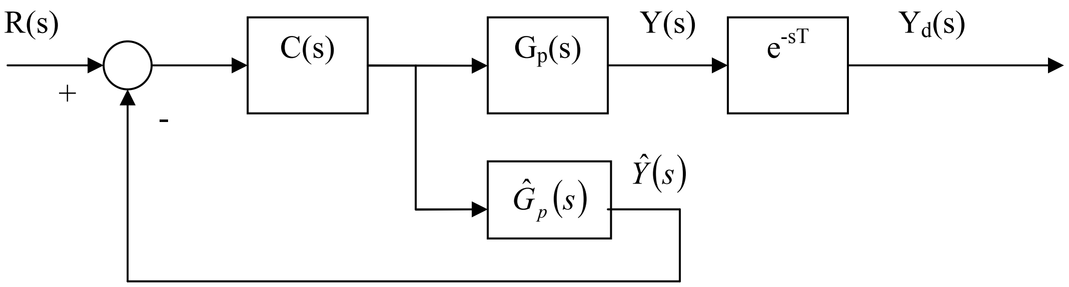


Figure 4.2: Preliminary Smith Predictor

lumped, and in case of teleoperation any information flow between the user and remote site will need to undergo pure transmission delay. Furthermore even if it is possible to isolate the pure delay term, there is no reason that it will be placed after the rest of the plant.

Although we can not use the un-delayed output of the actual plant, if we have a model of the plant we can use its un-delayed output instead. Consider a modified block diagram as shown in Fig. 4.2. The hat sign above the variables indicates estimated value. In this setup only the un-delayed modeled output is used to close the feedback loop. The controller would be very good at controlling the model, but not necessarily the actual plant because the actual plant dynamic is in an open-loop. To solve this problem an outer loop is added to the block diagram in Fig. 4.2 to create an algorithm represented by Fig. 4.3. Now the control block *C(s)* can be designed with higher gains more suitable to control an un-delayed plant since the effect of pure delay is minimized by the inner loop. This is the complete Smith predictor algorithm. In much of the literature, the Smith predictor diagram of Fig. 4.3 is rearranged such that the predictor can be lumped into a single control block as in Fig. 4.4. The block diagram in Fig. 4.4 is almost mathematically identical to that in Fig. 4.3 except for the addition of a disturbance at the plant input. But now we can see the Smith predictor control is lumped into a single inner loop as outlined by the dashed box.

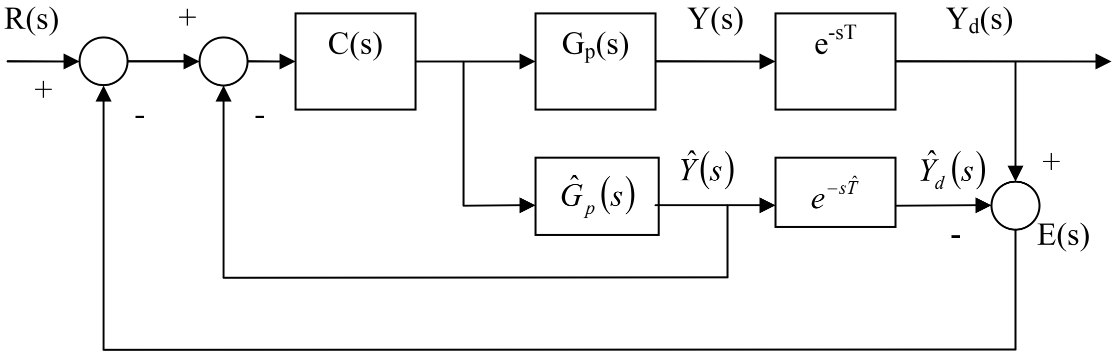


Figure 4.3: Complete Smith Predictor

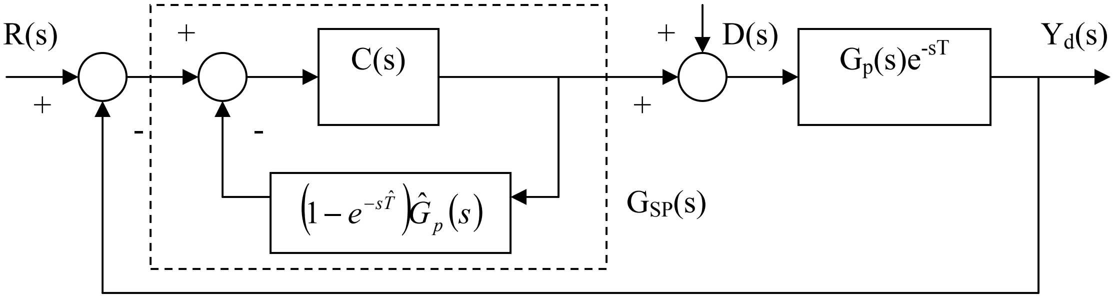


Figure 4.4: Complete Smith Predictor New Arrangement

We can see the effectiveness of the Smith predictor algorithm by observing the closed-loop transfer function with *R(s)* as input. Assume for now that there is no disturbance. The closed loop transfer function for Fig. 4.4 is

$$\frac{Y_d(s)}{R(s)}=\frac{C(s)G_p(s)e^{-sT}}{1+C(s)\hat{G}_p(s)-C(s)\hat{G}_p(s)e^{-s\hat{T}}+C(s)G_p(s)e^{-sT}} \tag{4.1}$$

If we assume perfect knowledge of the plant and pure delay for the Smith predictor, then $G_p(s)$ is equal to $\hat{G}_p(s)$, and T is equal to $\hat{T}$. Equation (4.1) will be reduced to

$$\frac{Y_d(s)}{R(s)}=\frac{C(s)G_p(s)e^{-sT}}{1+C(s)\hat{G}_p(s)} \tag{4.2}$$

Notice that the pure delay terms have been completely eliminated from the characteristic equation, although the term still remains in the closed-loop transfer function numerator. While this will not produce a stability problem, it does imply that tracking at the output will be delayed by the amount indicated by the system dead time. In fact with perfect

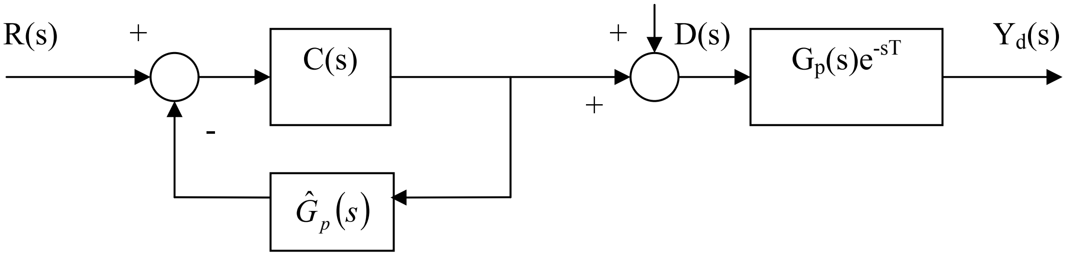


Figure 4.5: Smith Predictor Reduction by Assuming Perfect Modeling

modeling, the block diagram in Fig. 4.4 can be reduced to the Fig. 4.5.

The Smith predictor works just as well with disturbance force. Consider the transfer function of the closed-loop transfer function with *D(s)* as input, assuming perfect knowledge of the plant and pure delay, defined as

$$\frac{Y_d(s)}{D(s)} = \frac{C(s)G_p(s)\left[1+\hat{G}_p(s)C(s)\left(1-e^{-s\hat{T}}\right)\right]}{1+C(s)\hat{G}_p(s)} \tag{4.3}$$

Notice that the characteristic equation of this transfer function is the same as that of Eq. (4.2) without the pure delay term, indicating that pure delay will not cause instability although there will be tracking delay.

We can easily prove the stability of the Smith predictor in the inner loop by observing the closed-inner-loop transfer function shown as

$$G_{SP}(s) = \frac{C(s)}{1+C(s)\hat{G}_p(s)\left(1-e^{-s\hat{T}}\right)} \tag{4.4}$$

If the denominator of Eq. (4.4) becomes zero, the inner loop becomes unstable because *G_{SP}(s)* will become infinite. In this case we can derive the condition for inner loop instability to be

$$e^{-s\hat{T}} = \frac{C(s)\hat{G}_p(s)+1}{C(s)\hat{G}_p(s)} \tag{4.5}$$

This is clearly impossible unless you allow negative transmission delay. Hence the Smith predictor inner loop is always stable.

Thus far we have always assumed perfect knowledge of the plant dynamics. With inaccurate models however, the Smith predictor potentially becomes unstable. Consider a case where the plant model $G_p(s)$ is modeled incorrectly, but the transmission delay is modeled correctly. The estimated plant model is then

$$\hat{G}_p(s) = G_p(s) + \Delta G_p(s) \tag{4.6}$$

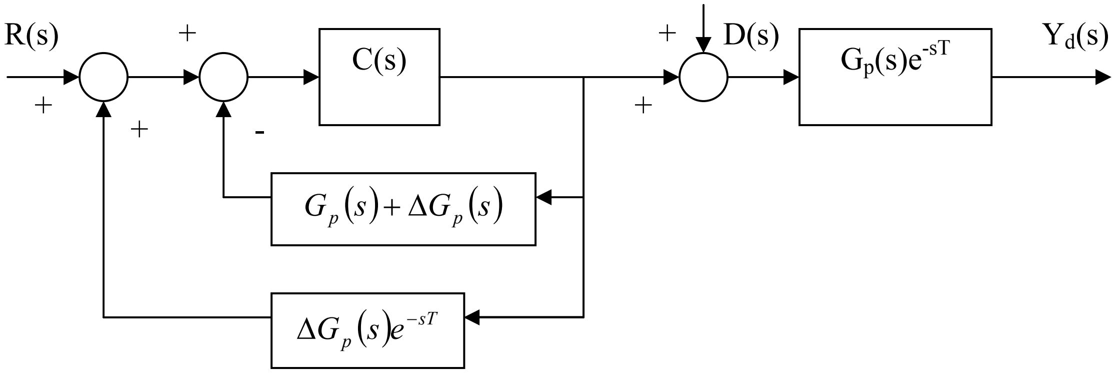


Figure 4.6: Smith Predictor with Inaccurate $G_p(s)$

In such case, the block diagram can be modified into Fig. 4.6. From Fig. 4.6 we can clearly see that with plant model error, the positive feedback in the outer loop can potentially destabilize the overall system. Now consider a case where the transmission delay *T* is modeled incorrectly, but the plant model *$G_p(s)$* is modeled correctly. The estimated transmission delay is then

$$\hat{T} = T + \Delta T \tag{4.7}$$

In such case, the block diagram can be modified into Fig. 4.7. As we can see, now the block diagram has an additional uncompensated pure delay term in the outer loop. Of course this brings back the original problem where the pure delay term becomes part of the feedback loop.

The predictor to be added to the wave variables is very similar in function the

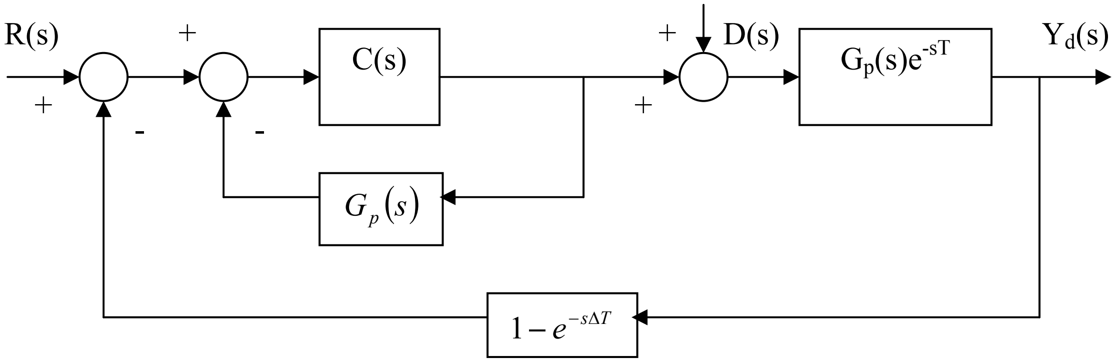


Figure 4.7: Smith Predictor with Inaccurate T

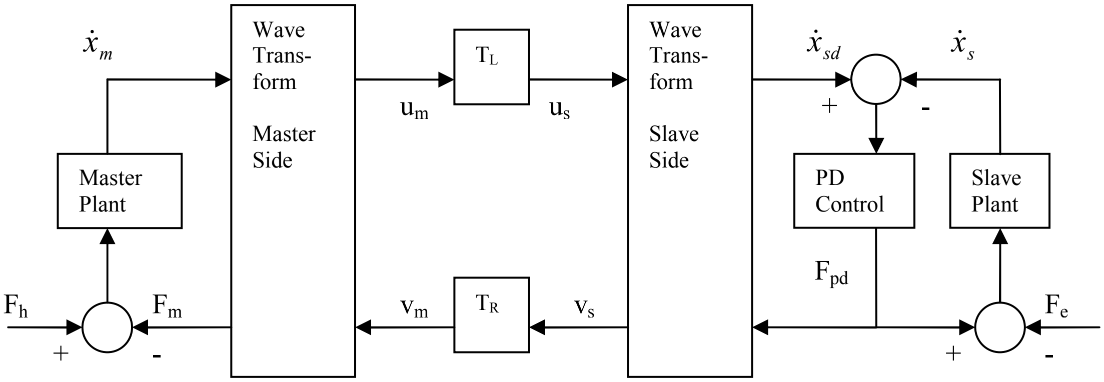


Figure 4.8: Wave Variable with PD Slave

Smith predictor explained above. However, to take advantage of wave variables' passivity guarantee, it has to operate in the wave variable domain. Though inaccurate modeling in the predictor would no longer cause instability, it will cause steady state error. The next sections will detail the steps taken to incorporate the predictor in wave variables to explain other modifications to compensate for the shortcomings.

4.2 Wave Predictor

Before moving on to the incorporation of a predictor into the wave variables, let us first define the dynamics of the master and slave. Consider Fig. 4.8. Notice that Fig. 4.8 is similar to Fig. 3.8 except that a master and slave are placed at the two ports. The master plant accepts force for the master wave transform block as feedback force and outputs velocity. In contrast the slave side has a local controller, a PD control in this case, that accepts velocity from the slave wave transform block as desired velocity and actual slave velocity from the slave plant. The error between these velocities is used by the PD control to generate a force output that along with the environmental force is accepted by the slave plant to produce velocity.

We can place the predictor on the master side between the wave transform blocks as in Fig. 4.9. As seen by the master, the outgoing wave variable u_m is current, but the incoming wave variable v_a is delayed by the total time delay of the transmission lines out

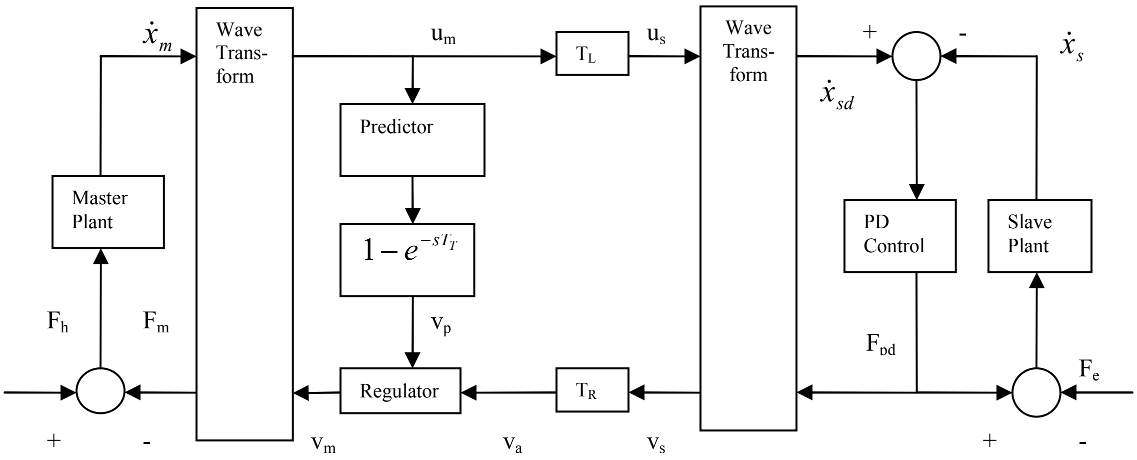


Figure 4.9: Wave Variable with Predictor

and back. The goal of the predictor is to estimate what the incoming wave variable is without the total time delay.

First the slave's states that includes the wave transform, PD control, and plant are sent directly back through transmission to the master side. These states are used as the initial conditions of the predictor that would anticipate the states after time T_T where T_T is the total delay transmission delay to and from the slave. Since the initial conditions of the predictor are delayed by T_T, the predictor is really anticipating the current states based on past states. The estimated current states of the slave are used to compute a new current incoming wave variable. This estimated incoming wave variable is compared with that value based on the initial condition of the predictor. Hence the equation for the predictor becomes

$$v_p = \left(1 - e^{-sT_T}\right) G_p(s) u_m \tag{4.8}$$

Here $G_p(s)$ is the slave transfer function. The difference v_p would be incorporated into the actual incoming wave variable v_a to create v_m. If a completely accurate model is used by the predictor to anticipate the current slave states, the master side would essentially see no transmission time delays.

One of the problems with this algorithm is how to incorporate the correction suggested by the predictor into the incoming wave variable. In theory, a simple summation junction should suffice. However, such a junction can not be guaranteed to be a passive element. An energy reservoir based regulator is used to insure passivity and incorporate the correction by driving v_m to the sum of v_a and v_p. Let the energy reservoir be defined as

$$E_\gamma(t) = \int_0^t \left[v_a^2(\tau) - v_m^2(\tau)\right] d\tau \tag{4.9}$$

Recall that the wave variable squared equals power. This energy reservoir merely keeps track of the net power going into the regulator.

The regulator also defines a distance-to-go term which is really the integral of the difference between the wave variable flow into and out of the regulator or

$$D_{tg}(t) = \int_0^t \left[v_a(\tau) + v_p(\tau) - v_m(\tau)\right] d\tau \tag{4.10}$$

The regulator output incoming wave variable v_m is then defined as

$$v_m(t) = \alpha\left[1 - e^{-\beta E_\gamma(t)}\right] D_{tg}(t) \tag{4.11}$$

Here α and β are positive constants. The square bracket term with the exponential of the energy reservoir insures passivity. If the energy reservoir E_γ is completely drained, the regulator would drive the output to zero and essentially shut down. Hence if β is large for a given amount of E_γ, then the regulator does not choke the output as severely. It is prudent however to keep an upper limit on the energy reservoir. Alternatively we can introduce a small leak in the energy reservoir with the leakage rate directly proportional to the amount of accumulated energy. This reason is with a large amount of energy accumulated in the reservoir it may take a long time for it to drain. This may violate passivity of the regulator on the short term and produce a temporary unstable system.

A large α causes v_m to converge faster to the sum of v_a and v_p. To show this, let us assume that the system has been running for a while, and the energy reservoir is sufficiently large such that the square bracket term is roughly constant at one. The regulator then becomes linear, and we can derive a transfer function with the sum of v_a and v_p as input and v_m as output. The approximate regulator equation becomes

$$\frac{V_m(s)}{V_a(s)+V_p(s)}=\frac{\alpha}{s+\alpha} \tag{4.12}$$

Notice now the regulator is just a first order filter a time constant of $1/\alpha$. We can easily see now that the smaller the time constant, the faster the convergence. Thus with this regulator, the algorithm can incorporate the predicted wave variable gradually while maintaining the passivity guarantee.

It is worth noting that using a predictor to fake the master into thinking that there is no delay can be very computationally expensive. The reason is because the predictor needs accept the slave's delayed states and output the anticipated current states as soon as possible. This means that it has to run a simulation spanning the duration of the total transmission delay T_T within a servo-loop if it were not to skip any servo-loop. For system running at 1 kHz with the return transmission delay of 800 ms, this implies the predictor simulation needs to run 800 simulation loops within 1 ms. Depending on the complexity of the model used and the computing power of the hardware, this type of performance may not be feasible. We can overcome this problem by allowing the predictor to skip a few servo-loops. This allows it more time to simulate, but outdated predictor output will need to be used for the skipped servo-loops. Alternatively we can increase the predictor simulation step size and effectively simplify the simulation. This can increase the speed of simulation allowing completion within the servo-loop. Either method will reduce predictor accuracy and negatively affect overall teleoperation performance.

4.3 Position Drift Analysis

A good model for the predictor is crucial for the system to work properly. Despite the guaranteed passivity even for an inaccurate predictor by the energy based regulator, the error can still accumulate over time resulting in position drift. For the rest of the thesis, we define position drift between the master and slave desired positions as

$$\varepsilon(t) = x_m(t-T) - x_{sd}(t) = \int_0^t [\dot{x}_m(\tau - T) - \dot{x}_{sd}(\tau)]d\tau \tag{4.13}$$

Notice the difference between desired slave position with subscript *sd* and actual slave position with subscript *s*. The tracking of actual slave position is left to the slave PD control. For simplicity sake, assume that the transmission delay is a constant *T* for both directions. The master position is delayed by *T* to match the time frame of the slave.

In case of the wave variable algorithm represented by Fig. 4.7, the master and slave desired velocities can be written as functions of wave variables, defined as

$$\dot{x}_m(t) = \frac{1}{\sqrt{2b}}[u_m(t) + v_m(t)] \tag{4.14}$$

$$\dot{x}_{sd}(t) = \frac{1}{\sqrt{2b}}[u_s(t) + v_s(t)] \tag{4.15}$$

The corresponding wave variables on the master and slave only differ by the delay or

$$u_s(t) = u_m(t-T) \tag{4.16}$$

$$v_m(t) = v_s(t-T) \tag{4.17}$$

Substituting Eq. (4.14) to (4.17) into Eq. (4.13), we get

$$\begin{aligned}\varepsilon(t) &= \int_0^t [\dot{x}_m(\tau - T) - \dot{x}_{sd}(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [u_m(\tau - T) + v_m(\tau - T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [u_s(\tau) + v_s(\tau - 2T) - u_s(\tau) - v_s(\tau)]d\tau = -\frac{1}{\sqrt{2b}}\int_{t-2T}^t v_s(\tau)d\tau\end{aligned} \tag{4.18}$$

Notice when the system is static in free space with no velocity and force inputs, the wave variables die out to zero resulting in zero steady state position error. However, under contact where velocity is zero but force is non-zero, there will be steady state position error. This is because the wave variables will be non-zero but constant.

Now let us assume that the wave variables are transmitted imperfectly. Equations (4.16) and (4.17) becomes

$$u_s(t) = u_m(t-T) + \delta_1(t) \tag{4.19}$$

$$v_m(t) = v_s(t-T) + \delta_2(t) \tag{4.20}$$

The δ's signify errors during transmission which can be the result of variable transmission delays or lost data packages. If we now use Eq. (4.19) and (4.20) to compute the position error, we get

$$\begin{aligned} \varepsilon(t) &= \int_0^t [\dot{x}_m(\tau - T) - \dot{x}_{sd}(\tau)] d\tau \\ &= \frac{1}{\sqrt{2b}} \int_0^t [u_m(\tau - T) + v_m(\tau - T) - u_s(\tau) - v_s(\tau)] d\tau \\ &= \frac{1}{\sqrt{2b}} \int_0^t [u_s(\tau) - \delta_1(\tau) + v_s(\tau - 2T) + \delta_2(\tau) - u_s(\tau) - v_s(\tau)] d\tau \\ &= \frac{1}{\sqrt{2b}} \int_0^t [\delta_2(\tau) - \delta_1(\tau)] d\tau - \frac{1}{\sqrt{2b}} \int_{t-2T}^t v_s(\tau) d\tau \end{aligned} \tag{4.21}$$

Equation (4.21) clearly shows how the transmission errors can integrate and accumulate over time in wave variable only algorithm to create position drift at steady state.

The wave predictor setup of Fig. 4.8 can also generate position drift given an inaccurate predictor model. But first let us assume perfect modeling in which case the predictor on the master side should propagate the incoming delayed wave variable to the present perfectly. We also assume that the energy reservoir in the regulator to be sufficiently large such that it is far from choking the wave variable flow. Furthermore we assume that the system is sufficiently slow such that the first-order filtering effect of the regulator is negligible. In this case, Eq. (4.14), (4.15), and (4.16) still hold, but Eq. (4.17) needs to be substituted by

$$v_a(t) = v_s(t - T) \tag{4.22}$$

$$v_m(t) = v_a(t + 2T) \tag{4.23}$$

In Eq. (4.23), the compensated incoming wave variable v_m used to compute master velocity is exactly the same as the raw incoming wave variable v_a straight from the transmission line except it has been marched forward in time due to the perfect prediction. The position drift can then be computed as

$$\begin{aligned}\varepsilon(t) &= \frac{1}{\sqrt{2b}}\int_0^t [u_m(\tau - T) + v_m(\tau - T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [u_s(\tau) + v_a(\tau + T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [v_s(\tau) - v_s(\tau)]d\tau = 0\end{aligned} \tag{4.24}$$

This shows that given perfect wave prediction, not only will there be no steady state error, but the desired slave position should track the master position perfectly.

Now let us assume there is an error in the predictor model such that the predictor no longer perfectly marches the delayed incoming wave variable to the present. In this case, Eq. (4.23) will be changed to

$$v_m(t) = v_a(t + 2T) + \delta(t) \tag{4.25}$$

Here δ is an error in the predicted wave variable. Now the position drift becomes

$$\begin{aligned}\varepsilon(t) &= \frac{1}{\sqrt{2b}}\int_0^t [u_m(\tau - T) + v_m(\tau - T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [u_s(\tau) + v_a(\tau + T) + \delta(\tau) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [v_s(\tau) + \delta(\tau) - v_s(\tau)]d\tau = \frac{1}{\sqrt{2b}}\int_0^t \delta(\tau)d\tau\end{aligned} \tag{4.26}$$

This shows that an error in the predictor could produce position drift at steady state. Again if we include other sources of error such as time varying transmission delay or lost data packages, we will also get steady state error in position.

Surprisingly if we made an error in estimating the transmission delay for the predictor model, the resulting system will not produce a steady state error in position. We can prove this by going through the same analysis. If there is an error in the estimated transmission delay in an otherwise perfect predictor model, Eq. (4.23) becomes

$$v_m(t) = v_a(t + 2T + \Delta T) \tag{4.27}$$

Here *ΔT* represents the error in the estimated transmission delay. Using this to compute the position drift, we get

$$\begin{aligned}\varepsilon(t) &= \frac{1}{\sqrt{2b}}\int_0^t [u_m(\tau - T) + v_m(\tau - T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [u_s(\tau) + v_a(\tau + T + \Delta T) - u_s(\tau) - v_s(\tau)]d\tau \\ &= \frac{1}{\sqrt{2b}}\int_0^t [v_s(\tau + \Delta T) - v_s(\tau)]d\tau = \frac{1}{\sqrt{2b}}\int_t^{t+\Delta T} v_s(\tau)d\tau\end{aligned} \tag{4.28}$$

Though position tracking is not perfect now, there is still zero steady state error if the wave variables die out when the system is static and in free space, not unlike that of the wave variable only algorithm. We must emphasize that this is still assuming constant transmission delay. In case of varying transmission delay or lost data packages, some of the data are skipped and not utilized. Recall from the previous chapter that wave variable only encodes velocity and force. If wave transmission is imperfect, then reconstruction of position will produce drift. The result is more akin to poor estimation of model plant than transmission delay.

4.4 Wave Variables with Predictor and Drift Control

As shown above, the wave predictor algorithm alone would work well if transmission delay is constant and slave model used in the model is accurate. If either of these is not true, then the system will exhibit position drift. The wave variable equations show a state transformation of velocity and force data. Under variations in transmission delay the controller will not preserve the integral of the wave variable which contains the position information. Under realistic conditions, obtaining an accurate model of a physical system may be difficult or even impossible. An inaccurate model will cause the predictor to produce an incorrect correction to the incoming wave variable. This error will accumulate into steady state error.

Most drift correction control methods derived in the past are specially designed to deal with drifts caused by variations in transmission delays. A good example is the drift control algorithm proposed by Munir and Book [18]. Their drift control algorithm attempts to drive steady state error between master and slave desired positions to zero via a correction input to the outgoing wave variable u_m. Because the drift control algorithm is located on the master side, it can not directly compare the master and desired slave positions. The desired slave position would have been outdated by the time it reaches the master, hence the master and desired slave positions would not be in the same "time frame." Instead, the drift control algorithm has to compare the difference between the expected and actual master-slave position difference. This implies a risk where the actual master-slave position difference may not settle to zero at steady state if the expected value does not do so due to predictor modeling error. The following is an introduction of a more robust drift control algorithm that can cope with drift caused by predictor modeling errors by directly driving the actual master-slave position difference to zero.

A block diagram of the modified system is shown in Fig. 4.10. Notice that the actual master and desired slave positions are compared on the slave side, signified by the block labeled *drift*. The goal of the drift control algorithm is to directly drive the difference between these two positions to zero. This is a reasonable goal because the two positions are in the same "time frame." The slave desired position should track the master position with T_L delay, while the master position data upon arrival on the slave

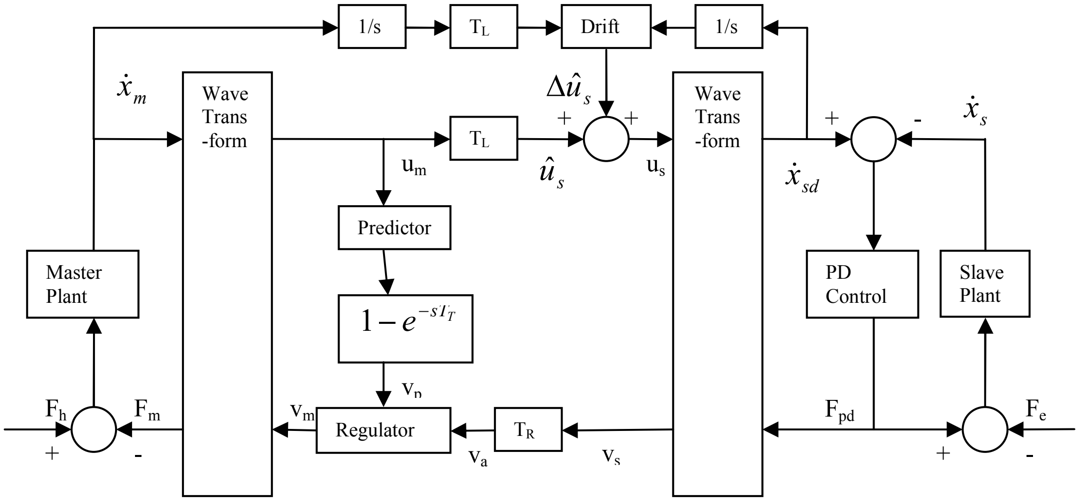


Figure 4.10: Bilateral Teleoperation in Wave Variables with Predictor and Direct Drift Control

side is also delayed by T_L. This information is used to produce a correction signal $\Delta\hat{u}_s$ that is injected to the wave variable $\hat{u}_s$ received by the slave on the slave side. The algorithm would also need to accomplish this passively.

The proposed drift control algorithm is similar to the predictor regulator algorithm. Consider an energy reservoir that keeps track of how much net energy is dissipated by the slave system that includes the PD control, plant, wave transform, and the correction term from the drift control itself. The energy reservoir is defined as

$$E_d(t) = \int_0^t \left[\hat{u}_s^2(\tau) - v_s^2(\tau)\right] d\tau \tag{4.29}$$

Recall that the square of the wave variable equals to the power flow. The correction term is computed as

$$\Delta\hat{u}_s(t) = \frac{\gamma\sqrt{2b}}{2}\left\{x_m\left[t - T_L(t)\right] - x_{sd}(t)\right\}\left[1 - e^{-\delta E_d(t)}\right] \tag{4.30}$$

Here γ and δ are positive constants. Similar to the regulator, the bracket term can ensure passivity by choking off the correction wave variable when the total energy dissipated by the slave system reaches zero. This can happen if the slave becomes active, i.e. when there is more energy coming out of the slave than going into it. The choice of the values for γ and δ can be determined in a similar manner with how the values of α and β can be determined in the predictor regulator algorithm in Eq. (4.11). Namely γ determines how fast the desired slave position converges to the master position, and δ determines how quickly the energy reservoir fills.

We can demonstrate how Eq. (4.30) is able to drive the desired slave to master position by showing that when the master and desired slave positions do not agree, there will also be restoring force. Consider an approximately steady state where the master position x_m and the PD control force F_{pd} is roughly constant. In free space, PD control force F_{pd} would be zero when the master is static. In contact, PD control force F_{pd} would be a constant nonzero value when the master is static. Under either circumstance, the transmission delay does not matter because the delay is applied to an essentially constant value. Also assume that the system has been running for long enough such that energy reservoir is sufficiently large enough to make $1 - e^{-\delta E_d(t)} \approx 1$. Making the right substitutions for u_s, wave variable equation for the incoming wave variable relative the master becomes

$$v_s = \hat{u}_s + \Delta\hat{u}_s - \frac{2}{\sqrt{2b}} F_{pd} \tag{4.31}$$

Keeping in mind that the system is at approximately steady state, the incoming wave v_s is roughly equal to v_a. Next the incoming wave variable relative to the master side would have to go through the predictor input. The predictor may not have an accurate model, but one can assume that it at least produces the same steady state response as the actual slave. This is a reasonable assumption because a constant master position should yield a constant force feedback from the slave, zero if in free space. Unless in cases of severe modeling error (such as using a free space instead of contact model), the predictor should not produce any correction during steady state. Hence the incoming wave variable v_m after predictor input should be roughly the same as v_s. Given the master velocity is zero, outgoing wave variable u_m is just the reflected and inverted incoming wave variable v_m. Finally at steady state, the transmission delay should not affect the outgoing wave variable, and $\hat{u}_s$ is roughly equal to u_m. To sum up all the equalities, one can conclude that $\hat{u}_s$ is approximately equal to $-v_s$ or

$$v_s \approx v_a \approx v_m \approx -u_m \approx -\hat{u}_s \tag{4.32}$$

It is then possible to rewrite Eq. (4.31) to get

$$\hat{u}_s = -\hat{u}_s - \Delta\hat{u}_s + \frac{2}{\sqrt{2b}} F_{pd} \tag{4.33}$$

$$\hat{u}_s = -\frac{\Delta\hat{u}_s}{2} + \frac{1}{\sqrt{2b}} F_{pd} \tag{4.34}$$

Then one can substitute Eq. (4.34) into the wave transform equation to compute the slave desired velocity and get

$$\dot{x}_{sd} = \frac{\sqrt{2b}}{b}\left(\hat{u}_s + \Delta\hat{u}_s\right) - \frac{1}{b} F_{pd} = \frac{1}{\sqrt{2b}} \Delta\hat{u}_s \tag{4.35}$$

$$\Delta\hat{u}_s = \sqrt{2b}\dot{x}_{sd} \tag{4.36}$$

The drift control law can now be written by substituting Eq. (4.36) into Eq. (4.30) to get

$$\dot{x}_{sd} \approx \frac{\gamma}{2}\left(x_m - x_{sd}\right) \tag{4.37}$$

At this stage one can easily see how Eq. (4.30) is able to drive x_{sd} asymptotically to x_m. Also notice that the algorithm is reduced to an approximately first-order filtering effect where the time constant is $2/\gamma$. This drift control scheme has the benefit of being indiscriminate regarding the source of drift. It is dependent only on the measured master and desired slave position difference. As long as there is disagreement between the master and desired slave positions, it would try to compensate for it.

Again it is prudent to set an upper limit to the energy reservoir or introduce a leak, just like the case for the predictor regulator. However we have to be more careful when setting this upper energy reservoir limit. When the slave moves from free space to contact environment, there will be a sudden surge in feedback force which in turn will produce an abrupt increase in the wave variable sent back to the master. The drift control algorithm perceives this as the slave becoming active, resulting in a rapid drain of the energy reservoir. This drain is only temporary since the wave variable eventually gets reflected off the master back to the slave and hence refilling the drift control energy reservoir. However it takes a while for this to happen depending on the duration of the transmission delay. If the upper limit of the energy reservoir is too low, the reservoir may completely drain before the wave variable reflected off the master reaches the slave, effectively shutting off the drift control prematurely. The result is a poor transition from free space to contact where the drift control energy reservoir is repeatedly emptied and refilled for several seconds. The tracking between the master and desired slave positions becomes relatively poor during this transition period. Therefore while a low upper limit on the drift control energy reservoir is important to prevent the system from becoming temporarily unstable; the limit must be large enough to prevent premature and unintentional emptying during transition from free space to contact.

4.5 Adaptive Predictor in Wave Variables

Up to this point it is assumed that the predictor model is determined beforehand. The predictor on the master side would know exactly where the contacts are, the damping in different workspace locations, when the slave becomes loaded, etc. In other words, the predictor would need to contain all the relevant information on the slave environment and workspace. Naturally this level of comprehensive knowledge of the slave environment is very difficult to achieve beforehand. Furthermore the slave environment may change in the middle of operation. The predictor model must adapt to the slave environment dynamics to remain accurate. As noted in the previous sections, inaccuracies in predictor model can produce wrong predictions on the returning wave relative to the master that negatively affects the feedback force on the master side and the desired position and velocity on the slave side.

This section introduces two possible methods to adjust the predictor model based on the changes in the slave environment. The first is the semi-adaptive predictor where the predictor model is categorized into discrete modes. The modes are then switched based on the relationship between the slave input and output. The second is the full-adaptive predictor where the parameters that define the predictor model are adjusted continuously over time using a recursive least-square algorithm.

4.5.1 Semi-Adaptive Predictor

Figure 4.11 shows the block diagram of a generalized semi-adaptive predictor algorithm in wave variables. The predictor has a library of predetermined models available. This library of models should ideally reflect all the possible states that the

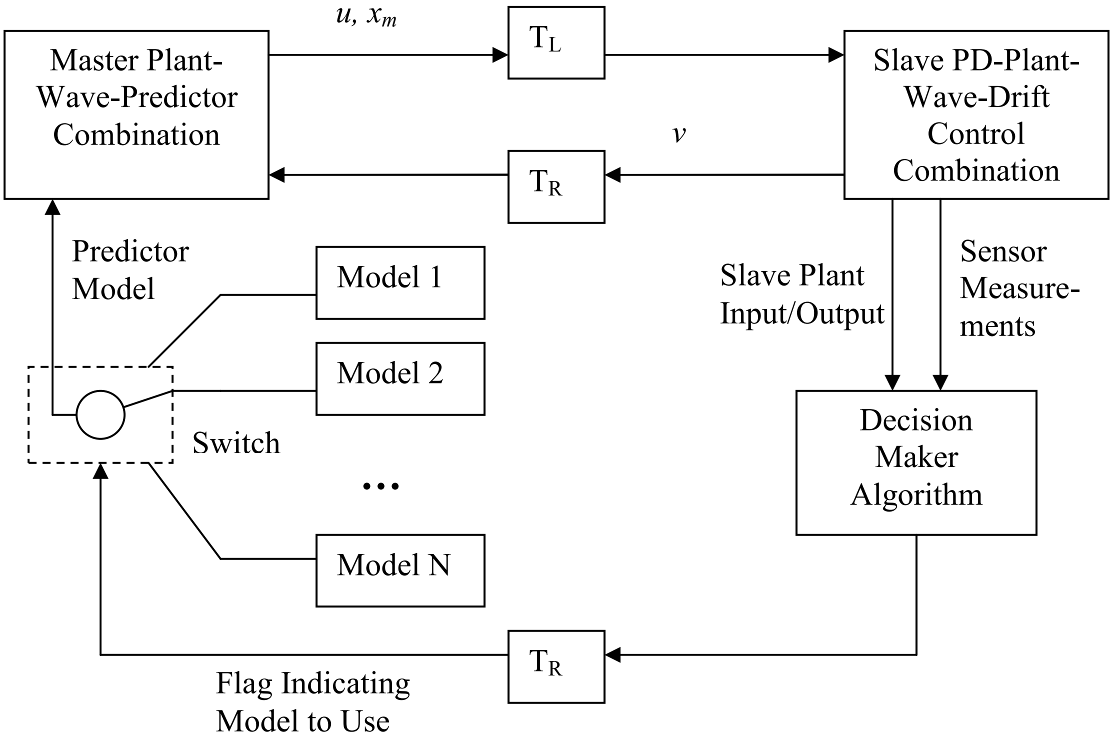


Figure 4.11: Semi-Adaptive Predictor

slave environment may encounter. A decision maker algorithm would be used to decide which model the predictor should use. This choice could be made directly based on sensor measurements such as load cell, contact detector, etc. or indirectly based on the relationship between slave inputs and outputs. The decision maker algorithm can be as simple as a sign check or something more elaborate such as back-propagation neural networks. Figure 4.11 shows the decision maker algorithm accepting slave plant input/output on the slave side while transmitting a flag back to the master telling the predictor which model to use. Placing the decision maker algorithm on the slave side is optional. We can always place the algorithm on the master side and have the slave send the slave plant input/output and sensor measurement data back to the master. However given the large computation requirement of the predictor on the master computer, it is desirable to divide up some of the decision maker computation with the slave computer.

Even when all the possible models available to the predictor may be stable, abrupt switching between stable models could still drive the system unstable under certain circumstances. Nevertheless the predictor correction input still has to pass through the predictor regulator that insures passivity by keeping track of the energy flow in the feedback wave variable.

This section will use a specific example of the semi-adaptive predictor for testing using the PHANTOM haptic devices where the possible number of slave environments is divided into two modes: free space and rigid contact. Of course there are many other possible modes, but we feel that these two modes are distinct enough, free space and rigid contact being extremely high admittance and impedance environments respectively, to sufficiently demonstrate the capability of the proposed algorithm. In addition, free space and rigid contacts are among the most frequently encountered environments. In free space, the slave is modeled as a mass-damper system. In rigid contact, the actual slave position is fixed at the point of contact.

As there is no contact sensor mounted on the PHANTOMs, the decision maker algorithm detects contact by the error between desired and actual position on the slave side. When in free space, the slave PD control can keep this error significantly smaller than under contact. So a threshold value of the error is set such that if the actual error is smaller than this value then the slave is assumed to be in free space, otherwise it is assumed to be in contact. For this algorithm to work more effectively, a low pass filter is used on the error data. This minimizes the possibility of false contact detection due to short and abrupt surges in the position error which may happen in free space due to noise or quick movements. Depending on the type of environment encountered by the slave, the semi-adaptive predictor model switches accordingly and maintain predictor accuracy.

As the decision maker algorithm makes a free space or rigid contact detection on the slave side, there is a delay in predictor model switching as the signal takes time T_R to travel from slave to master. When the slave goes from free space to contact, this will produce a delay before the user feels the force of contact on the master side. This is unavoidable if there is no a priori knowledge of the slave environment and the contact locations. However as the master commands the slave to move from contact back into free space, the predictor is able to anticipate the point where the slave leaves contact via the motion of the master and the newly acquired knowledge of the contact location. This allows the predictor to switch back into free space mode before it receives the flag signal from the decision maker algorithm on the slave side. Had the predictor waited for the flag signal and remained in rigid contact mode while the master position moves back into free space, it would predict a temporary feedback force pulling the master from free space back toward the contact point.

4.5.2 Full-Adaptive Predictor

The adaptability of the predictor is not limited to free space and rigid contact or any finite number of modes. We can continuously modify the predictor model depending

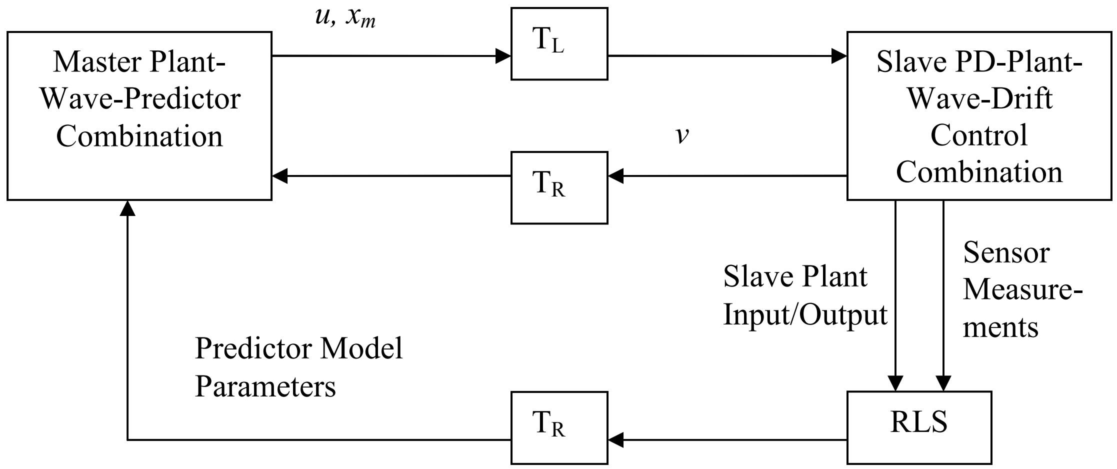

u, x_m
Master Plant-Wave-Predictor Combination
T_L
Slave PD-Plant-Wave-Drift Control Combination
v
T_R
Slave Plant Input/Output
Sensor Measure-ments
Predictor Model Parameters
T_R
RLS

Figure 4.12: Full-Adaptive Predictor Using RLS

on the changing slave environment by a self-tuning algorithm or direct measurement of the slave environment. Here a full-adaptive predictor algorithm using recursive least-square (RLS) method is used to accomplish this. Figure 4.12 shows the block diagram of a full-adaptive predictor in wave variables running on RLS. The RLS filter is located on the slave side. It reads the slave plant input/output and sensor measurements to estimate the parameters of the slave plant. The estimated parameters are sent back to the master and update the predictor model. Again the RLS filter can be place on the master side as well, but it is placed on the slave side to balance the computation load. Passivity can also be assured using the predictor regulator as in the case with semi-adaptive predictor.

The algorithm for RLS filter can be easily found in many textbooks. This section provides a quick summary for the convenience of the reader. Assume that the filter is trying to estimate p parameters of an unknown linear plant. The input into the filter is

$$\vec{y}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \\ \cdots \\ y_p(n) \end{bmatrix} \tag{4.38}$$

Here *n* is the iteration number. The list of parameters of the unknown plant to be estimated can also be stacked into a vector like

$$\vec{w}(n) = \begin{bmatrix} w_1(n) \\ w_2(n) \\ \cdots \\ w_p(n) \end{bmatrix} \tag{4.39}$$

If the actual output of the unknown plant is *d(n)*, and the estimated output is the linear combination of the $\vec{y}(n)$, then we can generate an error to minimize as

$$e(n) = d(n) - \vec{w}^T(n)\vec{y}(n) \tag{4.40}$$

The estimated parameter update equation can now be described as

$$\vec{w}(n) = \vec{w}(n-1) + e(n)\vec{g}(n) \tag{4.41}$$

$$\vec{g}(n) = \frac{P(n-1)\vec{y}(n)}{\lambda + \vec{y}^T(n)P(n-1)\vec{y}(n)} \tag{4.42}$$

$$P(n) = \left[I - \vec{g}(n)\vec{y}^T(n)\right]\lambda^{-1}P(n-1) \tag{4.43}$$

Here λ is the forgetting factor where the case of $\lambda = 1$ corresponds to infinite memory. Implementation requires proper initialization of the *P* matrix. A simple and commonly used initialization is to set *P(0)* to be a very small number multiplied by an identity matrix.

The full-adaptive algorithm is tested for its ability to distinguish between free space and rigid contact for comparison with the semi-adaptive case. The slave mass and damping constant are assumed to be known as rigid contact does not normally affect these parameters. Of course the forgetting factor would need to be kept slightly less than one to cope with changing slave environment. In the simplest case, it is possible to reflect rigid contact in the predictor model purely by environmental force acting on the slave that the RLS needs to estimate based on PD control force and slave position histories. Since the predictor only reacts to environmental force, there would be a delay in force feedback as the slave moves into and out of contact. As it does not know the contact location, it can not anticipate when the slave would move from contact back into free space and compensate accordingly.

Alternatively it is possible to reflect contact in the predictor model by the slave spring constant against a contact point. In this case, the RLS needs to estimate the spring constant and the contact point based on PD control force and slave position histories. Unlike the previous case, this predictor does estimate the contact location and can anticipate when the slave would move back into free space based on master position. It can reset the slave spring constant to zero before the RLS algorithm on the slave side updates the predictor model's parameters. If the predictor does not do that, the predictor model's slave spring constant would temporarily remain "hooked" to the contact point even when the master moves back into free space. The master would feel a temporary force pulling it back towards the contact point from free space. In many ways this is very similar to the semi-adaptive predictor in performance. However, in this case the predictor can theoretically cope with soft contacts because instead being limited into two modes, the predicted force is based on slave environment spring constant which ranges anywhere from zero (free space) to infinite (rigid contact).

In general, the full-adaptive predictor is more robust than the semi-adaptive predictor due to the continuously variable predictor model. The RLS algorithm used in full-adaptive predictor is also more computationally expensive and takes longer to converge. Hence the semi-adaptive predictor with its mode-switching algorithm may be more suitable for abrupt changes in the environment as from free space to rigid contacts, where the full-adaptive predictor may be more suitable for environments with vary slowly.

CHAPTER 5

EXPERIMENTAL SETUP AND RESULTS

To test the proposed algorithms based on wave variables, actual physical devices are used for experimental verification. The conventional, wave variable only, and wave-adaptive-predictor-drift-control algorithms are tested under free space and rigid contact environments. The primary hardware used is the PHANTOM haptic devices for both master and slave. The PHANTOMs used for the master and slave are comparable in size and dynamics, although the slave device is slightly larger than the master device. The algorithms are also tested to a lesser degree using the PHANTOM as master and the Hydraulic Actuator Lift (HAL) as slave to demonstrate the effectiveness in a highly asymmetric teleoperation setup. The only part of the experiment that is simulated is the transmission delay. To better relate to real transmission delay over typical UDP network, the simulated transmission is time-varying and imperfect with lost data.

5.1 PHANTOM Haptic Hardware

The primary machines used in the experiments are the PHANTOM haptic devices manufactured by Sensable Technologies. The name PHANTOM is derived from Personal Haptic Interface Mechanism. It is invented by Massie and Salibury [68] at the MIT Artificial Intelligence Laboratory and currently manufactured by Sensable Technologies. The PHANTOMs were originally designed to allow human users to touch and manipulate virtual objects, making them very useful in CAD-related applications. They can be seen as joystick with three servomotors to provide force feedback in three axes. Although not originally designed for this purpose, the servomotors can also move the entire PHANTOM, causing it to work like a robotic manipulator. The PHANTOMs are equipped with encoders to detect position in all three axes. However, they are not

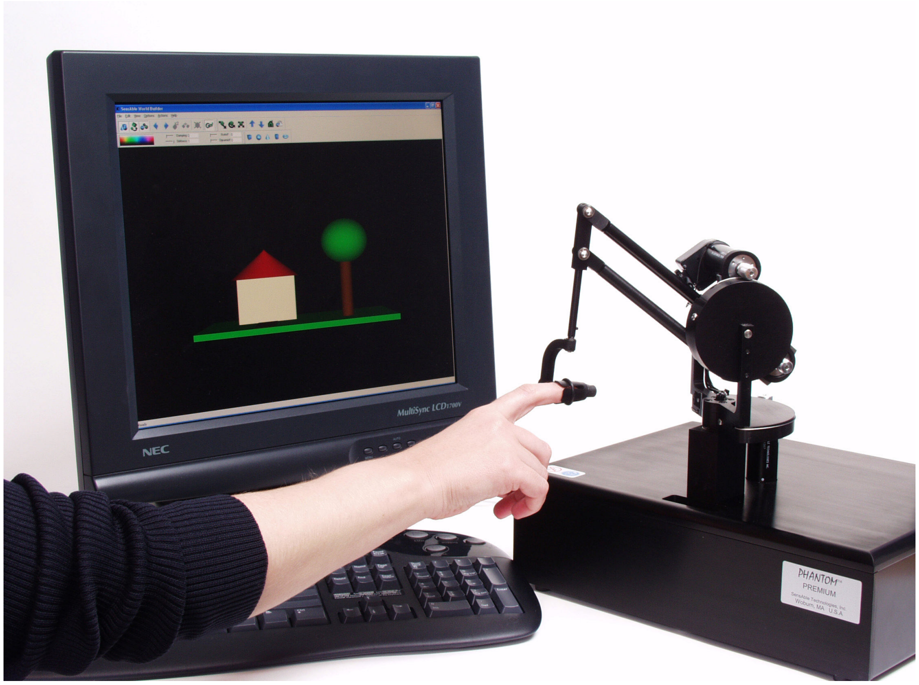


Figure 5.1: PHANTOM Premium 1.0

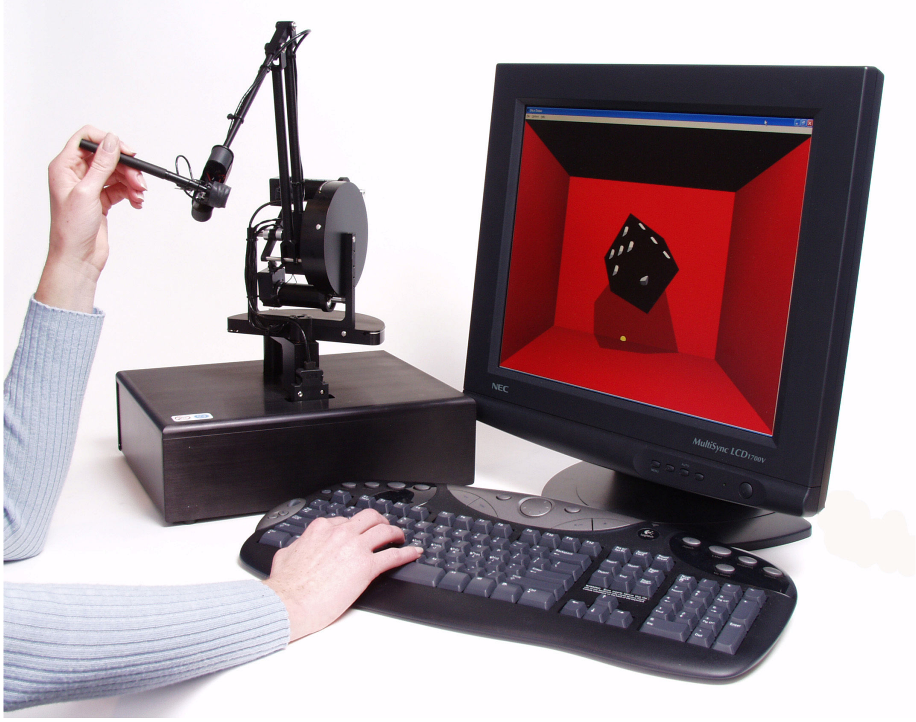
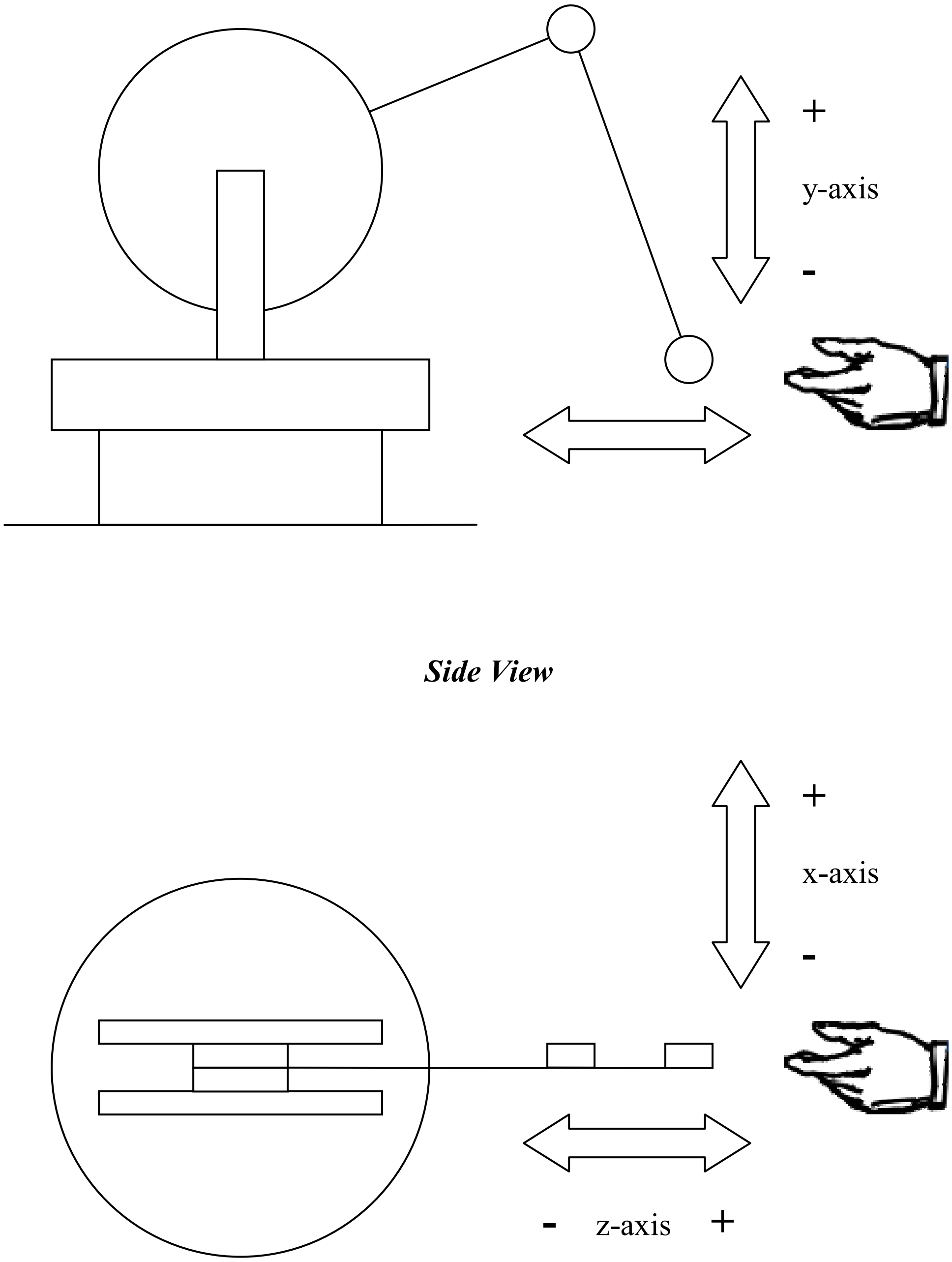


Figure 5.2: PHANTOM Premium 1.5



Side View

Top View

Figure 5.3:   PHANTOM Axis Directions

equipped with a tachometer; hence velocity data must be computed numerically from position data. If necessary, encoder gimbals at the tip are available to measure orientation although without any torque feedback. Through the course of this study, these encoder gimbals are not used.

The master device uses PHANTOM Premium 1.0 while the slave device uses the PHANTOM Premium 1.5. These devices are essentially identical except the workspace of the later is roughly 1.5 times that of the former. Pictures of the PHANTOM Premium 1.0 and 1.5 are shown in Fig. 5.1 and 5.2 respectively. The PHANTOM uses Cartesian coordinates in three directions *x*, *y*, and *z*. The axis and direction notations are better presented in Fig. 5.3. Although the master and slave PHANTOMs are different models, the servomotors used are of the same model. This means the force output characteristics of the PHANTOMs are similar despite the differences in workspace size. The PHANTOM Premium hardware specifications are summarized in Table 5.1. The specifications are recorded from an online brochure at the Sensable website [3].

Table 5.1: PHANTOM Hardware Specifications

Workspace Size	Premium 1.0: 254 x 178 x 127 mm (x, y, z axes). Premium 1.5: 381 x 267 x 191 mm (x, y, z axes).
Nominal Position Resolution	0.03 mm
Backdrive Friction	0.04 N
Maximum Exertable Force	8.5 N
Continuous Exertable Force	1.4 N
Apparent Inertia at Tip	~ 75 gm
Stiffness	3.5 N/mm

To interface with the computer, the PHANTOMs come provided with power electronic box and a PCI digital-to-analog card. The power electronic box is essentially an amplifier box for the servomotors with related routing circuitry connected to the PCI digital-to-analog card via parallel port. For the purpose of providing a rough model for the predictor, the PHANTOM 1.5 model used for the slave is estimated to be as

$$\frac{X_s(s)}{F_{pd}(s)} = \frac{1}{s(ms+b)} \tag{5.1}$$

Here X_s, F_{pd}, m, and b are the slave position, PD control force, PHANTOM mass, and PHANTOM damping constant respectively. Through experimentation, the PHANTOM 1.5 is found to have an approximate mass of 0.15 kg and approximate damping constant of 1 N/(m/s). These values are approximate because all hardware specifications beyond what is given in Table 5.1 are proprietary. Notice the effective mass is actually higher than listed because the PHANTOM has extra weights attached to the end-effector.

5.2 Ghost SDK

The PHANTOM comes with the Ghost Software Development Kit that permits easy programming, essentially taking care of the lower level control and allowing the programmer to focus on the higher level control. It converts the incoming encoder signals into position and, if applicable, orientation data of the end-effector. In addition, it takes care of the servo-control for output force tracking. The Ghost SDK calls on a servo-loop code every millisecond. The user only needs to program this servo-loop in C++. Because C++ is a very common programming language, an average user can easily learn to program the PHANTOMs. From the servo-loop he or she can request the Ghost SDK for the positions, velocities, and accelerations of the PHANTOM. As noted above, the PHANTOM has only encoders. Hence Ghost SDK needs to numerically differentiate the position data to get velocities and accelerations. At the end of each millisecond, the servo-loop is required to output a desired force vector computed by the programmer to be sent to the PHANTOM via the Ghost SDK.

The Ghost SDK can run on any Intel-based PC under several existing Windows Operation System. A Linux version of Ghost SDK also allows it to run under Red Hat

Linux. The minimum computer system requirement to run Ghost is given in Table 5.2. The information also comes from the online brochure at Sensable website [3].

Table 5.2: Ghost SDK Minimum System Requirement

Operation Systems	Intel-based PC on Windows XP, Windows 2000, Windows NT 4.0 SP6, or Red Hat Linux
Processor Speed	300 MHz Pentium processor
Memory	64 MB RAM
Hard Disk Space	30 MB free disk space

The specification for the computers used to control the master and slave PHANTOMs are shown in Table 5.3.

Table 5.3: Master and Slave Computer Specifications

	Master PC	Slave PC
Operation System	Windows 2000	Windows 2000
Processor Speed	1.7 GHz Pentium 4 Processor	1.5 GHz Pentium 4 Processor
Memory	256 MB RAM	256 MB RAM
Hard Disk Space	40 GB	6.4 GB

In the interest of safety for the PHANTOMs and the human users, the Ghost SDK also has several safeguard features. Because the PHANTOM encoder output is relative instead of absolute, the PHANTOM must be initialized as part of the startup procedure by setting all the joint angles to right angles. Immediately after initialization, the Ghost builds up the force slowly such that the human user would not be caught by surprise. This is very important under certain circumstances where the initial position would produce large feedback force that could harm both the user and the PHANTOMs. In addition, the Ghost SDK terminates the servo-loop whenever it detects either a large force or velocity spike that can be potentially damaging. Finally the PHANTOM servomotor temperature is carefully monitored. Motor overheating that can be caused by continuous force output will again cause the servo-loop to terminate.

Despite the strengths of the PHANTOM and Ghost SDK, we must also take their weaknesses into account. The ease of use of the PHANTOMs comes from the fact that they are manufactured devices complete with customer support. The downside is that many of its features are proprietary. Much is unknown about the PHANTOM hardware specifications, but most importantly we are not given access to the Ghost SDK source code. This means that we do not know how the position data is computed from the encoder bit data. We also do not know how the force data is translated to voltage for the servomotors in the PHANTOM.

As mentioned earlier, the PHANTOM does not have tachometers, meaning velocity and acceleration data must be obtained by numerically differentiating the position data. Of course this produces noise that must be filtered. Although the encoders produce very precise position measurements down to 0.03 mm, given a sampling period of 0.001 s the precision for velocity measurement is actually 30 mm/s. A first-order low pass Butterworth filter can significantly reduce the noise, although an averaging algorithm has also been tested with satisfactory results. Despite the filtering, the noise still severely restricts the range of values we can assign to certain parameters such as derivative gain in the PD control and the wave impedance.

We used the Windows 2000 version of the Ghost SDK to control the PHANTOM. Windows 2000, unlike RT Linux, is not designed as a real-time OS. The sampling period is only an average at 1 ms. The servo-loop has been measured to skip up to 16 ms and then try to catch up by performing up to 16 loops in the next millisecond. In practice this is tantamount to inadvertently introducing variation in transmission delay. Fortunately the proposed algorithms are designed to compensate for transmission variations. Furthermore, the simulated transmission delay used in the experiment ranges from 400 ms to 600 ms. Hence the Windows 2000 servo-loop skipping effect is perceived as having little influence as the simulated delay variations dominate.

5.3 Hydraulically Actuated Lifter (HAL)

The study includes a short experiment involving the Hydraulically Actuated Lifter as the slave device controlled by the master PHANTOM 1.5. HAL was designed and built by the Intelligent Machine Design Laboratory at Georgia Tech to mimic a forklift. The intended purpose is to test the algorithm with transmission delay on a highly asymmetric master and slave setup. HAL can be seen as a prismatic robot with two degrees of freedom. It can use its rotary hydraulic motor to move horizontally on a track and a hydraulic actuator to move the end-effector vertically. The hydraulic motor and actuator runs on hydraulic fluid at about 3000 psi. The workspace size is about 3 ft in the vertical direction and 5 ft in the horizontal direction. Encoders in both directions provide position feedback. A picture of HAL is shown in Fig. 5.4.

The control codes that run HAL is written in C++ on RT Linux Kernel that runs on top of Red Hat Linux operating system. A Servo-to-Go card was used to interface between the control codes and HAL sensors and actuators. The control codes are heavily based on previous codes by Kontz [69], but modified to incorporate the delay compensating algorithm. A brief description of the computer used to control HAL is given in Table 5.4.

Table 5.4: HAL Computer Specifications

Operation System	Red Hat Linux 7.1
Kernel	RT Linux 2.4.3 – 12
Processor	Pentium III 500 MHz
Memory	64 MB
I/O Card	Servo-to-Go, Model II, 8 axis, ISA Bus

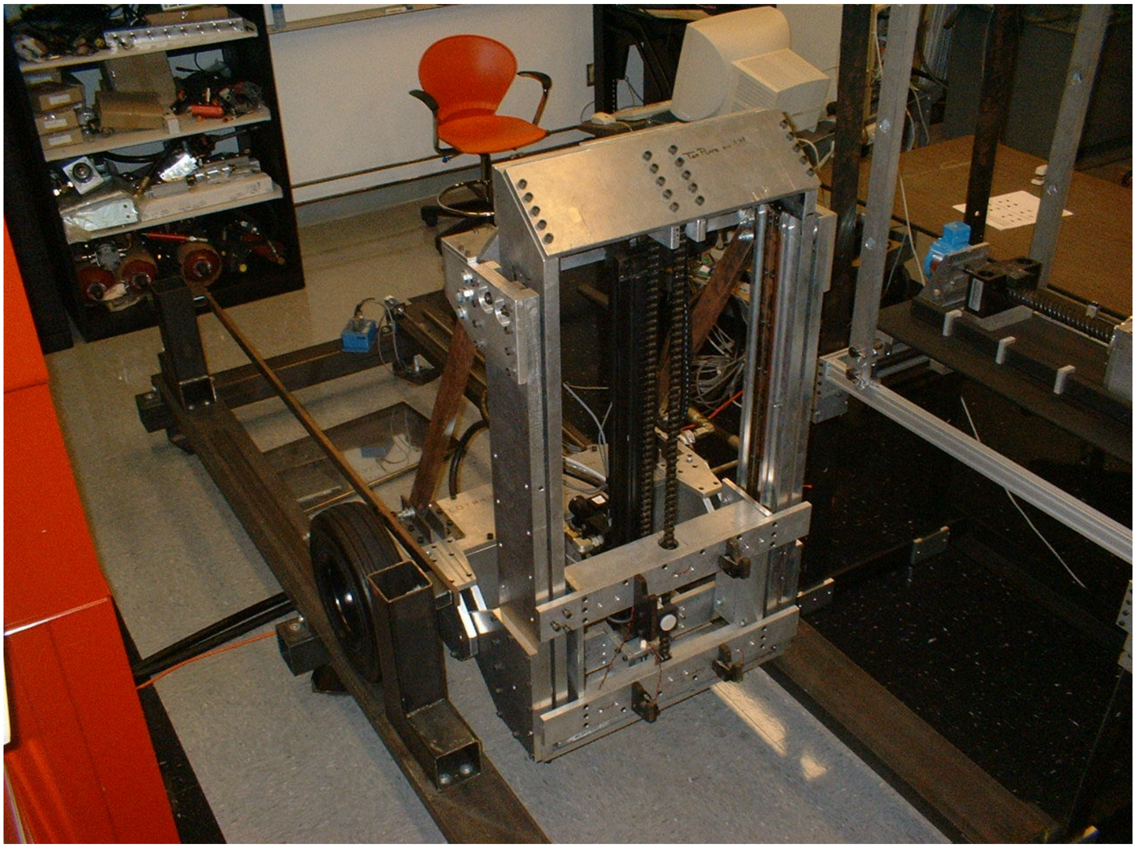


Figure 5.4: Hydraulically Actuated Lifter (HAL)

As it turns out, there are several properties of HAL that makes it less-than-satisfactory as a test bed for the wave-based algorithms. First, the horizontal direction is difficult to operate due to severe backlash of the rotary hydraulic motor. Such a motor is designed for continuous motion rather than abrupt changes in directions. Such dynamic response can be highly undesirable even in a conventional teleoperation.

Given the hydraulic actuator is flow-controlled; the output of the PD control is actually the opening size of the servo-valves. HAL velocity is roughly proportional to the error between desired and actual positions. Hence force feedback is computed using a virtual coupling that directly proportional to the error between the desired and actual slave positions. While this works well in free space environment, it does not work well in rigid contact case when the slave would always push against the contact with maximum pressure until the desired and actual slave positions agree. This is in contrast to the PHANTOM actuator that is effectively force-controlled; the force it applies towards the contact surface is directly proportional to the error between the desired and actual positions. For this reason, experiments using HAL is limited to only free space environment.

HAL can theoretically be modeled as a linear fourth-order plant. However, HAL's velocity in both directions is limited by the size of the servo-valve openings. This produces a saturation effect that is nonlinear. It appears this saturation effect is more severe with the vertical axis. We must also note that due to the difference in actuator piston area in the two vertical directions, there are two different velocity limits for the same vertical hydraulic actuator. This asymmetry is less apparent in the horizontal direction. Hence the predictor is technically nonlinear although effectively linear when the speed is low enough.

One characteristic that can be fully tested using HAL is the high asymmetry between the master and slave size. The scaling of desired position and force feedback is performed on the slave side. Incoming desired position in PHANTOM coordinate is

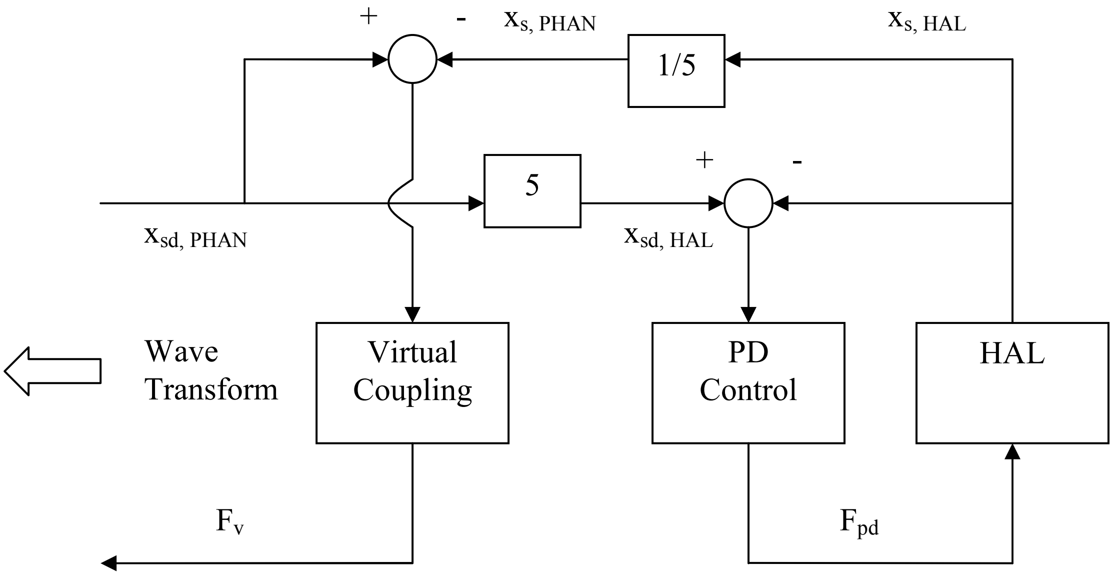


Figure 5.5: HAL Side Block Diagram

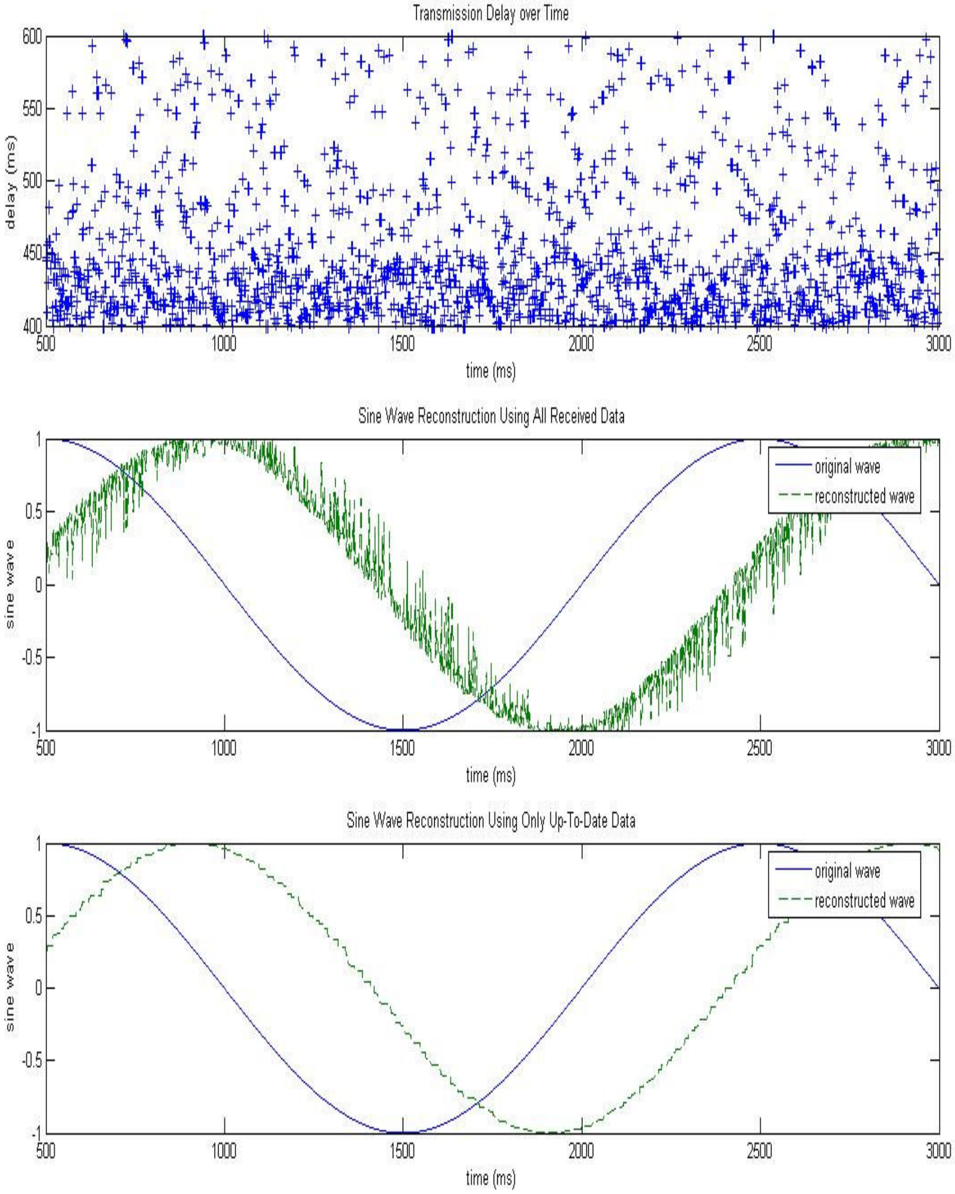
multiplied by a factor of 5 to produce the HAL coordinate in both axes. The force feedback is computed from the position error between the desired position and actual HAL position in Phantom coordinates multiplied by a properly selected gain. In effect the virtual coupling provides the force scaling. A modified block diagram of the HAL is shown in Fig. 5.5.

5.4 Transmission Delay

The transmission delay is the only simulated part of the experiment, although the computers controlling the master and slave will still be connected via the Internet. The master and slave are located in the same room; hence the nominal transmission delay between them should be negligible. Previously Munir and Book [21] used actual Internet transmission delay where the master and slave are both in Georgia Tech but the data are bounced off a reflector computer in France or Japan. Although a simulated transmission delay used in this study is not as realistic, it does offer more flexibility in adjusting the length of the delay as well as the variations.

The method of transmission is User Datagram Protocol (UDP) instead of the Transmission Control Protocol (TCP). TCP is a confirmation-based protocol where the receiving computer, upon reception of the sender's data stream, would send back a confirmation. If the sending computer does not receive the confirmation within a reasonable roundtrip time, the data stream in question would be retransmitted. This produces a very reliable data transfer system with 100% of the data transfer completion, making it an ideal protocol for common Internet web-surfing. However, since the sending computer frequently has to wait for confirmation to arrive, the data stream does not usually arrive in the correct order. In transmission of data for controlling robots, the order in which the data arrives is just as important as if not more so than the completion of the data transfer. On the other hand, UDP is not a confirmation-based Internet transmission protocol. The sending computer using this protocol breaks the data into packets and transmits them. The receiving computer does not send a confirmation back to the sending computer once it has received the data packets. Hence there is no retransmission in case of data packets that do not make it to the destination. While this means data transmission is not as reliable from a completion point of view, it has a better chance of arriving in the correct order. This makes UDP more suitable for applications where real-time performance is important. A comparison between the TCP and UDP transmissions can be seen in Fig. 1.4 and 1.5 in Chapter 1. The plots show that the transmission delays in UDP to be more stable and predictable than TCP. More importantly the receiving computer is better able to reconstruct the sine wave being sent by the sending computer using UDP.

The simulated delay can not simply be a data storage element acting like a buffer; such an element will be fine if the simulated delay is to be constant. It has to be modified to provide variations in delay as well as a possibility of losing data packets. A random number generator is used to compute the probability of variations. As seen in the typical UDP delay, the delay is mostly constant and stable at a baseline minimum delay, with


Transmission Delay over Time

Sine Wave Reconstruction Using All Received Data

Sine Wave Reconstruction Using Only Up-To-Date Data

Figure 5.6: Simulated Transmission Delay

occasional spikes and also possible lost data packets. In other words, the delay is skewed or biased towards the baseline delay. The simulated delay is designed such that a given data packets is stratified. In this case, each data packet would have a 50% chance of being subjected to between 400 and 450 ms delay, 25% chance between 450 and 500 ms delay, 15% chance between 500 and 600 ms delay, and 10% chance of total loss of data packet. Qualitatively this delay characteristic reflects the type of delay from UDP as shown in Fig. 1.5, except the simulated delay is more severe in terms of duration. The stratification and the delay durations can be modified to suit different transmission delay scenarios, but the stratification mentioned above would be used for all the experimental results described in this chapter. The plots of the simulated delay and reconstructed sine waves are shown in Fig. 5.6. Noticed if the receiving computer uses all the received data, the sine wave is not well reconstructed because outdated data are used. Taking this into account, the transmitting computer can include a time stamp along with the sent data packet. The receiving computer can check the time stamps and only use data with the most recent time stamps. Any data packets with outdated time stamps are discarded immediately. This method allows for better sine wave reconstruction. This simulated delay scheme is used for both PHANTOM and HAL systems

5.5 The Entire Setup

This section describes how the entire setup is put together for both PHANTOM-to-PHANTOM and PHANTOM-to-HAL teleoperations. As mentioned above, the master and slave computers are connected to each other via the Internet. Due to the real-time nature of the application, we made a special effort to make the codes as compact as possible. As a side-effect, the codes are not very modular, and this makes debugging extremely difficult. Nevertheless we are able to reduce the number of threads per computer to two for the PHANTOMs: the servo-loop thread that holds the control codes and sends the data to the other computer, and the UDP reception thread that receives data

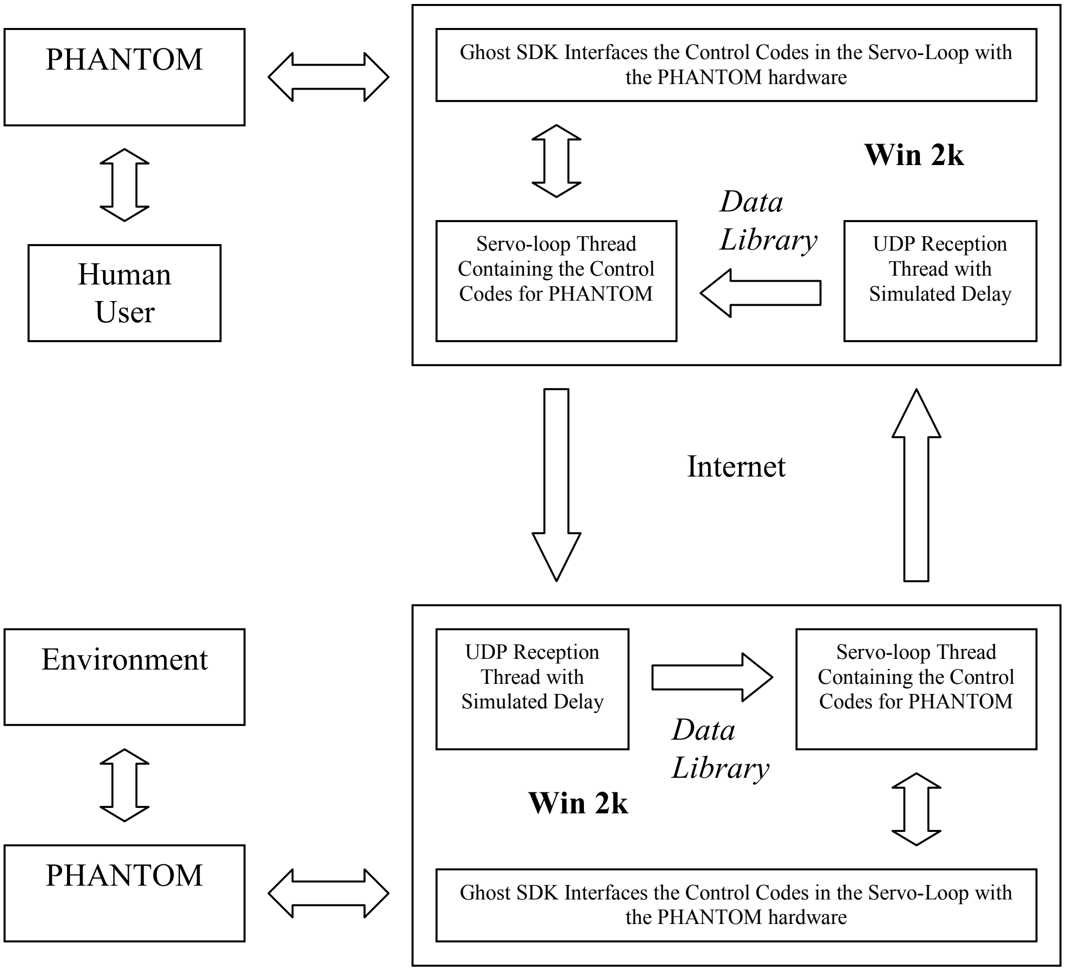

PHANTOM
Human User
Ghost SDK Interfaces the Control Codes in the Servo-Loop with the PHANTOM hardware
Win 2k
Data Library
Servo-loop Thread Containing the Control Codes for PHANTOM
UDP Reception Thread with Simulated Delay
Internet
Environment
UDP Reception Thread with Simulated Delay
Servo-loop Thread Containing the Control Codes for PHANTOM
Data Library
Win 2k
PHANTOM
Ghost SDK Interfaces the Control Codes in the Servo-Loop with the PHANTOM hardware

Figure 5.7:   PHANTOM-to-PHANTOM Schematic

packets from the other computer. The Ghost SDK may have its own separate threads, however we do not know for sure since the source code for the SDK is not available. The HAL code contains three threads: the servo-loop thread that holds the control codes, the UDP reception thread, and the UDP transmission thread. Figure 5.7 and 5.8 show the schematics for PHANTON-PHANTOM and PHANTOM-HAL teleoperation respectively.

In case of the PHANTOMs, the servo-loop thread asks the Ghost SDK for position and velocity data of the PHANTOM every millisecond on average. At the same time the servo-loop computes a force vector to be read by the Ghost SDK. In case of the wave variable with adaptive predictor and drift control algorithm, the master computer

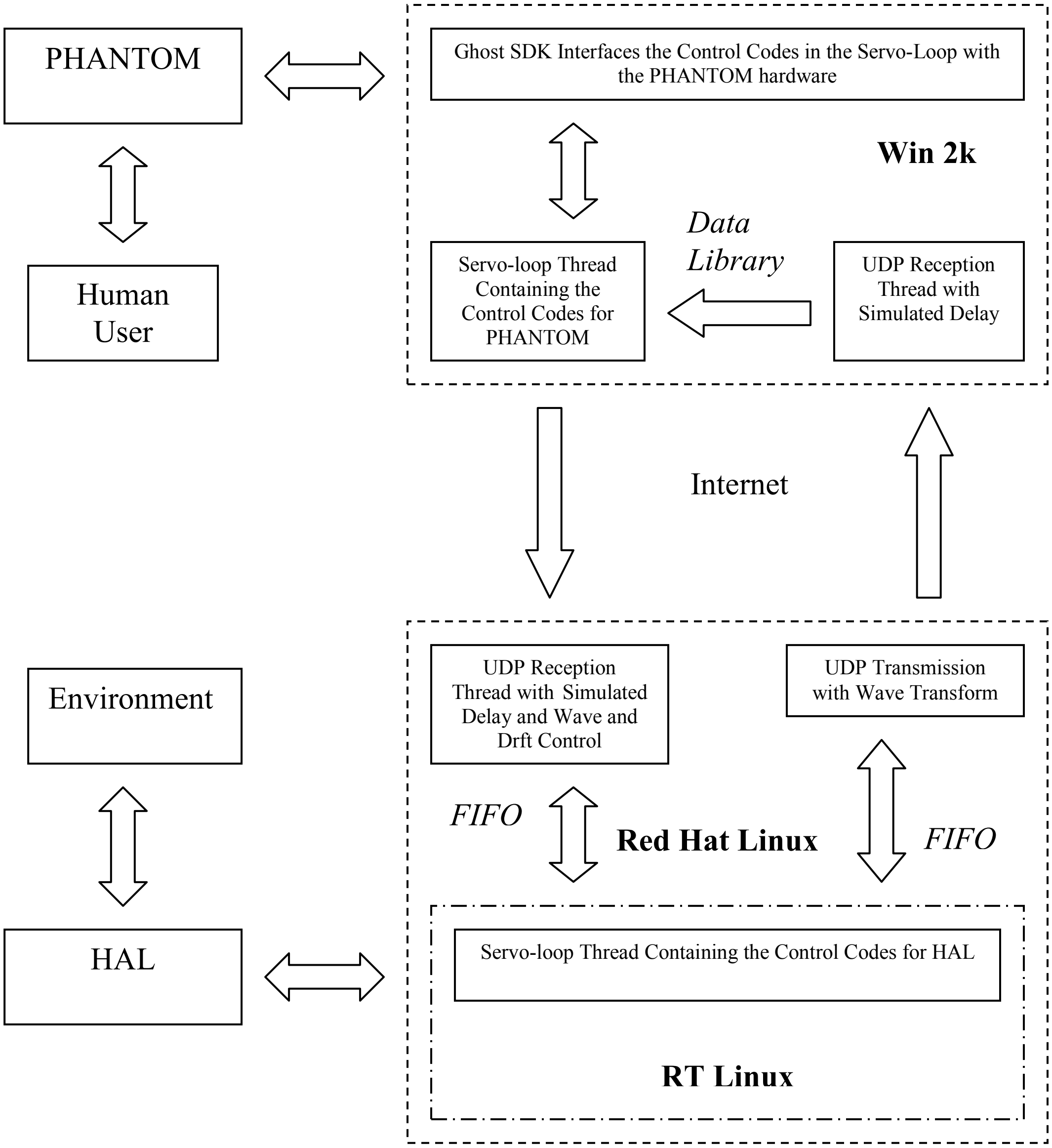

PHANTOM
Human User
Ghost SDK Interfaces the Control Codes in the Servo-Loop with the PHANTOM hardware
Win 2k
Data Library
Servo-loop Thread Containing the Control Codes for PHANTOM
UDP Reception Thread with Simulated Delay
Internet
Environment
HAL
UDP Reception Thread with Simulated Delay and Wave and Drft Control
UDP Transmission with Wave Transform
FIFO
Red Hat Linux
FIFO
Servo-loop Thread Containing the Control Codes for HAL
RT Linux

Figure 5.8: PHANTOM-to-HAL Schematic

servo-loop includes computations for the wave transform, predictor, and regulator algorithms. On the slave side, the servo-loop includes computations for the wave transform, PD control, drift control, and adaptive filter algorithms. Embedded in the servo-loops are also the codes needed to directly send the required data to the other computer via UDP. In addition to communication with the Ghost SDK and sending data to the other computer, the servo-loop also needs to receive data from the other computer via the UDP reception thread. Data reception code is placed in a separate thread because data packet reception on a timely manner is not guaranteed. Hence the computer needs to constantly listen for incoming data packets on a separate dedicated thread. Data transfer between the threads is accomplished using a data library. The UDP reception constantly writes into the library as new incoming data from the other computer becomes available while the servo-loop reads from it. In addition to listening for UDP packets, the UDP reception thread also subjects the incoming data to simulate transmission delay. The simulated transmission delay is another data library that temporarily holds the incoming data for a variable amount of time. Sometimes it deletes the data based on the delay stratification.

The HAL schematic is very similar to PHANTOM, except an addition thread is included to send data to the other computer. The servo-loop thread is run on the RT Linux kernel every one millisecond and directly receives data from and transmits data to HAL via the I/O card. The UDP reception and transmission threads are run on the Red Hat Linux OS. Communications between the servo-loop thread and UDP reception/transmission threads are perform via FIFO. Real time performance is better with HAL computer OS with the servo-loop running on time at 1 kHz without skipping. Besides this there is no observable differences in performance that can be attributed to different OS.

5.6 Experimental Results

A total of fifteen experiments are performed to verify the algorithms mentioned in Chapter 4 under both free space and rigid contact environments using both PHANTOMs and HAL. In eleven of them, PHANTOM 1.0 is used as master while PHANTOM 1.5 is used as slave. In most of these experiments, only the result of the x-axis is shown, although results from the other two axes would have appeared similar as shown in the conventional results without delay. In four of them, PHANTOM 1.5 is used as master while HAL is used as slave. Unless stated otherwise, the transmission delay is simulated to be similar to that described by Fig. 5.6. In free space experiments, the author tries to generate an abrupt step input to the master. Contact surface used as rigid contact environment is made of hard wood, hence effectively rigid. In all plots, the solid blue line represents the master while the dashed green line represents the slave. However, all the data is recorded on the master side where the master data is recorded immediately while the slave data is recorded as soon as it is received by the master. Hence the slave plot should lag behind the master plot by the given approximate round trip delay, nominally at around 800 ms. The fifteen experiments performed for this study can be summarized as follows.

- Three experiments of conventional bilateral teleoperation of the type shown in Fig. 1.2 are performed: two without delay under free space and rigid contact, one with delay under free space. PHANTOM 1.0 used as master, and PHANTOM 1.5 used as slave.
- Three experiments of wave variable teleoperation of type shown in Fig. 4.8 are performed: one with delay under free space, two with delay under rigid contact but with different values of wave impedance. PHANTOM 1.0 used as master, and PHANTOM 1.5 used as slave.
- One experiment of wave predictor teleoperation of type shown in Fig. 4.9 is performed with delay under free space. PHANTOM 1.0 used as master, and PHANTOM 1.5 used as slave.
- One experiment of wave predictor with drift control teleoperation of type shown in Fig. 4.10 is performed with delay under free space. PHANTOM 1.0 used as master, and PHANTOM 1.5 used as slave.
- Three experiments of wave adaptive predictor with drift control teleoperation similar to Fig. 4.11 but with the adaptive algorithms are performed with delay under rigid contact: one with semi-adaptive predictor with two modes, one with full-adaptive predictor estimating only environmental force, one with full-

adaptive predictor estimating contact rigidity as well as location of contact. PHANTOM 1.0 used as master, and PHANTOM 1.5 used as slave.

- Four experiments are performed using PHANTOM 1.5 as master and HAL as slave: one conventional bilateral teleoperation in free space without delay, one conventional bilateral teleoperation in free space with delay, one wave variable teleoperation in free space with delay, one wave predictor with drift control in free space with delay.

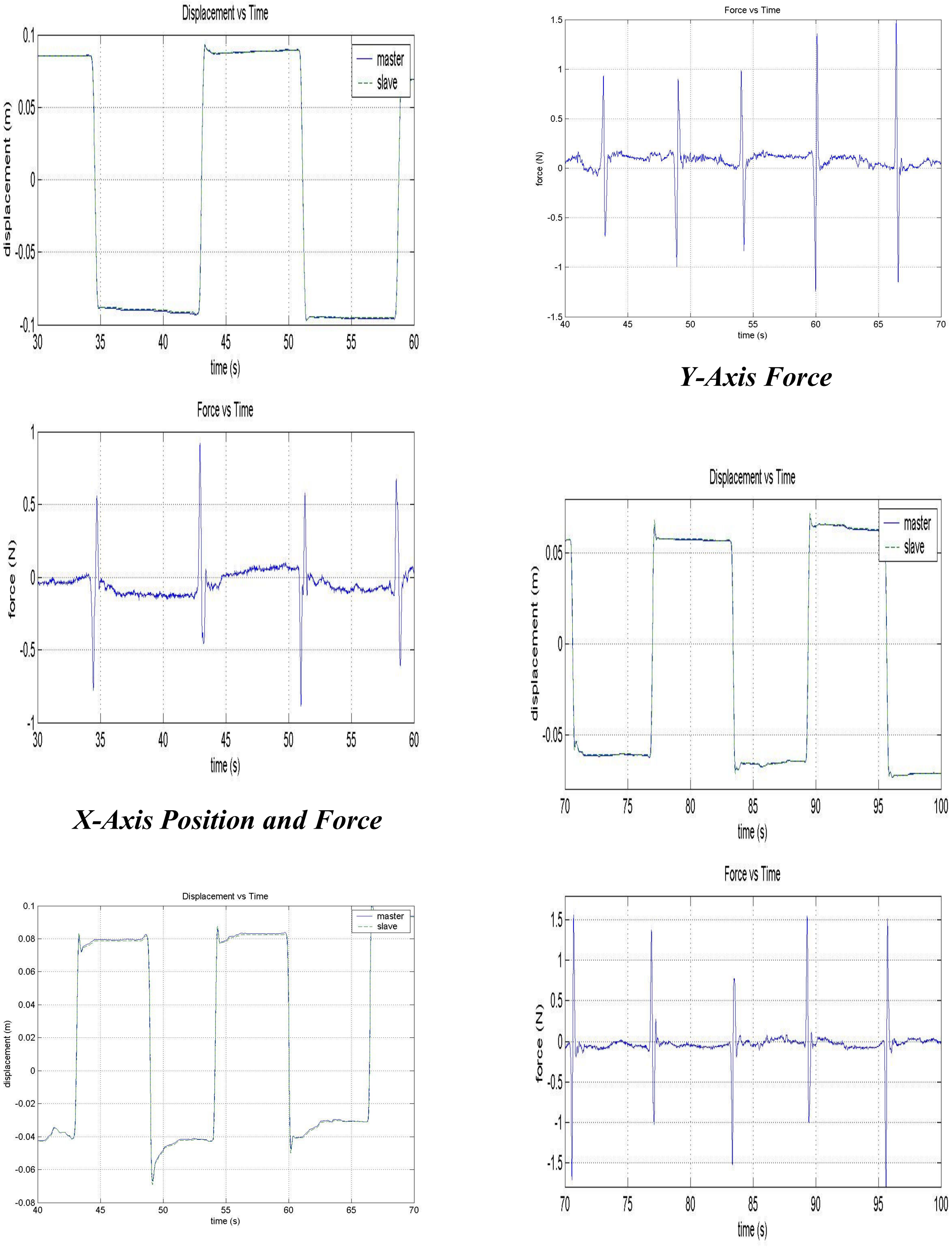
5.6.1 PHANTOM-PHANTOM Conventional Bilateral Teleoperation

Three experiments of conventional bilateral teleoperation of the type shown in Fig. 1.2 are performed. The two experiments without delay under free space and rigid contact are performed to provide a benchmark for comparison with other algorithms. The one with delay under free space is used to demonstrate the instability caused by transmission delay. All three axes are tested in this case.

In the free space environment without delay (Fig. 5.9), we can see that trajectory tracking is virtually perfect. Force remains roughly at zero until the operator provides quick motion approximating step input in position when there are two short spikes in master force: one to initialize and one to terminate the motion. Slave force is not shown because it is the same as the master force since PD control force on the slave side is fed back directly to the master.

In the rigid contact environment without delay (Fig. 5.10), we can see that trajectory tracking is virtually perfect until the point of contact at roughly zero position. At this point the master pushes into the contact surface to produce position error that leads to steady feedback force of over 3 N.

In the free space environment with delay (Fig. 5.11), we can only see severe oscillations. The operator had his hand on the master to constrain the vibration; otherwise the oscillation magnitude would have increased exponentially. The



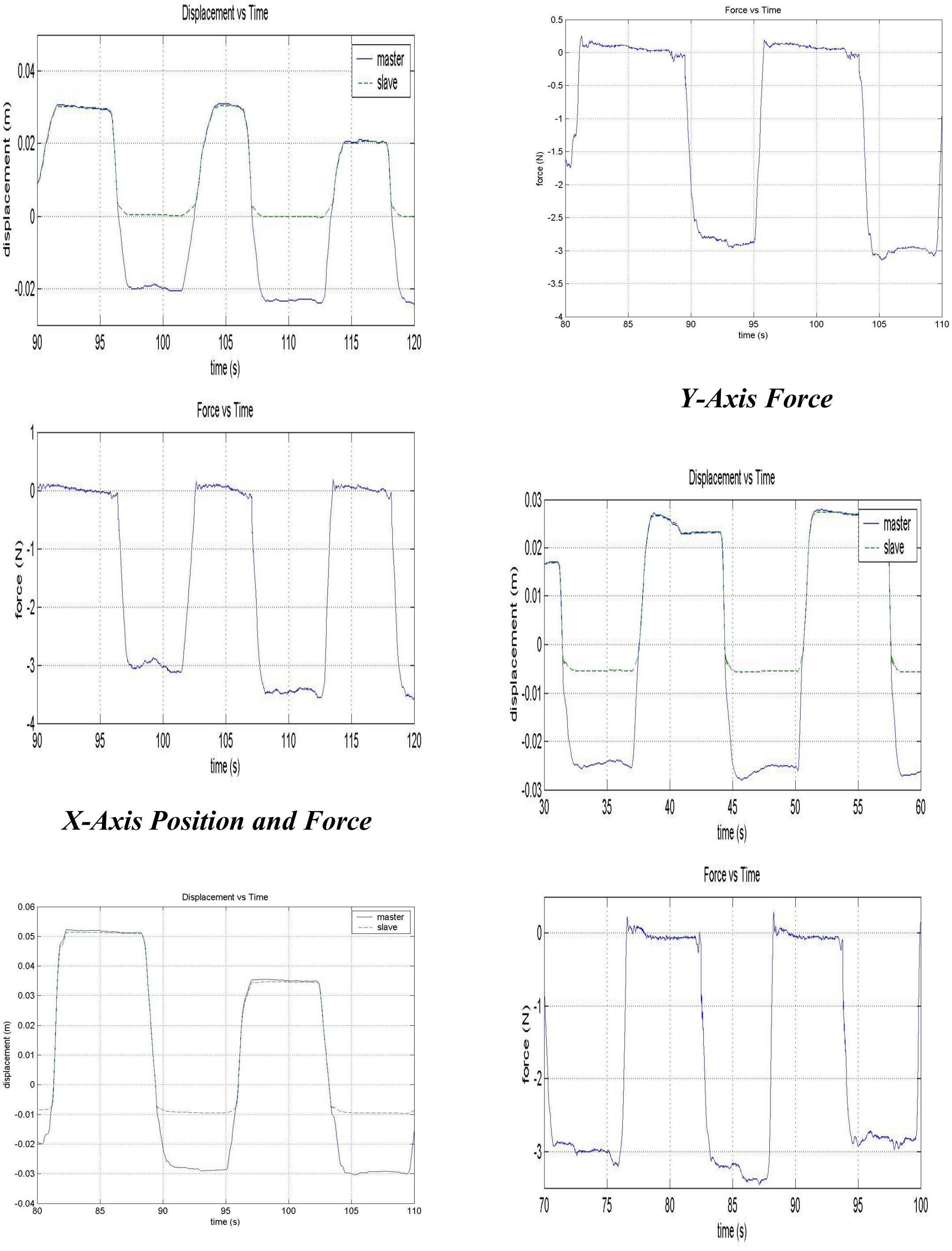
Y-Axis Force

X-Axis Position and Force

Y-Axis Position

Z-Axis Position and Force

Figure 5.9: Conventional Bilateral Teleoperation in Free Space without Delay



X-Axis Position and Force

Y-Axis Force

Y-Axis Position

Z-Axis Position and Force

Figure 5.10: Conventional Bilateral Teleoperation in Rigid Contact without Delay

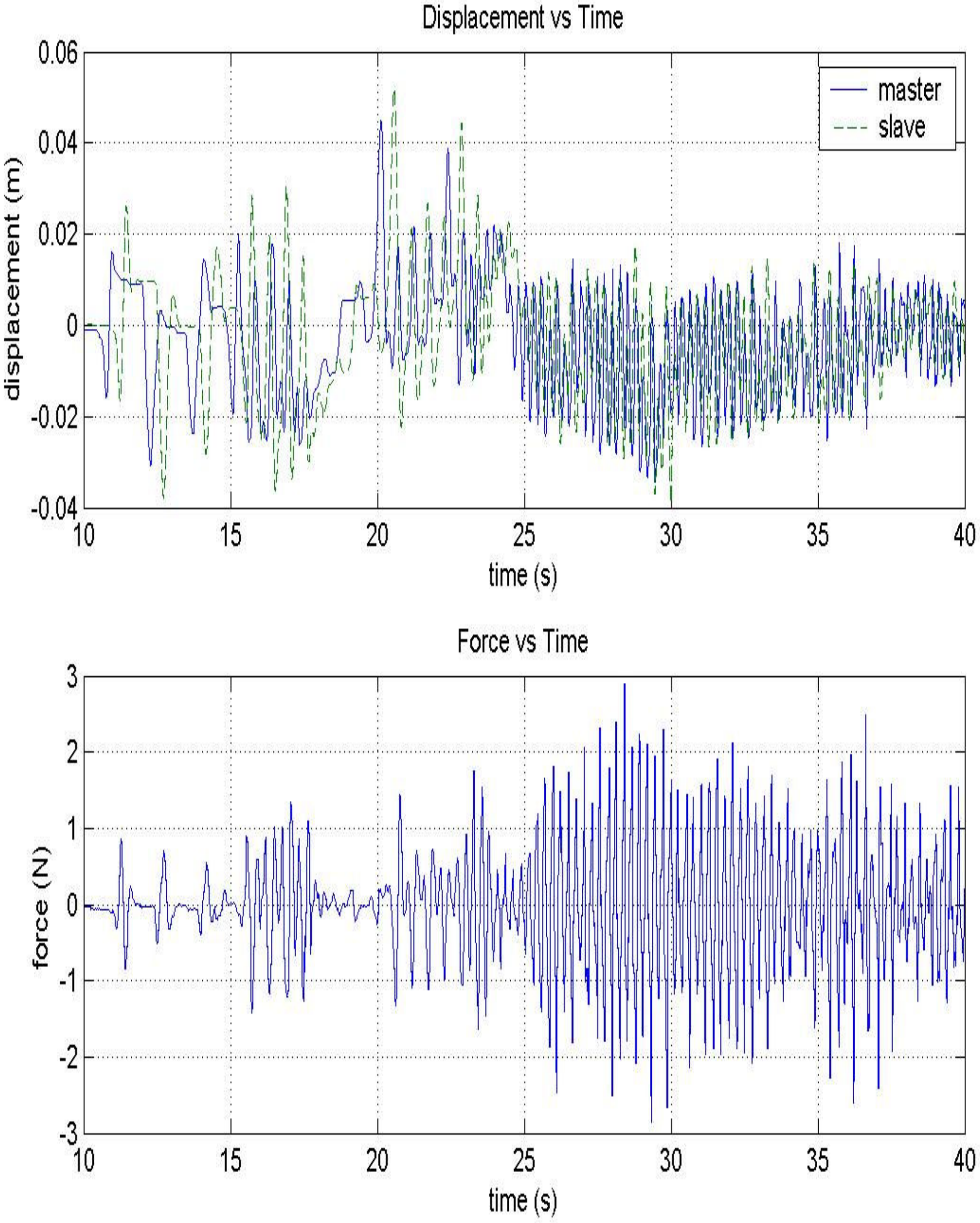


Figure 5.11: Conventional Bilateral Teleoperation in Free Space with Delay

PHANTOM without the operator constraint would have saturated its motor in a few seconds. This clearly shows the instability caused by the transmission delay.

5.6.2 PHANTOM-PHANTOM Wave Variable Teleoperation

Three experiments of wave variable teleoperation of type shown in Fig. 4.8 are performed. One is in free space with delay to demonstrate the stability and transient of the wave variable algorithm. Two are performed under rigid contact with delay but with different wave impedances to demonstrate their effect to rigid contact performance.

In the free space environment with delay (Fig. 5.12), the human operator applies abrupt changes to the master position to approximate step inputs. The result clearly shows the poor transient behavior where the slave oscillation took around five seconds to dissipate. Notice that the amplitude of the master force is much larger than that of the slave. The reason is that the user is holding the master position to be roughly constant during the transient oscillations. Since the master position does not yield, the wave variable algorithm requires the force amplitude to increase.

In the rigid contact environment with delay (Fig. 5.13), we can see the oscillatory transient response as the slave comes into contact at roughly zero position. More importantly, we can see that at steady state, the master has to push very deep into the contact, about 0.08 m, to get a little over 0.8 N feedback force. The wave impedance is too low compared to the sudden increase in impedance caused by the contact on the slave side. The rigid contact steady state characteristic can be improved by increasing the wave impedance. Figure 5.14 also shows wave variable algorithm in rigid contact with delay, but the wave impedance has been increased from 4 N/ (m/s) to 12 N/ (m/s). Now we can see that the master only has to push about 0.05 m into the contact to get about 1.2 N feedback force. In PHANTOM's case, noise from the numerically measured velocity data limits the choice allowed for wave impedance values; any wave impedance higher 12 N/ (m/s) results in unacceptable vibrations. Increasing the wave impedance can

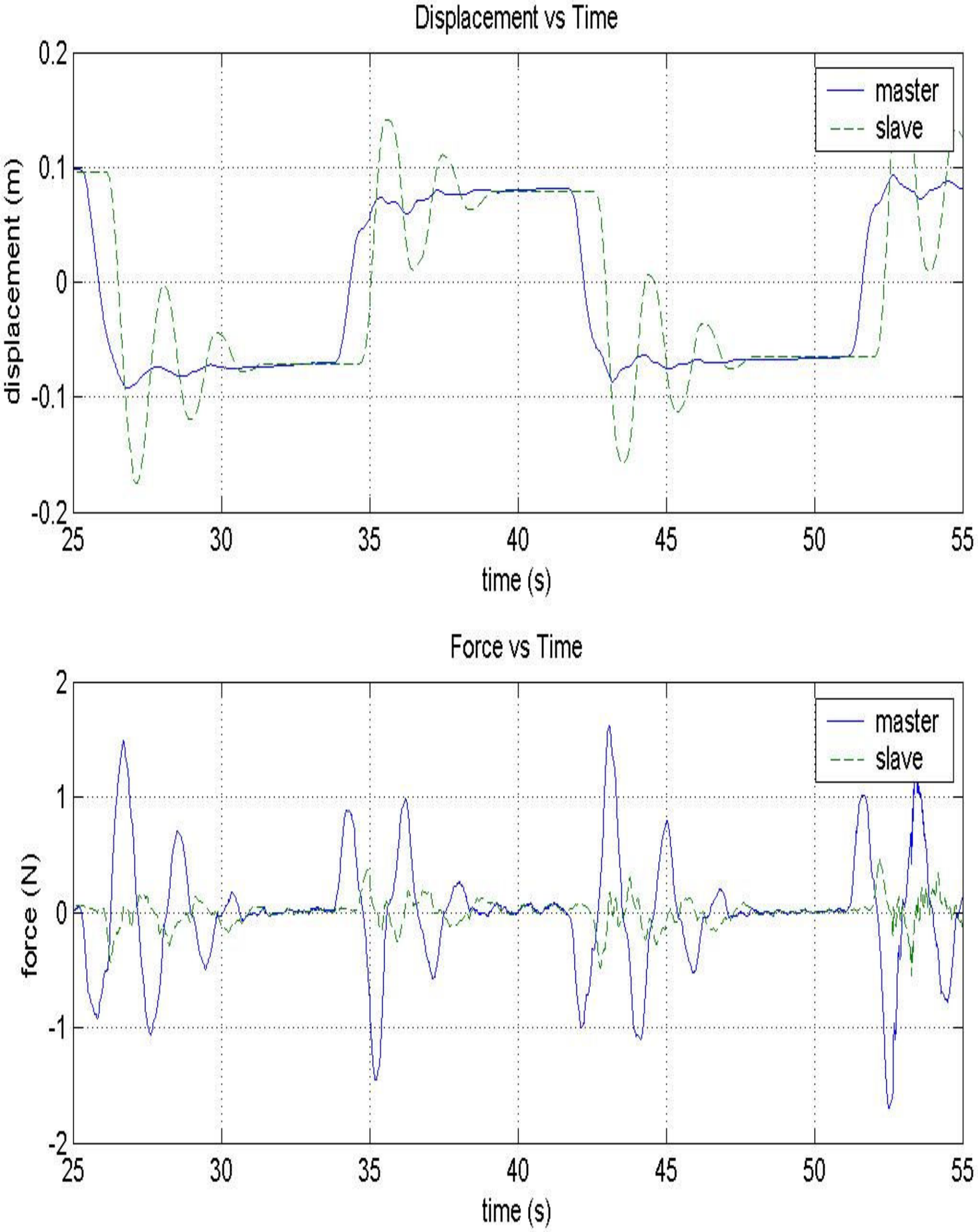


Figure 5.12: Wave Variable Teleoperation in Free Space with Delay

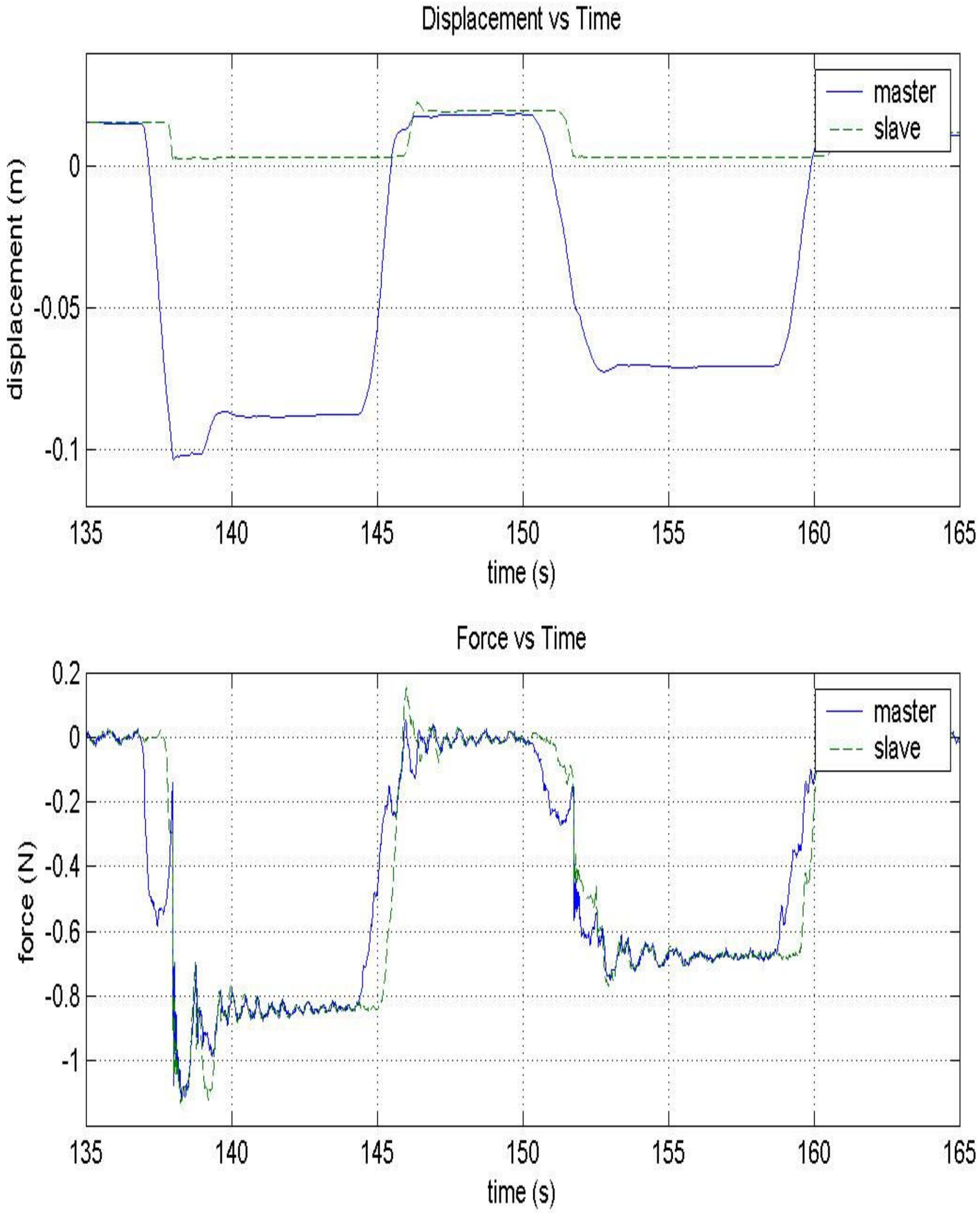


Figure 5.13: Wave Variable Teleoperation in Rigid Contact with Delay, Low Wave Impedance

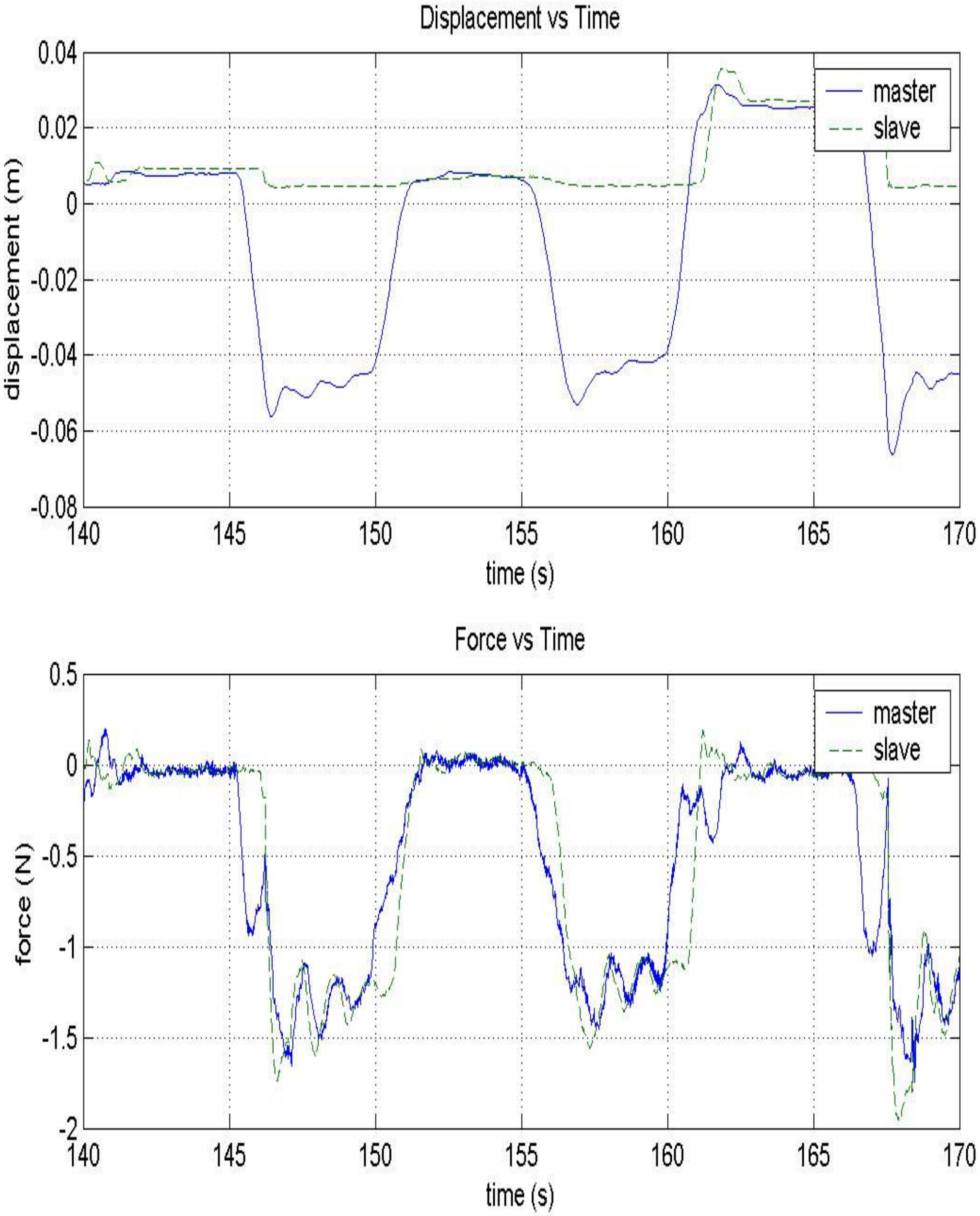


Figure 5.14: Wave Variable Teleoperation in Rigid Contact with Delay, High Wave Impedance

deteriorate performance in high admittance environment such as in free space. A compromise has to be achieved between high impedance and admittance performance when choosing the value for wave impedance.

5.6.3 PHANTOM-PHANTOM Wave Predictor Teleoperation

One experiment of wave predictor teleoperation of type shown in Fig. 4.9 is performed with delay under free space. The resulting plot is shown in Fig. 5.15. Again the human operator makes sudden changes in position similar to the previous cases. The results clearly show that while the transient has been significantly improved with only one overshoot and no oscillations, there is a significant amount of steady state position error. The cause of this error is primarily the result of inaccurate model used in the predictor. Notice that although the master force does not exactly match the slave force due to predictor model error, it does lead the slave force. The predictor is trying to anticipate the force feedback before the actual signal arrives from the slave.

5.6.4 PHANTOM-PHANTOM Wave Predictor with Drift Control Teleoperation

One experiment of wave predictor with drift control teleoperation of type shown in Fig. 4.10 is performed with delay under free space. This experiment shows the effectiveness of the direct control algorithm in eliminating the problem of the wave predictor teleoperation. The result of that is shown in Fig. 5.16 under similar input from the user and in free space. As one can see, the steady state position error has almost completely died out. The drift control algorithm also appeared to have improved transient by reducing the overshoot. Again the predictor is producing a lead in the master force feedback over the slave force as it tries to anticipate the slave force.

5.6.5 PHANTOM-PHANTOM Wave Adaptive Predictor with Drift Control Teleoperation

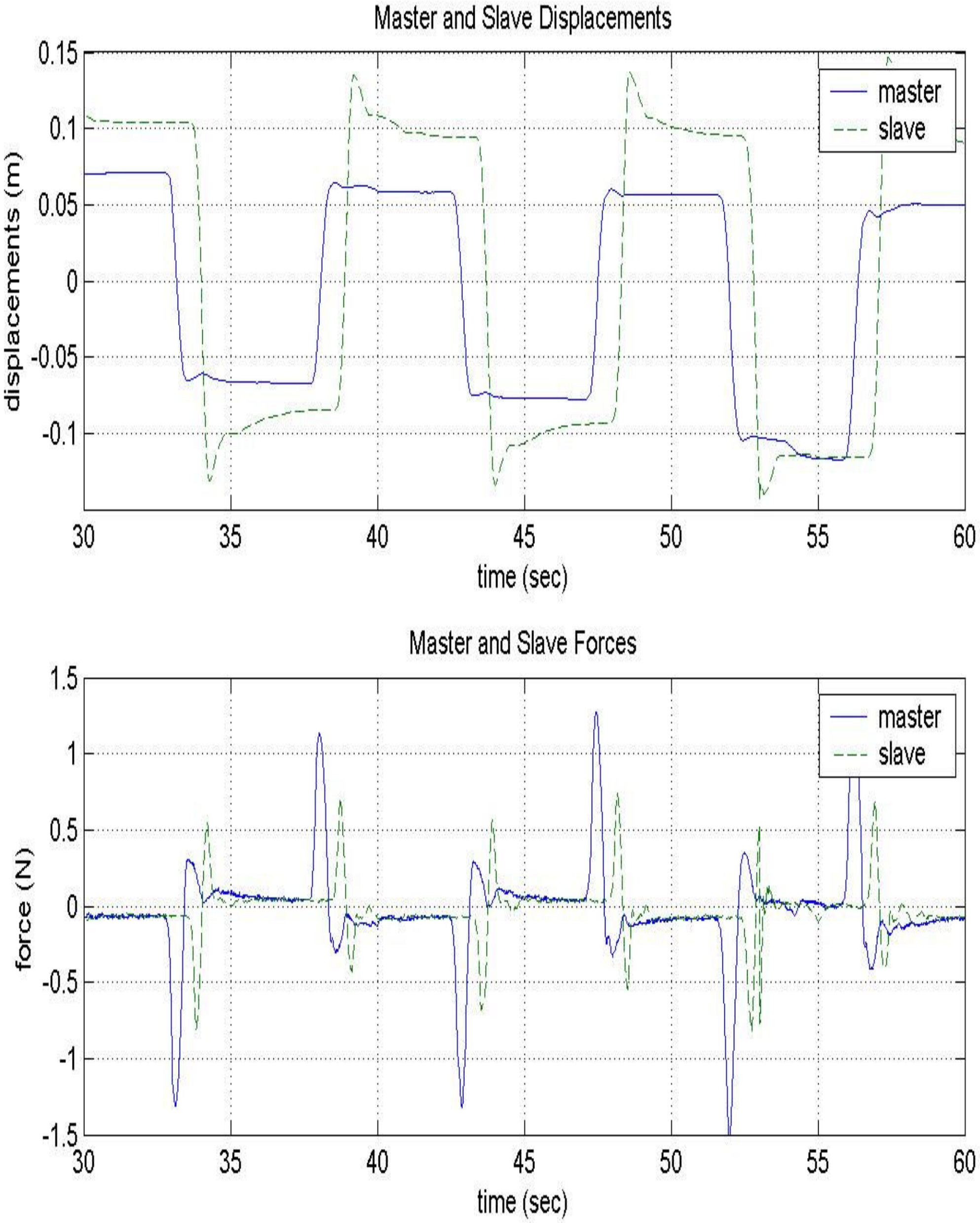


Figure 5.15: Wave Predictor Teleoperation in Free Space with Delay

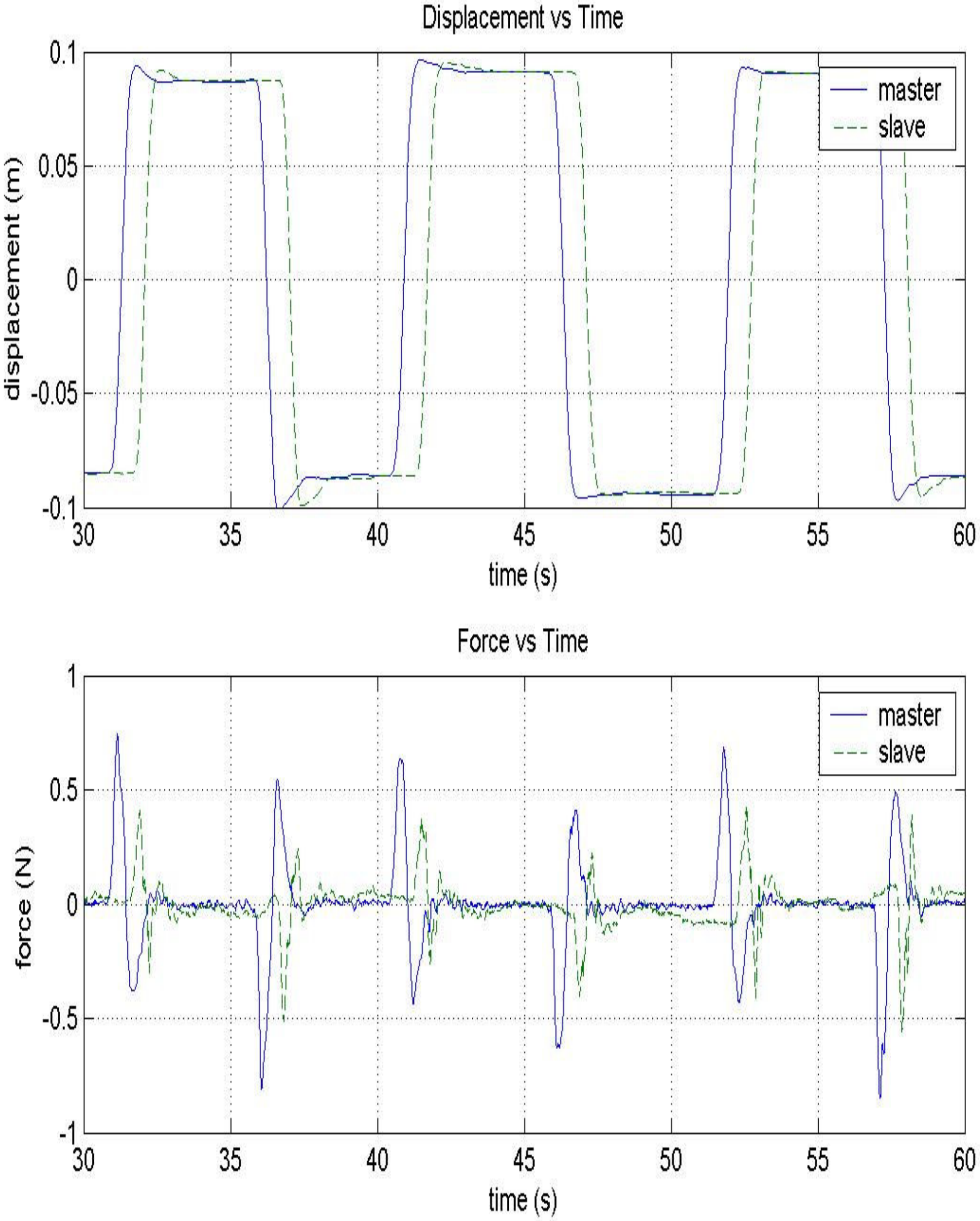
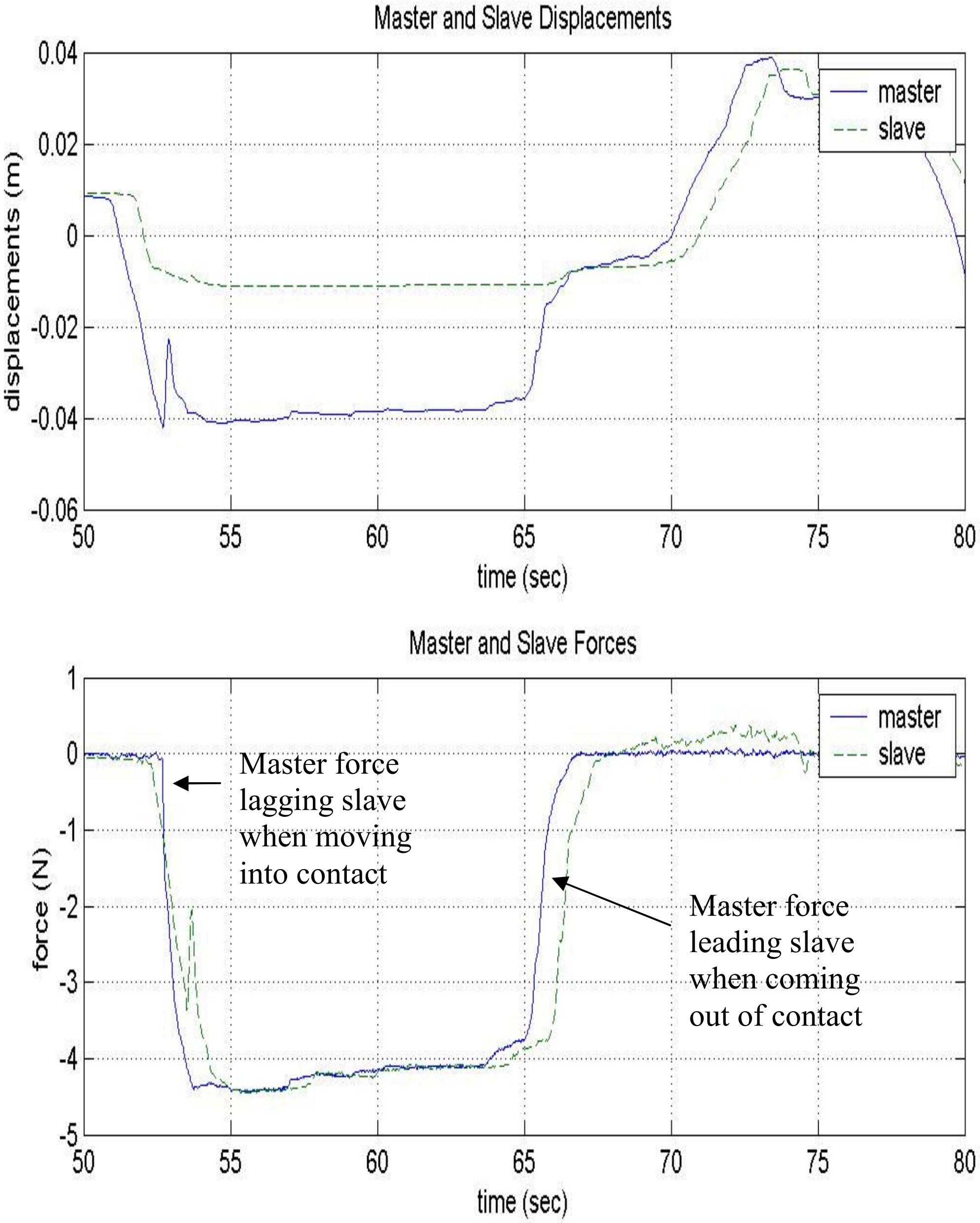


Figure 5.16: Wave Predictor with Drift Control in Free Space with Delay

Three experiments of wave adaptive predictor with drift control teleoperation similar to Fig. 4.11 but with the adaptive algorithms are performed with delay under rigid contact. One uses the semi-adaptive predictor where there are free space and rigid contact modes. Another uses the full-adaptive predictor where the environmental force is estimated by the recursive least square (RLS) method. The last one uses a RLS to estimate contact rigidity and the location of contact which is used to anticipate when the master would leave contact. Of particular interest is the transition between free space and rigid contact.

Figure 5.17 shows the result of using the semi-adaptive predictor in the wave variable domain with free space and rigid contact modes. The human user pushes the master into and out of contact with a surface located at around position -0.01. Naturally the positions do not track under contact, but the plots show that the forces track well. The master force does not lead the slave force just when the slave comes into contact because the predictor does not know beforehand where the location of the contact is. The predictor only switches models after the detection signal is received from the slave. However, as the slave goes back from contact to free space, the master force leads the slave force. Now the predictor knows where the contact is, it can anticipate when the master will return to free space and switch before it receives the detection signal from the slave. Notice there is a quick and temporary jump in position and force as the master is pushed into the contact. This is the point where the predictor switches from free space to rigid contact mode. The switch is abrupt such that the human user may have caught by surprise.

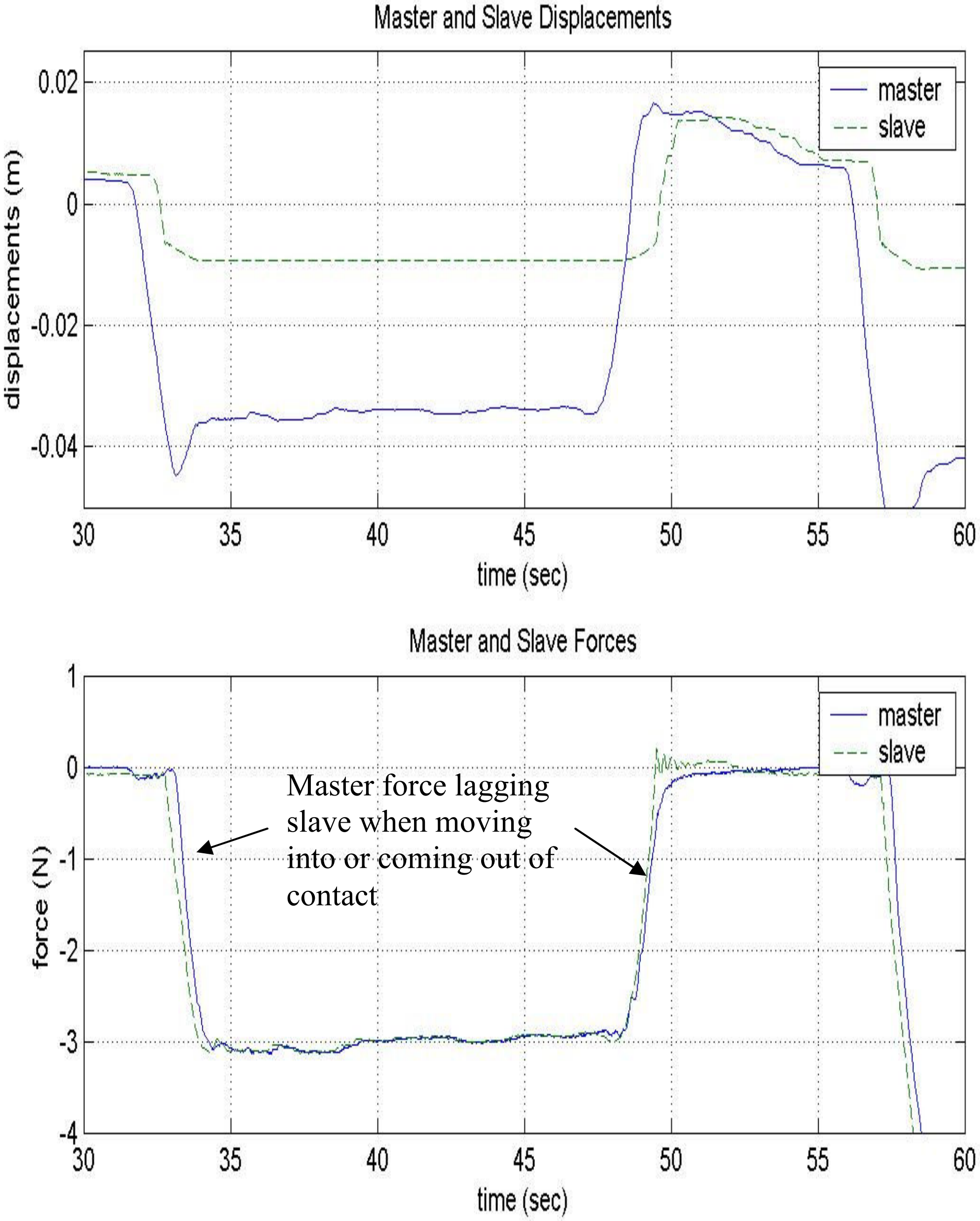
Figure 5.18 shows an experiment similar to that shown in Fig. 5.17, but instead of a semi-adaptive predictor, a full-adaptive predictor that estimates the environmental force is used. The results are similar to that of the semi-adaptive case with one key difference. When the slave returns to free space from contact, the feedback force as reconstructed by the master does not lead the slave force. This is because the predictor can not anticipate



Master force lagging slave when moving into contact

Master force leading slave when coming out of contact

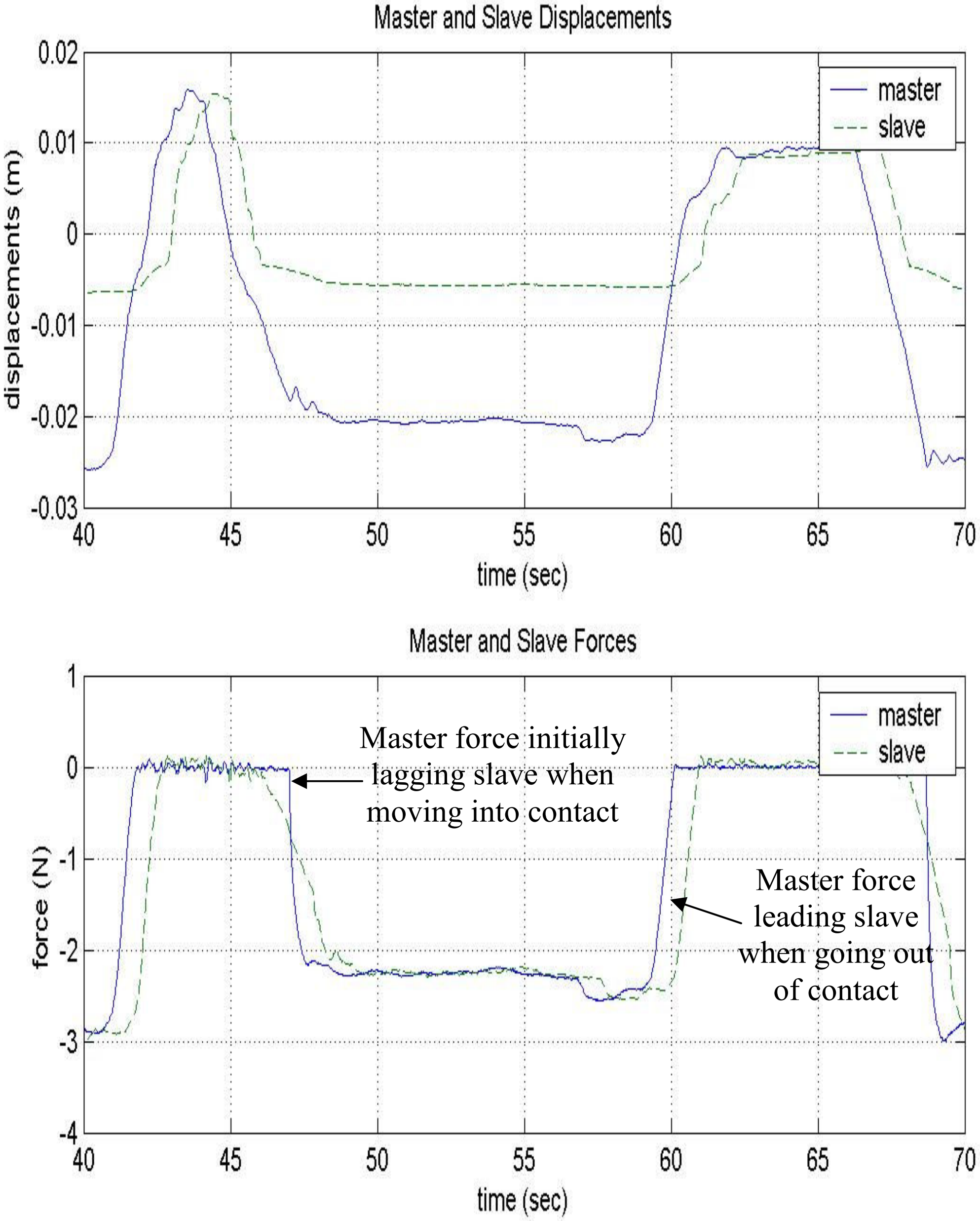
Figure 5.17: Wave Semi-Adaptive with Drift Control in Rigid Contact with Delay


Master and Slave Displacements

Master and Slave Forces

Master force lagging slave when moving into or coming out of contact

***Figure 5.18:** Wave Full-Adaptive Predictor with Drift Control in Rigid Contact with Delay, Environmental Force Estimation*



Master force initially lagging slave when moving into contact

Master force leading slave when going out of contact

Figure 5.19: Wave Full-Adaptive Predictor with Drift Control in Rigid Contact with Delay, Contact Rigidity Estimation

when the slave will come out of contact since the predictor model is entirely based on the estimated environmental force and has no knowledge of the location of the contact. In addition, one can notice that the feedback force on the master side lags a little behind the slave force even after the transmission delay is taken into account. This is the result of the time the RLS algorithm takes to adapt to the environmental force.

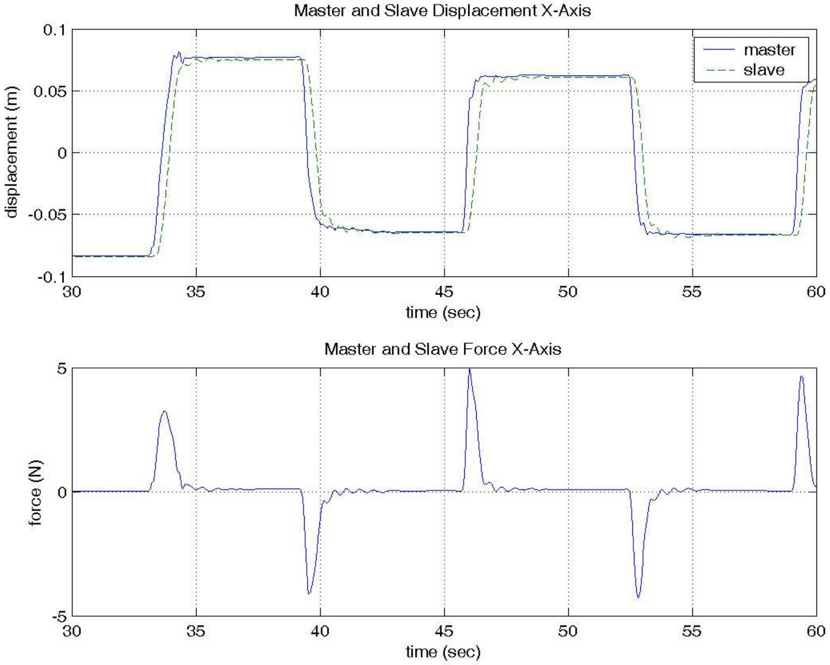
Figure 5.19 shows a similar experiment except this time a full-adaptive predictor is used to estimate the contact rigidity and point of contact. In addition it combines a feature with semi-adaptive predictor that anticipates when the master is going to come out of contact. At this point the predictor will automatically set the contact rigidity to zero and effectively becomes a free space model. The result shows that the performance is very similar to the semi-adaptive predictor case both when the system goes into and comes out of rigid contact. As the master goes into contact, the master force does not lead the slave because there is no a priori knowledge of the environment. However, the estimated contact rigidity builds up rapidly, and as soon as this information reaches the master, the predictor model quickly adapts and causes the master force to increase and overtake the slave force. Even though the estimated contact rigidity builds up quickly, it is not as abrupt as mode switching. This explains why the transition into rigid contact is somewhat smoother than the semi-adaptive predictor. Since the model estimates the contact point, it can anticipate when the master will return to free space which explains why the master force leads the slave force when the system comes out of contact.

From a human operator's perspective in tasks involving rigid contact, the full-adaptive predictor where only estimated environment force is the most counter-intuitive. The reason is the lag in contact force when the system goes into and comes out of contact. This delay seriously interferes with an operator's ability to pinpoint the location of contact. What many people find most unnatural is when persistency of contact force even when the master has supposedly moved back into free space. While the former lag is unavoidable without a priori knowledge of the slave environment, the later lag can be eliminated using the semi-adaptive predictor and the full-adaptive predictor that estimates contact rigidity and location. The anticipation of when the system moves out of contact and the subsequent dissipation of force without the lag significantly enhances the intuitiveness of the teleoperation. This makes the full-adaptive predictor that only estimates the environmental force to be the most inferior in environments that require frequent transitions between free space and rigid contact. This does not imply it is also worse in other situations, for example under loaded environments.

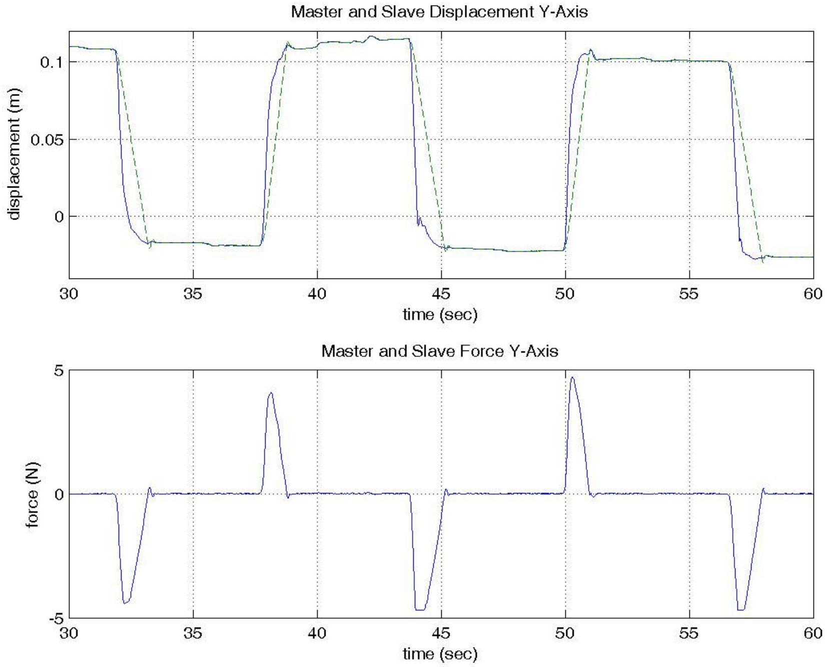
An adaptive predictor that can anticipate when the system moves out of contact still has the problem with the initial delay as the system moves into contact. This delay would result in the master pushing deep into the contact before the force feedback alerts the human user of the presence of the obstacle. This will result in a large feedback force trying to push the master back out into free space. If left unattended, this can catch the operator by surprise even though there is no feedback force once the master is in free space. The solution will be to add damping to the system either on the master or slave side to dampen the high feedback force that will throw the master into free space. This damping can also be adaptive that can be activated once contact is detected.

5.6.6 PHANTOM-HAL Teleoperation

The following four experiments are performed using PHANTOM 1.5 as master and HAL as slave. One is conventional bilateral teleoperation without delay. Another is conventional bilateral teleoperation with roughly 200 ms delay to demonstrate the instability caused by the delay. The wave variable algorithm is also tested as well as the wave predictor with drift control algorithm, both under roughly 400 ms delay. All experiments are performed in free space. Unlike PHANTOM-PHANTOM results, all results for both HAL axes are shown the axes are significantly different. Despite the limitation of HAL as mentioned above, these experiments still provide important insights and some verification of the applicability of the proposed algorithms across different

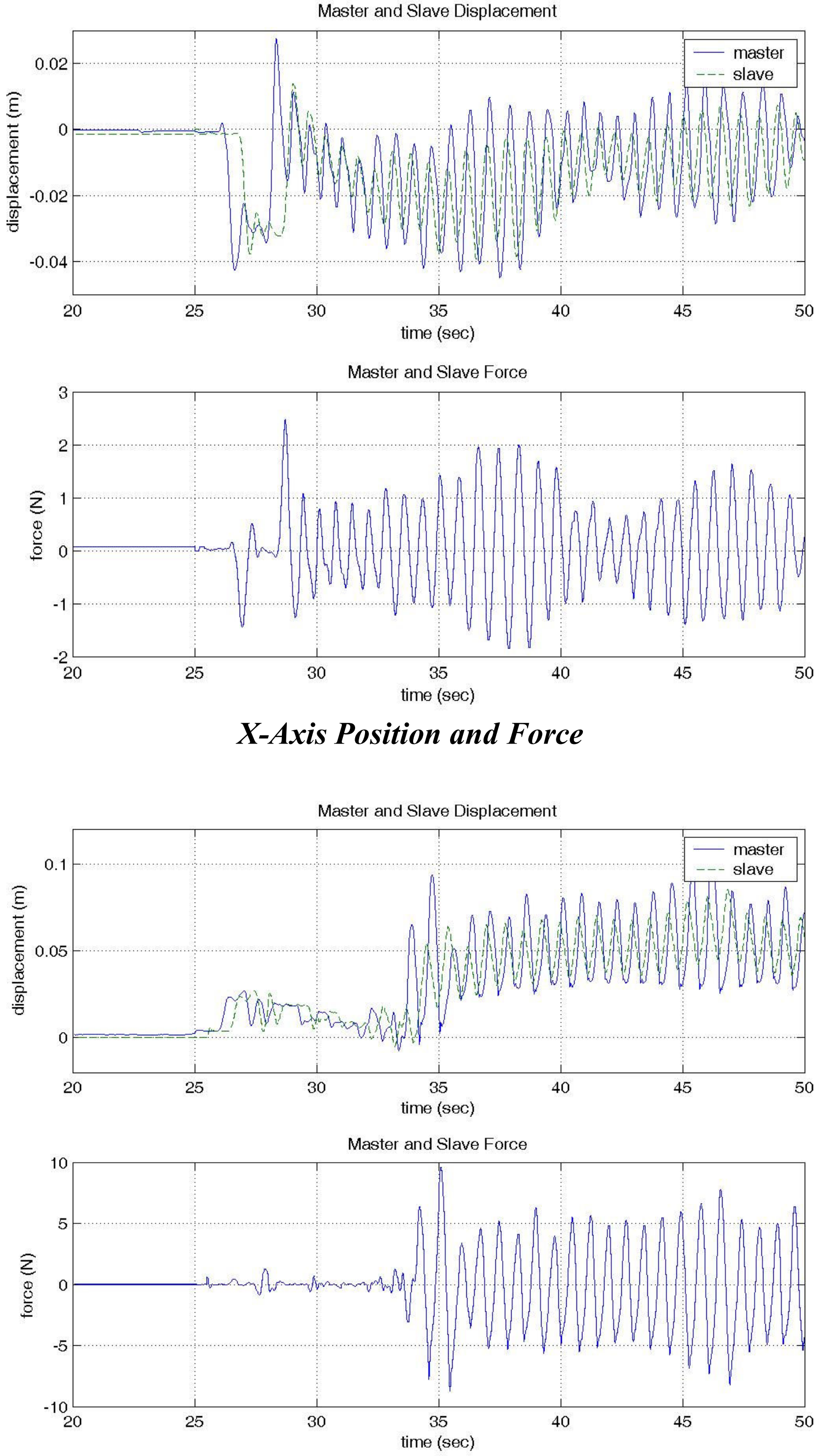


X-Axis Position and Force



Y-Axis Position and Force

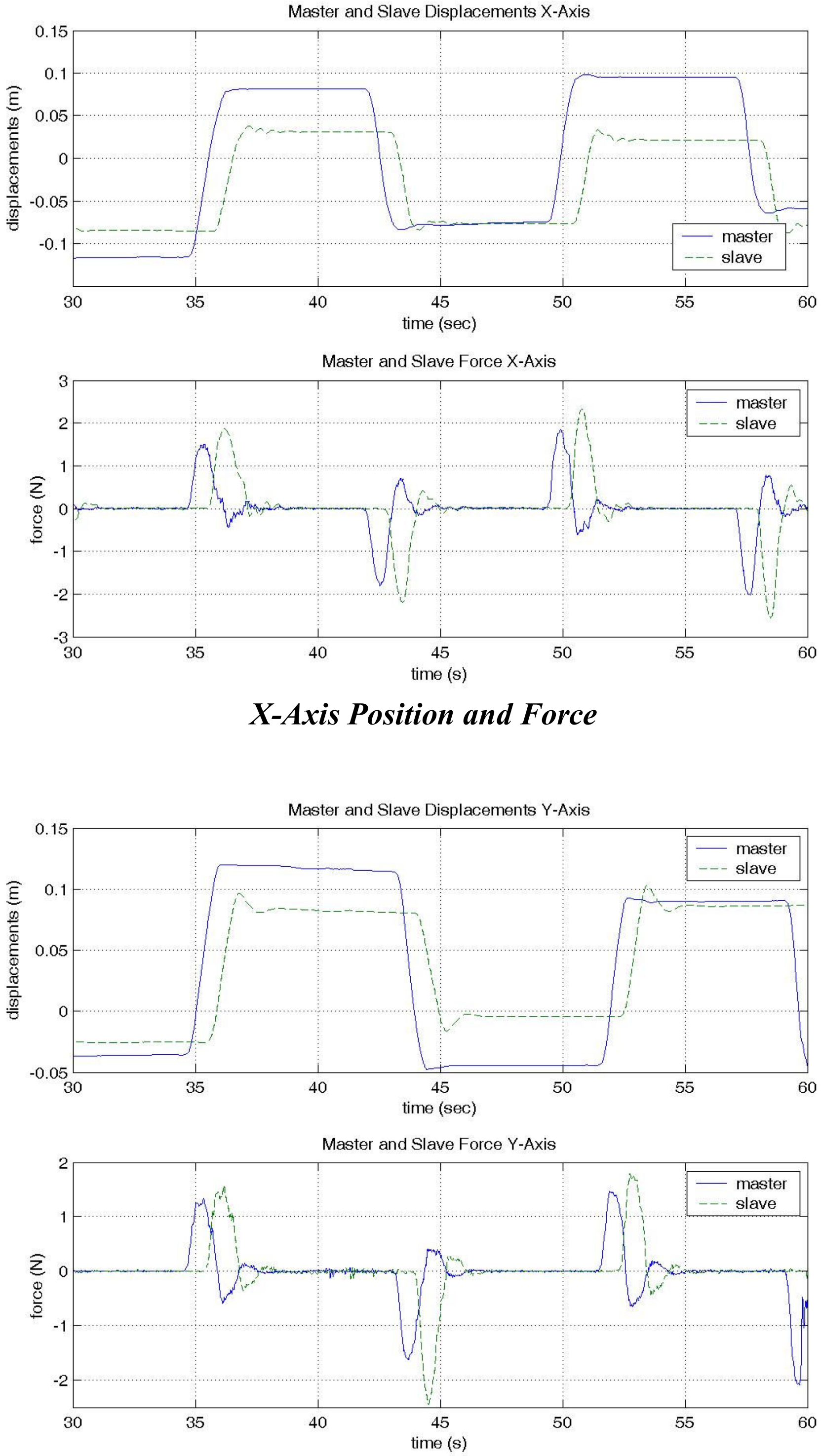
Figure 5.20: HAL Slave, Conventional Bilateral Teleoperation in Free Space without Delay



X-Axis Position and Force

Y-Axis Position and Force

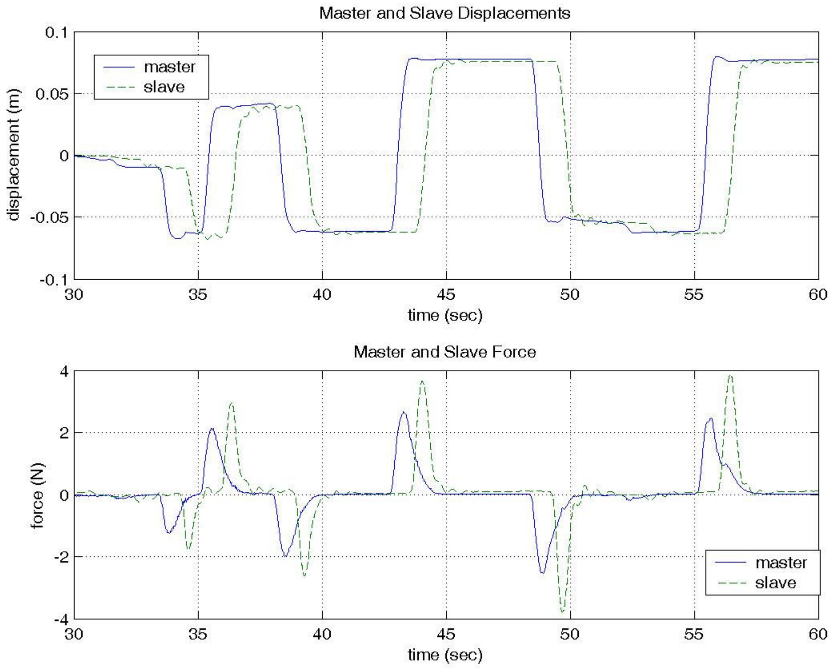
Figure 5.21: HAL Slave, Conventional Bilateral Teleoperation in Free Space with ~200 ms



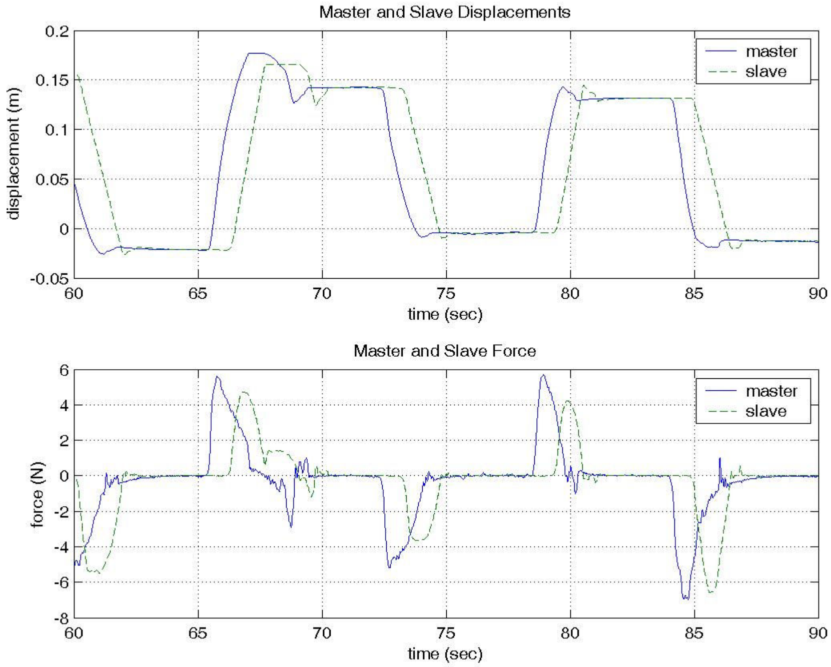
X-Axis Position and Force

Y-Axis Position and Force

Figure 5.22: HAL Slave, Wave Variable Teleoperation in Free Space with ~400 ms Delay



X-Axis Position and Force



Y-Axis Position and Force

Figure 5.23: HAL Slave, Wave Predictor with Drift Control in Free Space with ~400 ms Delay

machines. All position measurements are adjusted to PHANTOM scale for easier comparisons.

Figure 5.20 shows the result of conventional bilateral teleoperation without transmission delay with the operator trying to produce a step input. Notice the transient tracking error on the y-axis as HAL with the asymmetric speed limitation tries to catch up with the PHANTOM. This effect is also present in the x-axis but far less pronounced. Nevertheless there will be no steady state error since any position difference will produce a force on HAL causing it to eventually catch up with the master. With transmission delay of about 200 ms one way, the system is already exhibiting instability in Fig. 5.21. Again the human operator has his hand on the master to constrain the motion to prevent exponentially increasing oscillation in feedback force from damaging the machines.

Figure 5.22 shows the result of the wave variable teleoperation with delay which again shows that stability is guaranteed. However unlike the case with conventional bilateral teleoperation, there will be steady state error in position. This error can be minimized by adjusting the wave impedance to better match the slave-environment impedance. Nevertheless, the nonlinear characteristics of HAL also contribute to the steady state error. The velocity saturation prevents HAL from keeping up with PHANTOM in high speed. Significant asymmetric velocity limits in the y-axis can also produce a bias that gradually pushes HAL in one direction. Recall that since wave variable algorithm does not encode the position data, once error has been generated there is no way to recover.

Figure 5.23 shows the result of the wave predictor with drift control teleoperation with delay. The algorithm successfully stabilizes the system with little position drift because the drift control can eliminate steady state errors. The predictor model can easily incorporate the saturation limits, resulting in reasonably accurate predictions of force feedback as shown in the plots.

CHAPTER 6

HUMAN TESTING OF WAVE-BASED ALGORITHMS

Any form of teleoperation will ultimately require a human operator in the servo-loop. Even though we can learn extensively from experiments using the hardware and measure many performance parameters, we can never be quite sure how a human would perceive the performance characteristics just based on this information. For this reason, this thesis includes a study where average human users are asked to perform a number of simple tasks using the transmission delay-compensating algorithms. This allows us to evaluate the algorithms in a way that would be impossible simply by looking at plots.

Three algorithms are tested in this study: conventional bilateral teleoperation with no delay, wave variable only algorithm with 200 ms delay one way, and wave variable algorithm with adaptive predictor and drift control also with 200 ms delay one way. The adaptive algorithm used tries to estimate the environment contact rigidity as well as the location of contact. The use of conventional bilateral teleoperation algorithm serves two purposes: as a chance to allow the subjects to familiarize with the concept of haptic and as a baseline performance against which the other algorithms can be compared. It may seem unusual that we use conventional bilateral teleoperation without delay instead of with delay for comparison. Indeed the conventional algorithm without delay serves as an "ideal" teleoperation and is not meant to be compared directly with the other delayed cases. There are two reasons this algorithm is used without delay. Given the amount of transmission delay, a human operator can hardly accomplish any of the tasks assigned because he or she will be spending a tremendous effort just to stabilize the system. The same reason of instability also makes it very difficult for such algorithm to gain approval for use with actual human subjects due to the large risk involved.

The subjects are asked to perform three tasks that involve both free space and rigid contact environments. More importantly the tasks may involve transitioning between free space and rigid contact. The three tasks tested are free space trajectory tracking, surface contour identification, and maze navigation. Given three algorithms to be tested for each task, there will be a total of nine experiments per subject. The free space trajectory is primarily a free space test while the other two involve rigid contact as well. Over the course of the experiment, the subjects were asked to evaluate the effectiveness of the algorithms on a point system. At the end the subjects are asked to provide feedback in a more subjective fashion.

The subjects are recruited from the Georgia Tech student body across several majors and consist mostly of undergraduates as well as a few graduate students. This gives the age range between late teens to the late twenties. This group is selected for their relative inexperience in any haptic-based applications. The subjects are given a baseline compensation for participation as well as a bonus for good performance. The bonus is given as an incentive for best effort on part of the subjects.

6.1 General Setup and Protocol

This study uses the PHANTOM 1.0 as the master and PHANTOM 1.5 as the slave. The master and slave devices are laid side by side, although a blind can be put between them as necessary. The blind is necessary in some tasks where the subjects need to identify contour surfaces and navigate mazes using feedback force only. However, the proximity of the master and slave is useful during the training phase at the start of each experiment. Figure 6.1 shows the master side of the setup with the blind in place. The slave PHANTOM on the other side shown in Fig. 6.2 with a mount placed underneath it. This mount is used to place the contour surface blocks and mazes. The blocks and mazes are attached to the mount using Velcro due to its ease of attachment and removal. The force of the PHANTOM is not large enough to detach the Velcro from any direction.

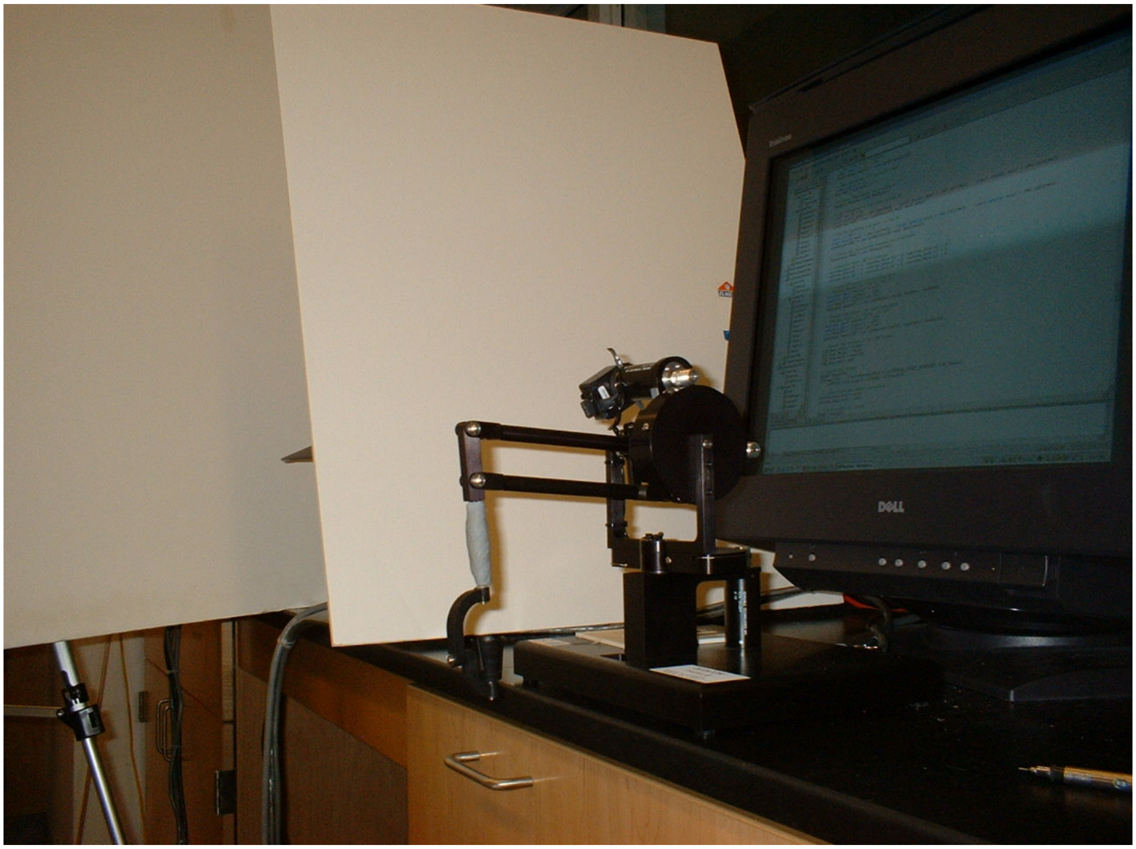


Figure 6.1: Human Experiment Setup Master Side

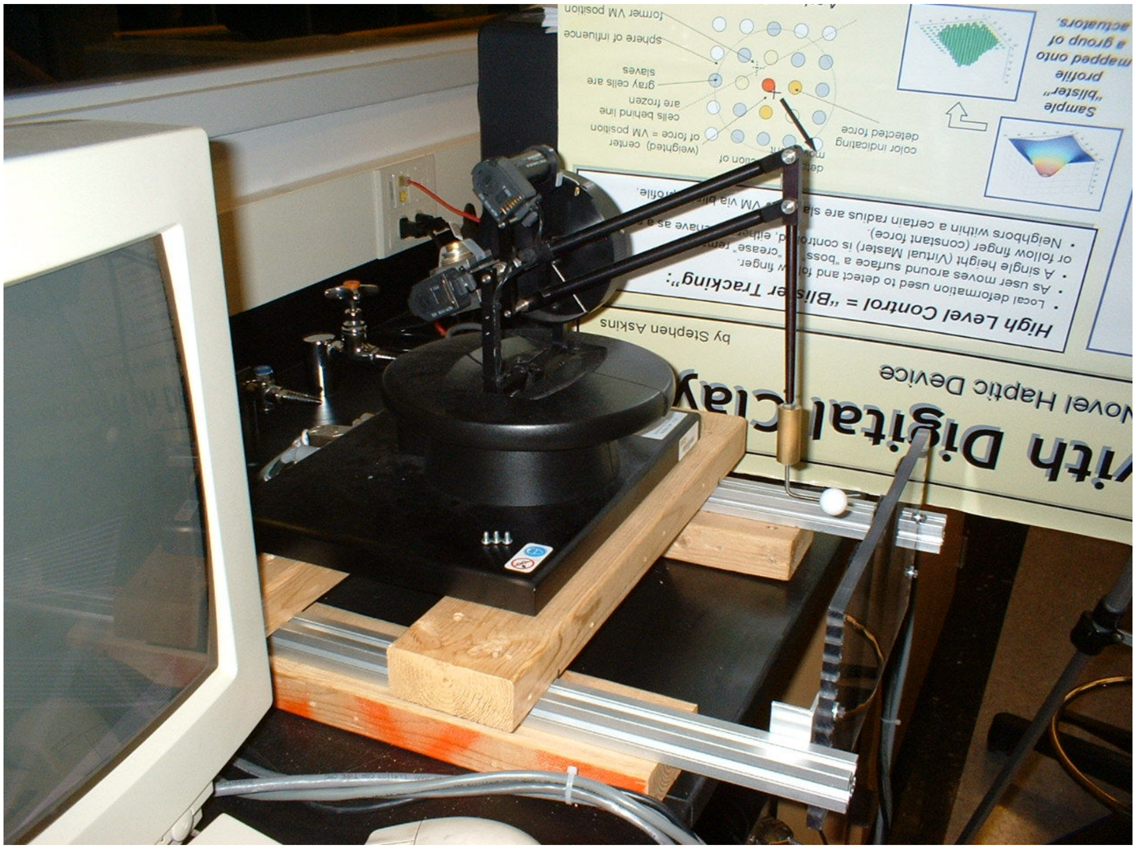


Figure 6.2: *Human Experiment Setup Slave Side with Mount for Surface Contour and Maze Blocks*

The slave end-effector is modified into Teflon sphere as shown in the Fig. 6.2. This allows clear contact point in all directions except the negative z-axis direction. Teflon is chosen as the end-effector material for its low friction coefficient with many other materials. The size of the Teflon sphere is 0.75 inch in diameter which is the approximate size of a human thumb.

The master and slave computers are connected via the Internet with simulated variable transmission delay as described in Chapter 5. However, the nominal one way delay time is set to be roughly 200 ms instead of 400 ms. Previous tests have shown that 400 ms could be too difficult for an average human operator, and besides 200 ms appear to be more realistic for Internet transmission delay. The subject on the master side has access to a computer monitor because some task may require visual aide from the display. He or she also has access to headphones from which noise would be played. The reason is to eliminate audio cues that may result when the slave PHANTOM knocks against the environment.

A total of 25 subjects are recruited by means of flyers and verbal advertisements. To encourage participation, an award system based on Barnes and Nobles gift card is set up to compensate the subjects. Each subject is given a US$10 gift certificate just for participation. However, to encourage best effort, a bonus system was also set up. There are a total of nine experiments in this study per subject. We have tested out all the experiments as researchers and are aware of approximately how well an average operator can perform. Based on the researchers' performance, we set up a benchmark score for each experiment for the subjects to aim. If the subject is able to perform better than the benchmark in six out of nine experiments, he or she is given an extra US$5 gift card. If the subject is able to outperform the benchmark in eight out of nine experiments, he or she is given an extra US$10 gift card. The performance of the subjects relative to the benchmark is not revealed to the subject until the very end. This is to insure that the subject would not adjust his or her effort based on assessment of previous performance.

Each subject is scheduled an appointment lasting two hours via e-mail, although the actual study rarely exceeds 90 minutes per subject. As soon as the subject arrive, the investigator would go over the consent form with him or her which covers the purpose of the study, an introduction to the hardware, a brief description of the experiments, compensation and confidentiality policies, and contact information. If the subject understands and agrees to the terms, he or she is asked to sign the forms. The subject also receives the baseline US$10 gift card at this point.

The investigator then proceeds with the experiments. The experiments are divided into three sets of free space trajectory tracking, surface contour identification, and maze navigation, and the subject performs the sets in this order. The details on these tasks will be described in a later section. For each set, conventional bilateral teleoperation with no delay, wave variable only algorithm with 200 to 300 ms (biased toward 200 ms) delay one way, and wave variable algorithm with adaptive predictor and drift control also with 200 to 300 ms (biased towards 200 ms) delay one way algorithms are tested. The subjects always start each set with the conventional bilateral teleoperation because it is the easiest and gives them a chance to familiarize with haptic applications. The later two algorithms may be performed in any order as part of the randomization process. At the beginning of each experiment, the subject is given a chance to practice with the algorithms to get used to them. After each set of experiments, the subject is asked to rate the wave variable and wave-adaptive predictor with drift control algorithms based on their preference on a scale of ten with the assumption that the conventional bilateral teleoperation without time delay gets perfect score (10/10) and a zero means the algorithm is completely useless. At the end of the study, the subjects are given a chance to provide feedback in a more informal and subjective manner. This gives them the opportunity to describe the haptic sensation in their own terms; this feedback is not possible with objective rating method. Finally the subject's performance relative to the researchers' benchmark is revealed, and the bonus is awarded accordingly.

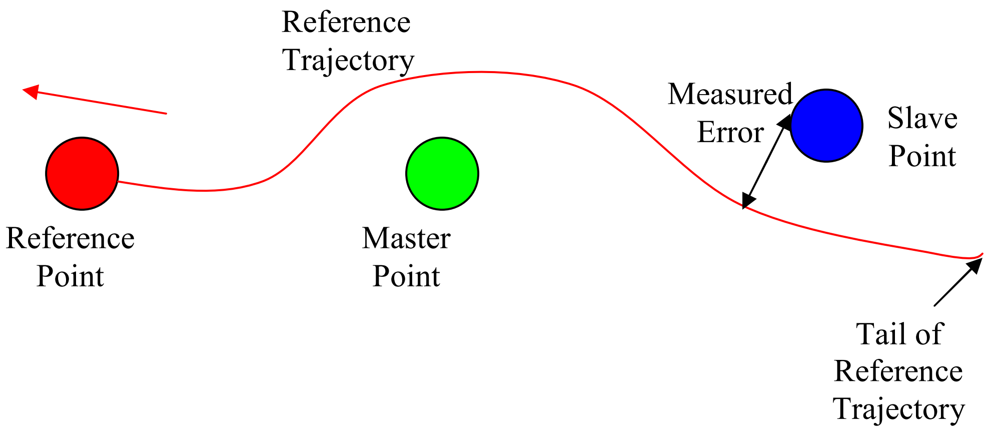


Figure 6.3: Free Space Trajectory Tracking Task

6.2 Tasks

This section describes in detail the tasks the subjects are asked to perform. The nature of the tasks, the necessary props, and the rules will be fully explained. Note that in all tasks the PHANTOM z-axis is never used. Hence the z-axis is essentially locked by providing a very stiff feedback force to resist any attempt from the user to displace it in this direction. The delay when present is always between 200 and 300 ms. Each data packet would have a 50% chance of being subjected to between 200 and 225 ms delay, 25% chance between 225 and 250 ms delay, 15% chance between 250 and 300 ms delay, and 10% chance of total loss of data packet. These delay probabilities can be equivalent if not worse than the characteristic delay of UDP transmission across the Atlantic Ocean.

6.2.1 Free Space Trajectory Tracking

The motive for this task is to evaluate the free space performance of the algorithms. The subject is shown a display of circles of different colors showing the master, slave, and reference positions. The displayed master position is the current, but the displayed slave position is based on the received data from the slave with the corresponding simulated transmission delay. The circle representing the reference position will move around the computer screen drawing a reference trajectory at a constant velocity. The task of the subject is to manipulate the PHANTOM such that the slave traces out the reference trajectory with minimum error as shown in Fig. 6.3. The error is measured as the shortest distance between the slave position and the reference trajectory, not the reference position. This means the operator can afford to let the reference point move ahead of the master and see the reference trajectory ahead before attempting to track it. However, we do not want the operator to spend a long time in tracking and getting nearly zero error in the process. Hence the reference trajectory is programmed to gradually disappear after appearing for three seconds as a way of pushing the operator along. The subject is advised not to lag behind the tail of the reference trajectory as then the error would be measured from the slave position to the trajectory tail. The performance index of this task is the average error between the slave position and reference trajectory.

In total there are four programmed reference trajectories: one for each algorithm as well as one for practice. The same one is used for practice all the time, but the others could be shuffled for the various algorithms. All the mazes are designed to closely match each other in terms of difficulty. The trajectories are assembled using equal numbers of four types of track segments: short and long straight tracks, large and small curvature semi-circular tracks that provide 90 degree turns. We can think of it like assembling a model railroad track from given track segments. This results in all the trajectories having equal track lengths. Connections between track segments that produce abrupt and sharp turns are also set to be equal in all trajectories.

6.2.2 Surface Contour Identification

The motive for this task is to evaluate the subject's ability to identify a contour surface on the slave side based only on haptic feedback. The contour surface is carved into a wooden block and placed on to the slave mount vertically using Velcro. The

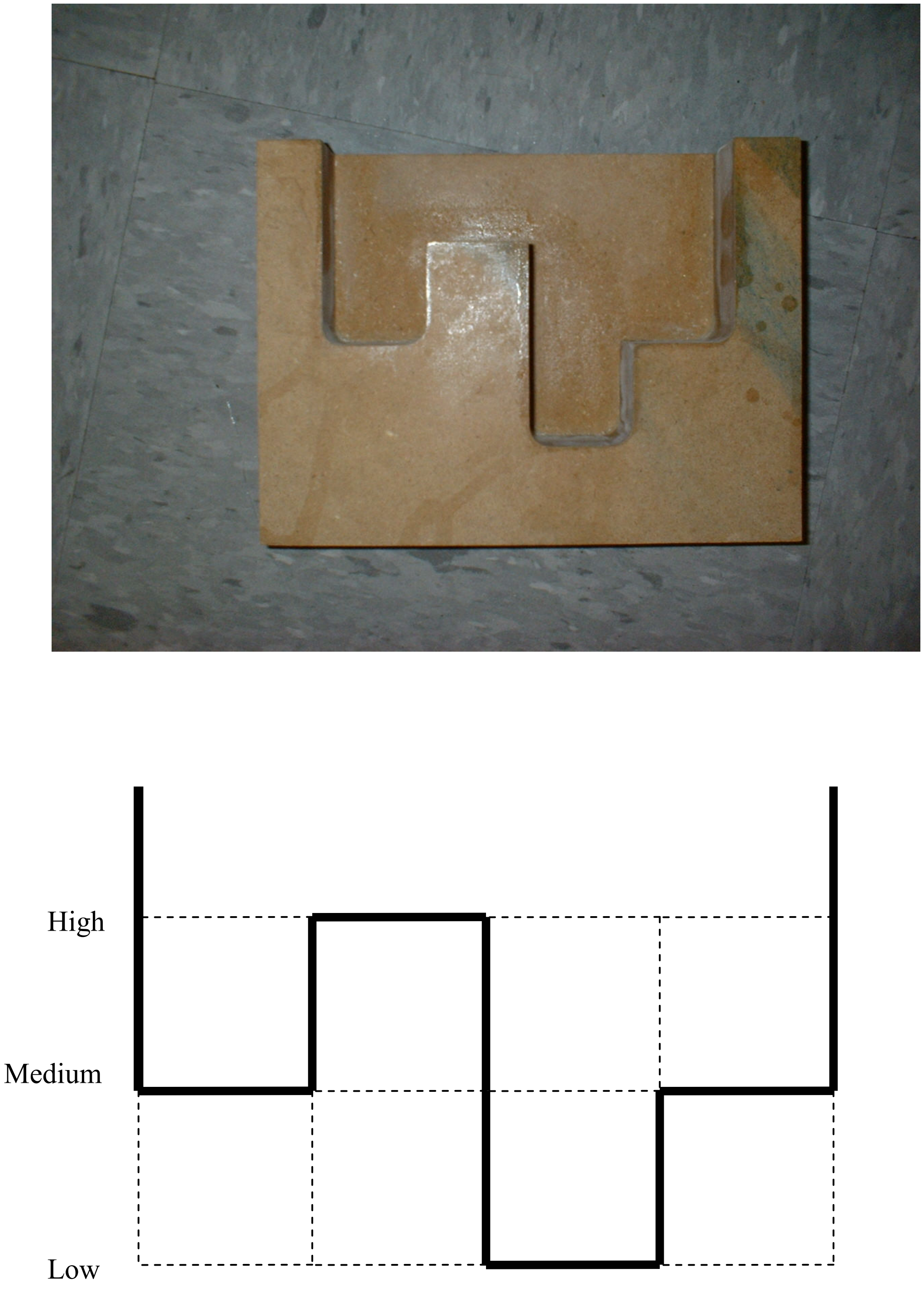


***Figure 6.4:* Contour Surface Block and Diagram**

subject is then asked to explore the shape of the contour using the master as much as he or she likes within a 30 second time limit. Then he or she is asked to identify the contour surface via multiple-choice with four possible choices for each block. For each algorithm the subject is required to identify four contour surfaces. A blind is placed between the master and slave throughout the experiment so that the subject is not able to identify the contour surface visually. Furthermore since the subject is required to listen to noise via headphones to prevent him or her from hearing the knocks resulting from collision between the slave end-effector and wooden block.

The shape of the contour surface is designed based on four partitions. Each partition may have a low, medium, or high level. Each block will have at least one high and one low level partition. Figure 6.4 shows how the partition system works as well as a picture of one of the blocks. Since there are three experiments with four contour identifications per experiment, there is a total of twelve blocks of different surfaces. The set of blocks used for each algorithm may be randomized for different subjects. A random block is selected for use as practice block for the training session.

Although the subject is not allowed to see the slave, he or she is allowed to look at the display on the master side. The display shows the position of the master as well as two lines indicating the border of the workspace to the left and right. This gives the subject an idea of where he or she is relative to the workspace borders without giving away the shape of the contour surface.

6.2.3 Maze Navigation

The motive for this task is to evaluate the subject's ability to navigate a maze based only on haptic feedback from contact between the slave end-effector and the maze walls. The maze is cut from acrylic glass and placed on the slave mount vertically using Velcro. The maze has a start and goal point, and the subject is asked to navigate his or her way from the former to the later point in as little time as possible. Again the blind

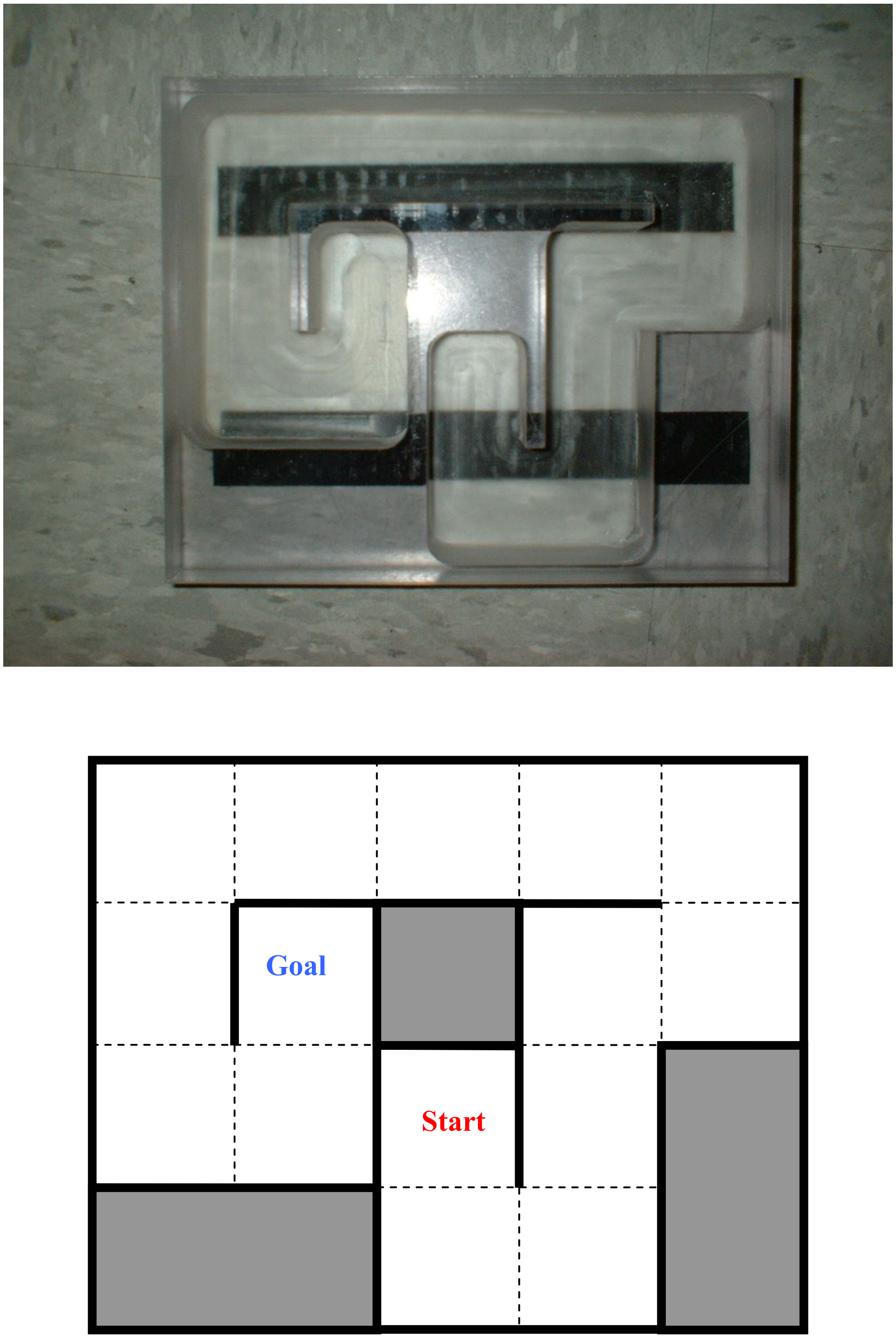


Figure 6.5: Maze Block Picture and Diagram with 8 Turns Covering 15 Partitions

between the master and slave and the headphone with noise are used so that the only feedback is haptic in nature.

The maze is designed based on a four-by-five partition matrix. Walls are place between the partitions to create a winding path. Figure 6.5 shows a picture of a maze and a diagram explaining the partition matrix design. The description *maze* may be a misnomer because there are no branches in the path, and there are no dead-ends since the purpose of this study does not involve deceiving the subject. There is one maze block for each algorithm to be tested and one for training purpose. Hence there are a total of four distinct mazes. Nevertheless like the trajectories in the tracking experiments, the mazes are designed such that they all have similar level of difficulty. All mazes are designed with the same number of turns covering equal distances.

Again the subject is allowed to view the display on the master side. The display shows the master position, start location, and goal location. This gives the subject a sense of direction of the goal point and also prevents him or her from unintentionally backtracking back to the start point.

6.3 Results

This section describes the results of the human testing. The results reflect the performance of the subjects in the three tasks using the three algorithms. They also reflect feedback from the subjects regarding their personal preferences of the algorithms. The subject feedback is divided into objective rating system and subjective opinions.

Besides performance data and feedback ratings, we also record the sex, handedness, and hours per week spent on video games of the subjects. However, we will not attempt to draw any conclusions from this information primarily because the sample is highly unbalanced. Our sample of 25 subjects contains only 5 females and 3 left-handed people. Only 6 subjects report to play over 5 hours of video games per week with 8 subjects report to play no video game at all.

6.3.1 Subject Performance

The subjects' performance is evaluated for all three tasks. In the trajectory tracking task, the performance index is the average error between the slave position and the reference trajectory. In the contour surface identification task, the performance index is how many of the four contour blocks for each algorithm does the subject get correct. In the maze navigation task, the performance index is how long the subjects take to reach the goal point from the start point. The results can be summarized by Table 6.1. In each entry, the mean, standard deviation, and and 95% confidence interval of all the subjects' performance are shown. The acronym *WAPD* stands for wave variables with adaptive predictor and drift control. The conventional signifies that no time delay is intentionally added.

Table 6.1: Subject Performance

	Mean	Std Dev	95% C.I.
Conventional – Tracking	1.49 mm	0.26 mm	± 0.11 mm
Wave Only – Tracking	3.63 mm	0.53 mm	± 0.22 mm
WAPD – Tracking	1.87 mm	0.52 mm	± 0.22 mm
Conventional – Contour ID	3.96	0.2	± 0.08
Wave Only – Contour ID	3.28	0.84	± 0.35
WAPD – Contour ID	3.6	0.87	± 0.36
Conventional – Maze	14.48 sec	5.24 sec	± 2.16 sec
Wave Only – Maze	42 sec	23.02 sec	± 9.72 sec
WAPD - Maze	21.28 sec	6.70 sec	± 2.77 sec

In the trajectory tracking task, it is apparent that the wave variable with adaptive predictor and drift control algorithm has performs better. The average error using wave variable algorithm alone with roughly 200 ms delay is almost 100% larger than the value using WAPD algorithm. Despite the delay in tracking, WAPD is able to trace the

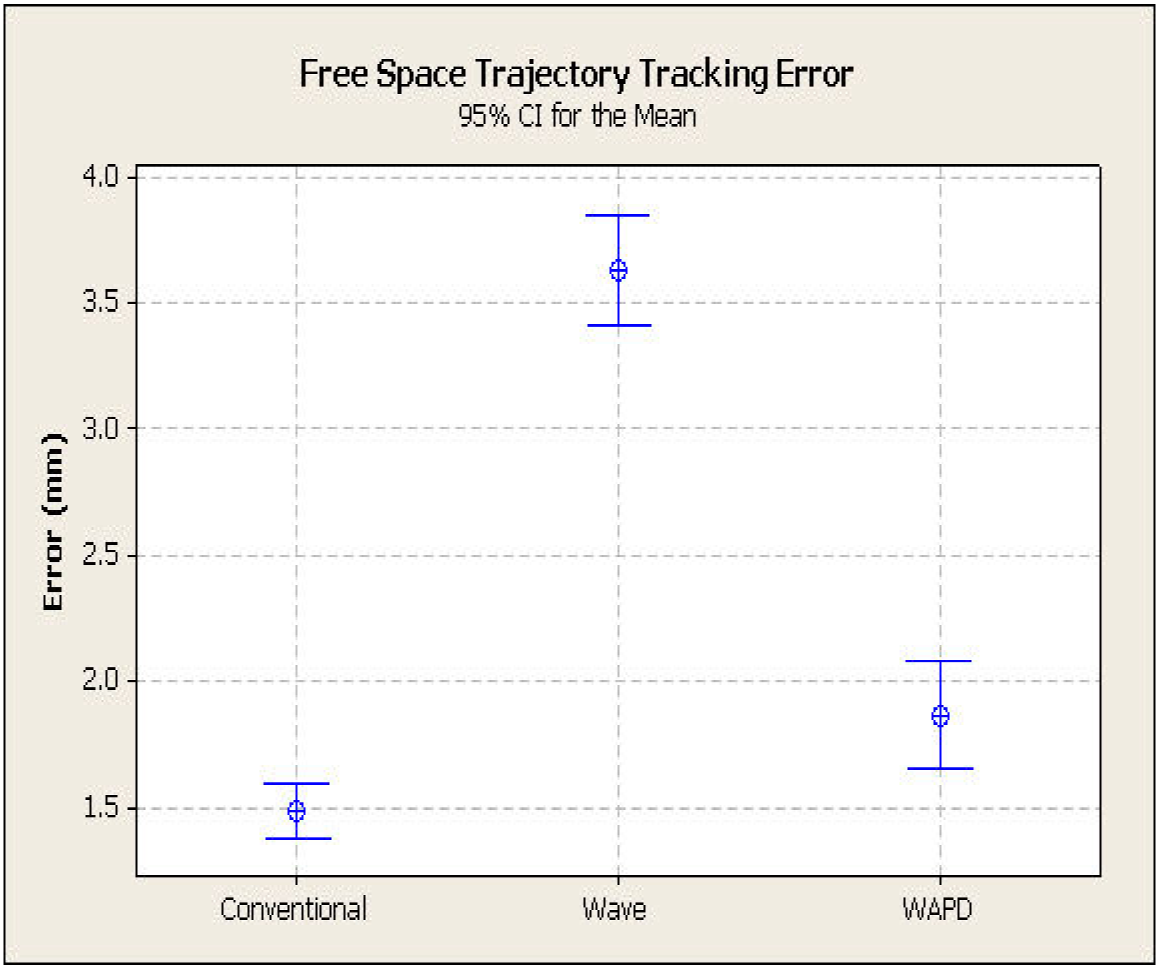


Figure 6.6: Free Space Trajectory Tracking Error

reference trajectory well, approaching the conventional case without delay in performance. Figure 6.6 shows the result graphically.

In the surface contour identification task, the improved wave based algorithm still appears to be superior compared with the wave variable only algorithm, but the difference is far less distinct. In fact the confidence interval of the wave variable algorithm overlaps with that of the WAPD, suggesting the difference may not be statistically significant. In retrospect, the task design does not appear to contain enough challenge with several subjects correctly identifying all the contour surfaces for all the algorithms. Testing the subjects to identify blocks in a multiple choice format has the benefit of higher objectivity, but it also gives the subject an opportunity to guess the correct answer. Subject can increase his or her chance of getting the correct answer by the process of elimination. Nevertheless, the data recorded from this task is still useful, especially the feedback ratings from the subjects to be discussed in the next section. Figure 6.7 shows the subject performance graphically.

In the maze navigation task, again the improved wave based algorithm appears to be superior compared with the wave variable only algorithm. Subjects using the wave variable only algorithm with roughly 200 ms delay take roughly twice as long to complete the maze as in WAPD case. It is interesting to note that the completion times with the wave variable only algorithm has a much higher variance. The investigator noticed that one of the reasons for this could be psychological. Some subjects show visible signs of frustration after one minute trying to finish the maze. This certainly hinders their ability to effectively accomplish the task. A few resorting to returning to the start point repeatedly even though they were told that the mazes do no have branches or dead-ends. One subject forfeited after five minutes. Her completion time is considered undefined and not included in the result in Table 6.1. Despite this, the completion time of the wave variable only algorithm is still longer than the WAPD with no overlap in

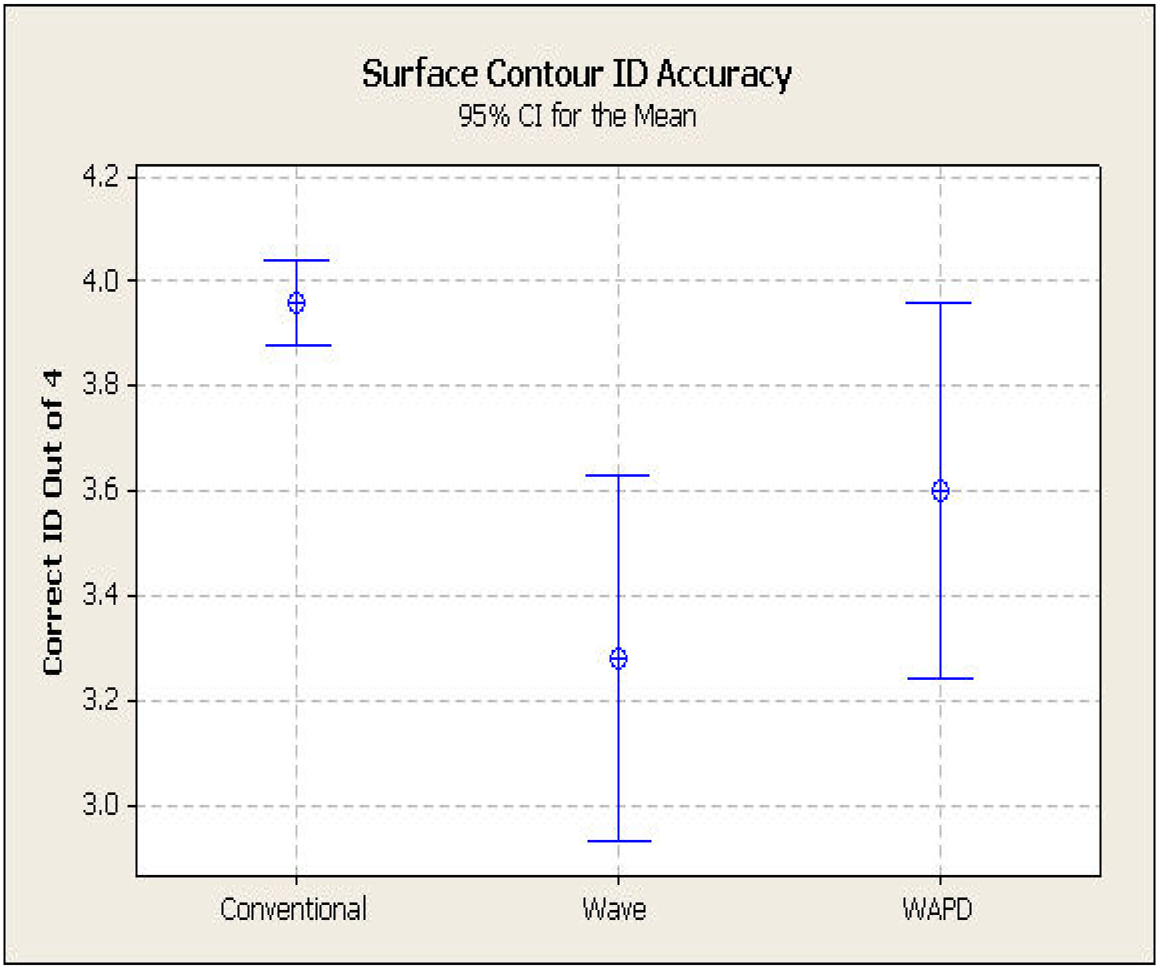


Figure 6.7:   Surface Contour ID Accuracy

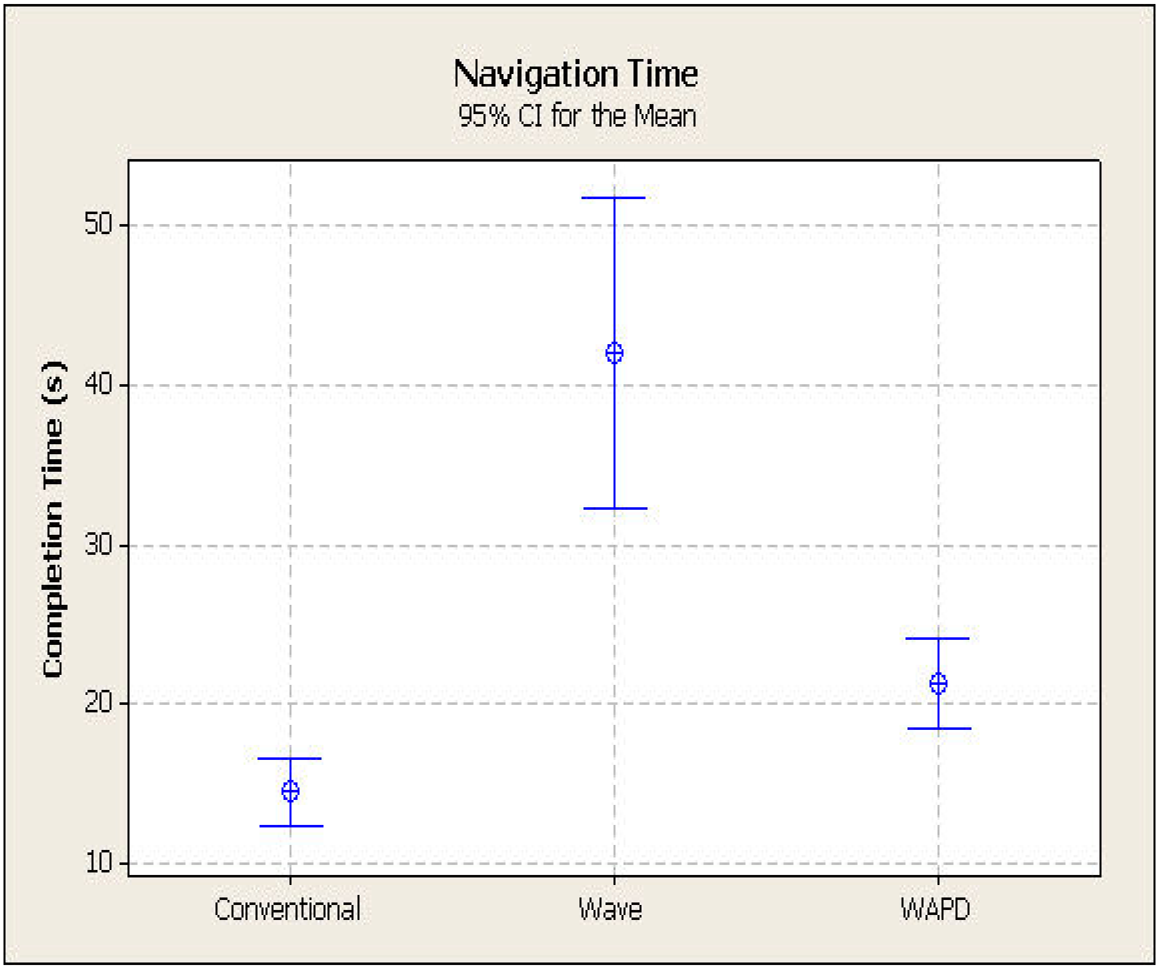


Figure 6.8: Maze Navigation Time

confidence interval, indicating the difference is still statistically significant. Figure 6.8 shows the results graphically.

6.3.2 Subject Feedback

After each task, the subjects are asked to give feedback on their preference between the wave variable only and WAPD using a rating system. The conventional algorithm is arbitrarily given ten points as a benchmark while zero point signifies complete uselessness. The results are shown in Table 6.2 and graphically in Fig. 6.9.

Table 6.2: Subject Feedback

	Mean	Std Dev	95% C.I.
Wave Only – Tracking	5.08	1.47	± 0.61
WAPD – Tracking	8.10	0.87	± 0.36
Wave Only – Contour ID	5.04	1.74	± 0.72
WAPD – Contour ID	6.48	1.45	± 0.60
Wave Only – Maze	5.24	1.33	± 0.55
WAPD – Maze	7.16	1.21	± 0.50

In all tasks, the WAPD algorithm scored higher than the wave variable only algorithm. This is especially true in the trajectory tracking task where the former scored over 60% higher than the later. Even with the other tasks, the former still scored higher than the later by about one standard deviation. Despite the fact the performance distinction is less pronounced with the surface contour identification task, the subjects still appear to prefer the modified wave based algorithm. In all cases, there is no overlap in confidence interval, suggesting statistically significant results.

After the study, the subjects are given a chance to provide feedback in a more subjective format to explain why they chose to rate the algorithms the way they did. Unlike the rating system, such feedback is difficult to assess in structured manner, but it provides greater flexible and can provide insights that can not be obtained otherwise.

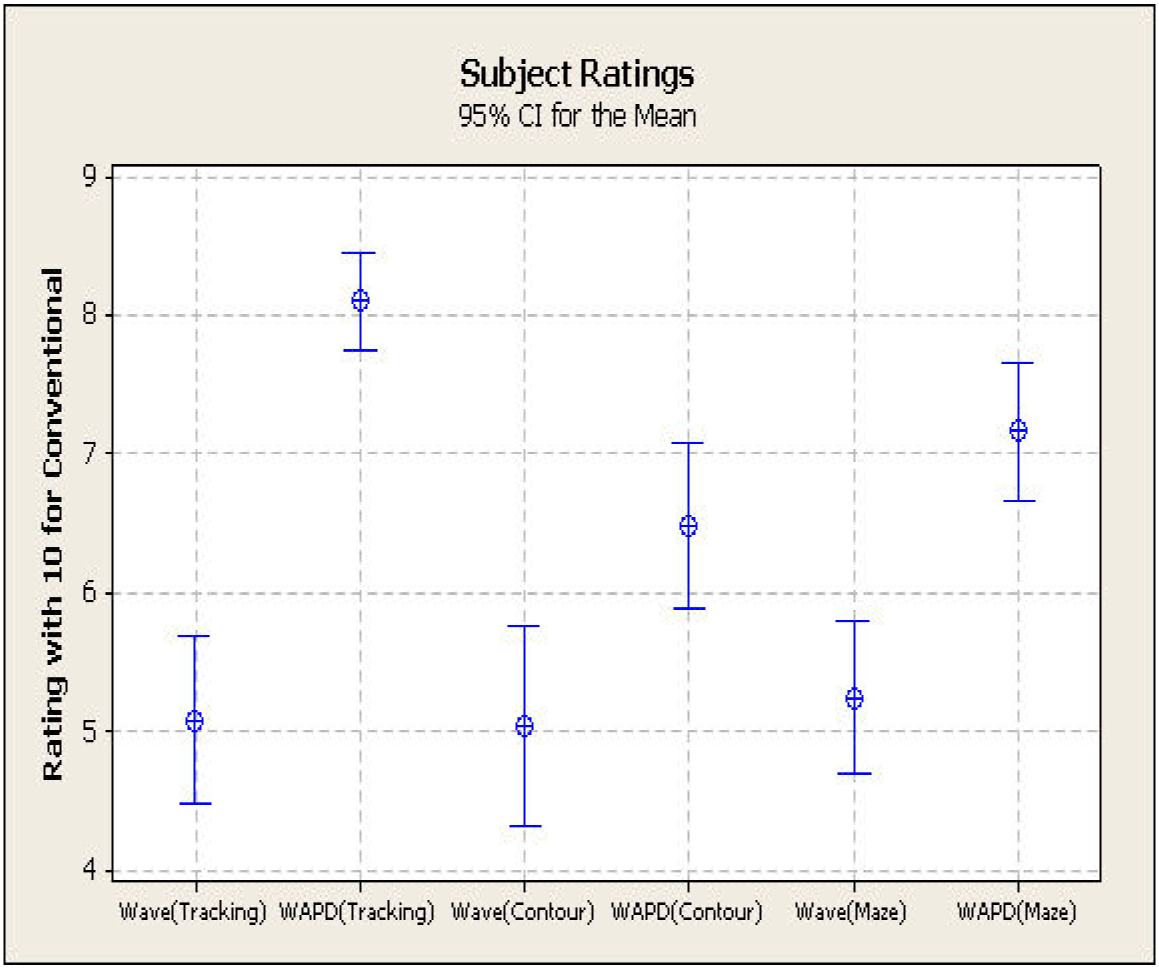


Figure 6.9: Subject Rating for Wave and WAPD for Different Tasks

The comments from the subjects can be summarized for each algorithm according to Table 6.3.

Table 6.3: Comments from Subjects

Wave Variables	WAPD
• Slave can not track master very well where it tends to cut corners during sharp turns. The slave trajectory also tends to be oscillatory. • The slave and master positions do not always agree even at steady state. Some operators find this annoying. • Contact force feels weak but gradual, making the obstacle feel soft (like jelly). For this reason, the operators are slower in detecting contact. • Oscillation in the contact force makes it difficult to identify the location of obstacle.	• Slave can track master well despite the delay. When the operator traces a path with the master, he or she can expect the slave to trace out a similar path after the round trip delay. • Contact force is more abrupt and hence provides a clearer and quicker indication of obstacle. Some however indicate this abruptness to come as a surprise. • Identification of location of obstacle is hindered by the initial delay when moving into contact. This distorts the perception of contact location. • The contact force may feel violent in pushing the master back into free space. Some finds this uncomfortable as it is too hard on the user.

In the trajectory task, the subjects unanimously agree that the slave in wave variable algorithm has much difficulty tracking the master. Some find the steady state error in position to be a problem. Even though this error is relatively minor, the discrepancy between master and slave still shows up on the display. This is expected because the wave variable only directly encodes velocity and force data for transmission. No one find tracking to be an issue with WAPD. Many do find the delay in motion between the master and slave to feel unusual; however they noticed that the slave ultimately follows the path of the master. Steady state error is virtually nonexistent in

WAPD with the help of the drift control algorithm. As a result, all subjects have a strong preference for the later than the former algorithm in this task.

In surface contour identification and maze navigation tasks where transition characteristic between free space and rigid contact comes into play, the subjects still mostly prefer the WAPD over wave variable algorithm but to a lesser degree. Many subjects complain about the very soft contact force with wave variable algorithm. This is one main reason why they are unable to quickly detect contact via the master. While it is possible to tune to wave variable algorithm to provide harder contact force by increasing the wave impedance, this may not be desirable or feasible for various reasons. Increasing wave impedance would also produce a very viscous sensation when the operator is moving in free space. The wave impedance value used in this study is 8 N/(m/s) based on a balance between free space and rigid contact performance. However, individual operators may have different preferences on this balance. Another reason large wave impedance may be undesirable is the transient oscillation when under contact. Figure 5.13 and 5.14 from the previous chapter show that the larger the wave impedance, the more distinct the oscillations in force upon contact. Several operators felt the oscillation in force at contact, and this negatively affect their ability to identify contact location due to distortion in force feedback. Furthermore, in practice there is a limit on how small or large the wave impedance can be set due to velocity noise.

The subjects also perceive the contact transition to be more gradual with wave variable algorithm. When moving in free space, the wave variable computation of master feedback force contains a term that is directly proportional to master velocity. In the operator moves into contact, this produces an initial damping effect. When the feedback force encoded in the wave variable reaches the master, this initial "damping" force reduces the abruptness of the impact. Some subjects find this characteristic to be a good quality because impact does not come as a surprise. Others find this to contribute to sluggishness of the algorithm.

When it comes to contact, WAPD has certain qualities that can be interpreted by different individuals as strengths and weaknesses. In WAPD, the subjects feel no distortion in feedback force due to oscillation. Many feel the contact force feedback to be more abrupt. When the operator pushes into a contact, the predictor will still use free space model until the detection signal arrives from the slave after the transmission delay. By then the master may have pushed deep into the contact surface resulting in large contact force due to the instant switching in the predictor model. Despite the first order filtering effect of the regulator, some subjects find this sudden jump in contact force to come as a surprise and may catch them off guard. However, there are also many subjects who interpret this abruptness as a clear indication of contact and allows for quick contact detection.

The delay in switching of predictor model from free space to rigid contact also distorts to force feedback symmetry. When pushing into contact, this lag produces an illusion to the operator that the contact surface is located deeper than it should have been. However, as the operator moves back out of contact, the predictor remembers the contact location and dissipates the force at the correct point. This generates an asymmetric sensation in contact perception where detection point when moving into contact is deeper than that when moving out of contact. Some subjects see this as a contributor in the difficulty in surface contour identification where contact detection accuracy is critical.

The WAPD algorithm also causes some issues when it comes out of contact back into free space. Some subjects feel the sudden jump in force feedback from the initial contact as if the master is trying to throw the operator back into free space. While it is desirable for the operator to detect contact as soon as possible with clear indication of force feedback in contact, we do not wish the operator to feel overpowered by the system. In fact the predictor has been tuned with enough damping effect under contact such that even when the operator releases the master, the contact force feedback will not throw the master more than a few millimeters back into free space. Nevertheless, some subjects feel that the push from the master under contact to be too violent.

Based on the user rating, subjective feedback, and performance index, we can conclude that most favor WAPD over wave variable method only in all tasks, although in individual preference also plays an important role. While wave variable and WAPD algorithms do have their share of complaints particularly regarding rigid contact performance, most can be attributed to individual preference. Many of the distinctive characteristics of wave variable and WAPD algorithms can be both an aid and a hindrance to teleoperation. For example, the fast contact transition of WAPD algorithm can be interpreted as beneficial in providing quick contact detection for some, but also detrimental as an unpleasant surprise and abrupt jump in contact force for others. Individual algorithms can be tuned to fit different individual operators. Nevertheless, the WAPD algorithm offers more adaptability and flexibility when it comes to tuning for the user.

CHAPTER 7

CONCLUSION

This research has focused on bilateral teleoperation with significant transmission delay. The wave variable algorithm serves as the basis that can guarantee stability under any transmission delays. Other algorithms are added to enhance performance: predictor for transient performance, drift control for steady state error, and adaptive algorithm for robustness to environment. The effectiveness of the algorithm is evaluated via the use of human experiment. Human subjects with no background in teleoperation are asked to perform three simple tasks under one way transmission delay of above 200 ms. The tasks are free space trajectory tracking, surface contour identification, and maze navigation. The tests show the effectiveness of the augmented wave variable algorithm over the simple wave variable algorithm.

7.1 Summary

The thesis began with an introduction of bilateral teleoperation with transmission delay. Teleoperation can be interpreted as remote control over large distance or scale. It is a useful method to manipulate an environment where a human can not be physically present due to distance or hazard. Autonomous robot is a feasible alternative, but current technology in artificial intelligence may not be robust enough to cope with any environment. Teleoperation has the advantage of keeping a human in the control loop either in a supervisory or direct role. The use of haptics or force feedback in addition to visual or audio feedback in teleoperation can greatly improve telepresence and enhance teleoperation task performance. However in many instances there is significant latency in communication between the human and remote sites. Latency is essentially pure time delay within a closed control loop which can cause instability. Furthermore the latency may vary over time, especially over the Internet with TCP showing more variations than UDP.

Chapter 2 provides the audience with background research in teleoperation with time delay. A simple solution is the supervisory control where the human operator sends instruction to the remote robot. The remote robot then performs its task via a local control loop and sends the data back to the operator. The operator never becomes part of the control; hence stability is not a problem. The wave variable algorithm allows the operator to be directly involved in the teleoperation with guaranteed stability under any transmission delay. However the algorithm has significant transient oscillation and steady state error in position. Various researchers have proposed augmented wave variables to compensate for the disadvantages such as wave predictor, wave variables with drift control, and wave impedance matching. Smith predictor is another algorithm that can effectively compensate for time delay but is heavily dependent on accurate slave model. Many researchers have worked on incorporating an adaptive algorithm into the predictor. A virtual slave can also provide haptic feedback to the operator without the need of direct force feedback information from the remote site. Because the virtual slave is on the master site, the haptic feedback is not affected by transmission delay. Other time delay compensation algorithms include PD-type control with Llewellyn stability criterion, shared compliant control, passivity observer and controller, sliding rule, and H-infinity and μ-synthesis. In recent years, Handshake VR Inc. produced one of the first commercially available toolkit called TiDeC that is specially designed to compensate for transmission delay in teleoperation.

Chapter 3 provides the background research in wave variables. The wave variable algorithm is inspired by the telegrapher's equations which models after the power transmission line as a cascade of inductors in series and capacitors in parallel. Because the model consists of only energy storage elements, the resultant system is passive and lossless. The wave variable algorithm is derived by performing an eigen decomposition on the telegrapher's equation such that the pure delay terms are isolated. The wave impedance term is defined as the effective impedance of the power transmission line. However unless the impedance at the terminal exactly matches the wave impedance, the transmitted signal will be reflected which contributes to the poor transient response. Stability of the wave variable algorithm can be rigorously proven using scattering theory and energy balance.

Chapter 4 explains the steps taken to compensate for the shortcomings of the wave variable algorithm while taking advantage of its guaranteed stability. First a predictor is added that tries to anticipate what the returning signal will be without the delay. This enhances the transient response, but without a good predictor model to make accurate predictions, the errors accumulate into steady state errors in position. A drift control algorithm is added that can compensate for drift from any source, including predictor modeling error and time varying transmission delay. Another problem with this setup is the lack of adaptability to changing remote environment. An adaptive algorithm is added that can change the predictor model based on changes in the remote site. This modification is tested under free space and rigid contact environments. The semi-adaptive predictor algorithm is designed to estimate the location of the rigid contact when there is one based on slave input force and output position. If rigid contact is detected, the predictor model slave position is locked. As the slave moves into contact, the master contact force does not lead the slave force because there is no a priori knowledge of the contact location. But as the slave moves out of contact, the master force leads the slave force because now the contact location is known, and the predictor can anticipate when the slave will leave contact. The full adaptive predictor is also developed to estimate the environmental force. While this method is continuous with infinite modes, the master force never leads the slave force. Another full adaptive predictor where the contact surface spring constant and contact location are estimated provides a performance very similar to the semi-adaptive predictor case.

Chapter 5 shows experimental results of wave variable algorithm and the modified wave variable algorithms. The PHANTOM 1.0 Premium is used as master, and PHANTOM 1.5 Premium is used as slave. Transmission medium is the Internet using UDP, but the transmission delay is simulated as a stratified time varying delay of about 400 ms one way. The results show the stability and poor transient of the wave variable algorithm. The wave predictor algorithm shows steady state error in position which is driven to nearly zero by the drift control. The algorithms are also tested with rigid contact. The wave variable algorithm shows low contact force which can be improved by increasing the wave impedance at the cost of increasing transient oscillation. The wave variable algorithms with semi-adaptive and full adaptive predictors show their ability to provide the correct haptic feedback based on the changing slave environment. The wave based algorithms are also tested using PHANTOM 1.5 Premium as master and HAL as slave that has a workspace five times larger than that of the PHANTOM. HAL has several problems such as limited velocity due to small hydraulic valve opening and asymmetric force output in the positive and negative direction due to different piston areas. This poses a problem for the wave variable algorithm that shows severe position drift, but the drift control is able to compensate for this source of steady state error.

Chapter 6 shows the human experiment results. In this experiment, 25 human subjects are asked to perform three simple tasks using three algorithms. In the free space trajectory tracking task, the subjects attempt to trace a reference trajectory traced out on the display with minimum error. In the surface contour identification task, the subjects attempt to identify the contour shape placed on the slave side based on haptic feedback alone. In the maze navigation task, the subjects attempt to navigation a maze on the slave side based on haptic feedback alone in minimum time. The three algorithms used in each task are conventional teleoperation with no delay, wave variable only algorithm with over 200 ms delay one way and WAPD algorithm with over 200 ms delay one way. The results show the superiority of the proposed WAPD algorithm over the wave variable only algorithm in all tasks, although the performance between these two is less discernable in surface contour identification task. The subjects also give a higher rating for the WAPD algorithm. Subject feedback suggests that the WAPD provides a clearer indication of contact, but its abruptness is disliked by some operators. The wave variable algorithm is mostly seen as inaccurate in tracking and sluggish in providing haptic feedback. However its gradual force buildup is seen as a plus by some subjects.

7.2 Contributions

The major contributions of this thesis are the following to the best of our knowledge: the first implementation of wave variable based algorithm in a commercially available multiple degree of freedom hardware (PHANTOM) as master and slave, the first implementation of wave variable based algorithm involving a hydraulic slave, the development of a robust drift control algorithm that can compensate for any source of drift, the development of an adaptive predictor that operates in wave variable domain, and the comprehensive testing of the wave variable based algorithms involving real human subjects.

7.2.1 Implementation of Wave Variables

Wave variables have existed for over a decade, but many researches into this algorithm have involved simulation. The few implementation attempts have used custom built haptic devices with one degree of freedom. We believe that this thesis consists of the first attempt to implement wave variable based algorithms using commercially available three-degree-of-freedom haptic devices. Namely the PHANTOM haptic devices are used. Furthermore we believe this is the first implementation using real master and slave in wave predictor type algorithm. This is a significant step in wave variable research because it opens up the possibility of future collaboration with other institutions that also possess the PHANTOM.

The wave variable based algorithm has also implemented using PHANTOM as master and HAL as slave. We believe this is the first implementation of a wave variable based algorithm in such an asymmetric teleoperation setup involving an electromechanical master and a hydraulic slave. The scaling of velocity and force in wave variable teleoperation is put into practice in this implementation.

7.2.2 Drift Control Algorithm in Wave Predictor

The wave predictor algorithm has been proposed and tested earlier using partial simulation with a virtual slave in Munir and Book [11]. However upon implementation using a real slave, flaws in the algorithm were discovered. One of the most significant problems is that poor predictor model accuracy negatively affects performance. Although the system is still robust enough to guarantee stability, the prediction error accumulates over time to generate steady state error in master and slave position. Munir and Book also included a drift control algorithm. However it was discovered that while the algorithm is effective against drift caused by variations in transmission delay, it is ineffective against drift caused by poor predictor modeling. The reason is that their drift control is based on the expected difference between master and slave positions. However poor predictor model accuracy produces the wrong expected difference.

An alternative drift control scheme is used where the adjustment is made after transmission on the slave side. The master and slave desired positions are directly compared and their difference is used to compute a correction input injected into the wave variable flow. Passivity is still guaranteed by a regulator that keeps track of the net energy flow into the slave. Correction input is gradually stopped if it is found that the slave has begun to generate energy. Because it is not based on any expected value but on direct comparison between actual master and slave desired positions, this drift control algorithm is more robust in that it can compensate for any source of drift including time varying transmission delay, modeling error, and slave actuator saturation. We believe that it is the first time this type of drift control algorithm has been used in wave variable teleoperation. An original proof of the effectiveness of the algorithm is given along with an original derivation of the conditions under which the drift control will function.

7.2.3 Adaptive Predictor in Wave Variables

It is found that even with drift control algorithm, the wave predictor still lacks the ability to adapt to changing environments. Hence an adaptive algorithm is proposed that can change the predictor model as necessary based on the changing environment. A semi-adaptive predictor with mode switching algorithm and a full adaptive predictor that estimate slave parameters using a RLS algorithm have both been tested. While adaptive algorithms such as RLS and neural network have been used in predictor based teleoperation, this is the first time this method has been implemented with the wave variable algorithm. This combines the robustness and adaptability of the adaptive predictor with the stability guarantee of the wave variable algorithm. The combination has been tested on free space and rigid contact environment. It is found that in all forms of adaptive algorithms, the master feedback force does a good job tracking the slave contact force. Special attention is paid to the transition between free space and rigid contact and whether the master feedback force leads the slave feedback force. The distinctiveness in transition performance among the adaptive predictor algorithms are first noted in this thesis.

7.2.4 Human Experiment in Wave Variable Based Teleoperation

To better evaluate the effectiveness of the proposed WAPD algorithm compared to wave variable algorithm, 25 human subjects are recruited to participate in teleoperation tasks. The tasks involve free space trajectory tracking, surface contour identification, and maze navigation. It must be noted that this is one of the most comprehensive studies involving inexperienced human subjects in any time delay teleoperation in terms of the variety of tasks and the number of subjects. It is certainly the most comprehensive human subject study involving time delay teleoperation using wave variable based algorithms. For the first time the opinions of common laymen are recorded regarding the haptic feedback of wave variable and WAPD algorithms. This information is invaluable for the algorithms to be modified for use by the general public. Subjective user feedback can also provide insights that are not normally available simply by looking at master and slave position and force plots.

7.3 Directions for Future Work

The WAPD algorithm has been proven effective in time delay teleoperation under free space and rigid contact environments. These constitute the two most common environments that will encountered during teleoperation as well as two environmental extremes with free space having maximum admittance and rigid contact having maximum impedance. Of course the environment model can be far more complicated than these two extreme. For example, in surgery body tissue can often exhibit soft contact. The ability of the surgeon to distinguish among the various degrees of contact softness via the haptic device in teleoperation is vital to a successful operation. Another example will be free space motion but in a more viscous environment such as water or even glycerin. The adaptive predictor can be tested under such circumstance and modified where necessary to prove its robustness under a wider range of slave environment. Another form of adaptive algorithm, such as the neural network instead of RLS, can also be tested with the wave variable method. The result can prove particularly fruitful since the neural network is not limited by a linear slave environment.

Currently the slave PHANTOM does not include a force or even a contact sensor. While the adaptive algorithm is still able to operate based on the relationship between the control force input and position output, a force sensor will provide a more direct measurement of contact force and detection and decrease the possibility of false detection. Such sensor is considered fair since most past research involving teleoperation includes force sensor on the slave, and it does not affect the transmission delay in any way. In case of the PHANTOM, a multi-directional force sensor may be necessary for the teleoperation to work on the tasks prescribed in the human experiments. Packing this sensor into something the size of the PHANTOM end-effector may be challenging, but it will be useful for teleoperation with rigid contact.

Another area of future work is the application of wave variable based algorithm on HAL. Although the algorithm has been tested using PHANTOM as master and HAL as slave, the experiments only consist of free space environment. Rigid contact environment has not been tested yet due to the lack of effective force sensor. HAL is a hydraulic actuator that is effectively velocity controlled because the control signal opens the servovalve that controls flow rate and hence velocity. Unlike the PHANTOM, the control signal can not be used in place of environmental force. In this case direct force measurement is necessary from the environment. With force sensors functional on HAL, we can perform more thorough teleoperation experiments on HAL in the future.

Finally with the wave variable based teleoperation functional with a commercially available haptic device, we can request for collaboration with other institutions with similar devices. We only need to send the other researcher a copy of the code and exchange IP address and port numbers to allow collaboration. Testing the algorithms with different locations of various distances is a good way to prove their effectiveness. It also gives other researchers a chance to improve upon the current WAPD and foster the exchange of ideas in the field of teleoperation.

APPENDIX A

REFERENCE TRAJECTORIES

Figures A.1 to A.4 lists the reference trajectories used for the free space trajectory tracking test. The first trajectory is used for training purpose while the rest of actual test trajectories. The three test trajectories have similar difficulties in that they have same number of large curves, small curves, straight lines, and sharp 90 degree turns. The workspace is divided into a 3-by-4 grid, each grid being 5 cm by 5 cm. The reference point moves along at a constant 5 cm/s. The trajectories shown below are divided into four or five sub-trajectories each starting at the red circle and ending at the green circle. Where the start and end points are the same, red circle with green outline is used.

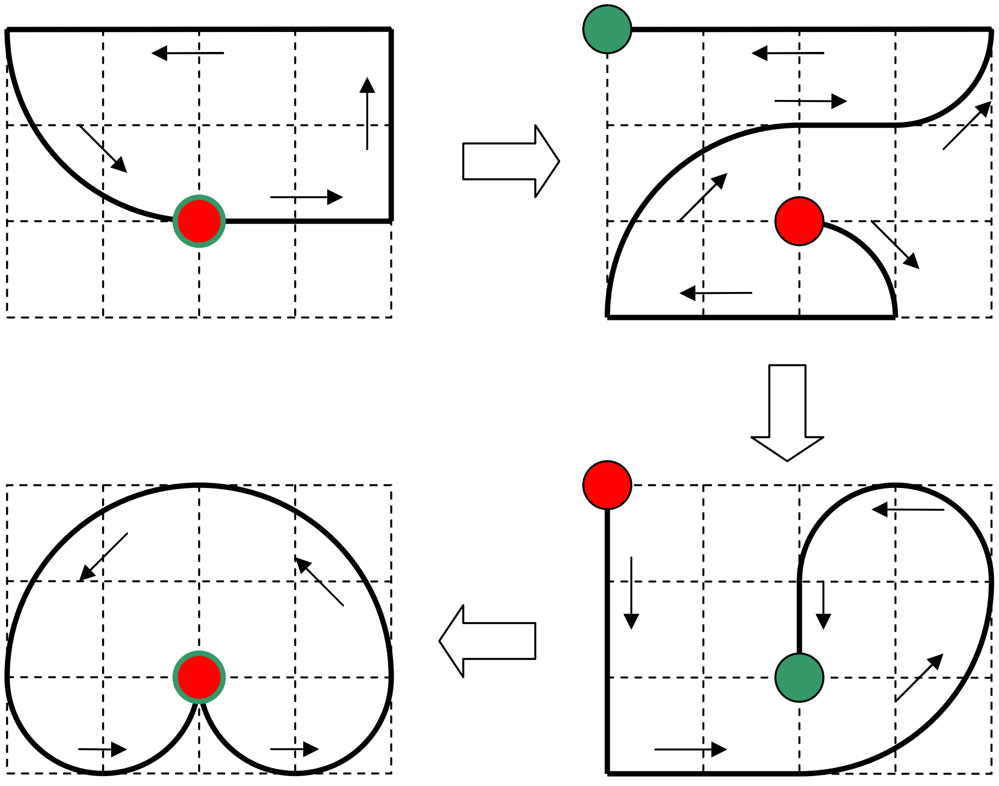


Figure A.1: Training Reference Trajectory

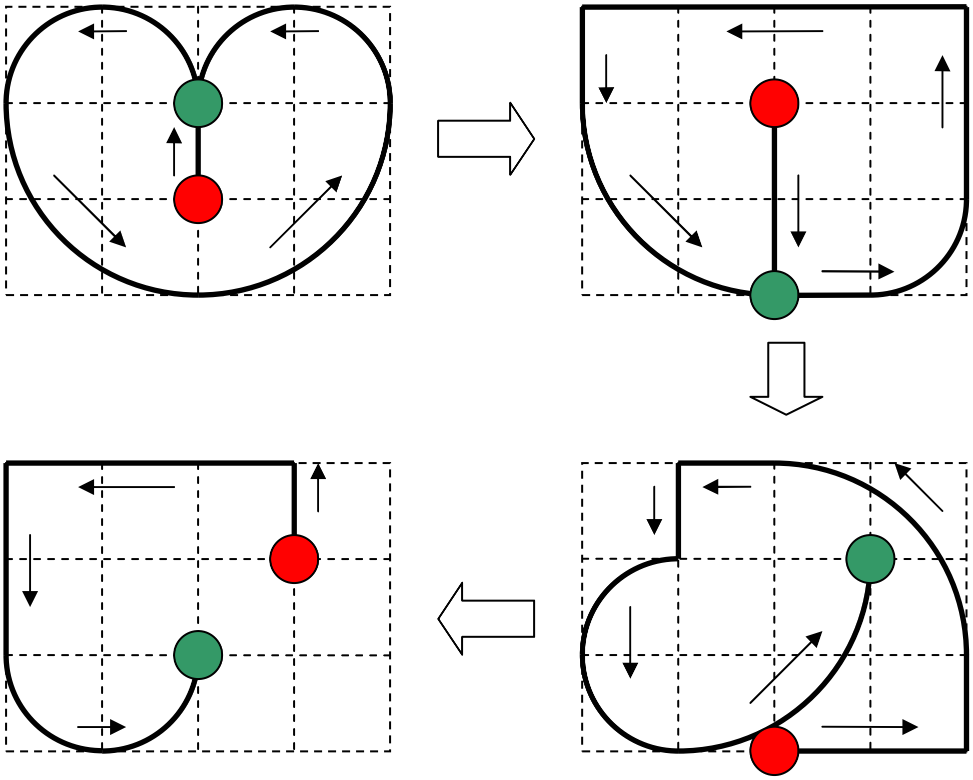


Figure A.2:   Test Reference Trajectory A

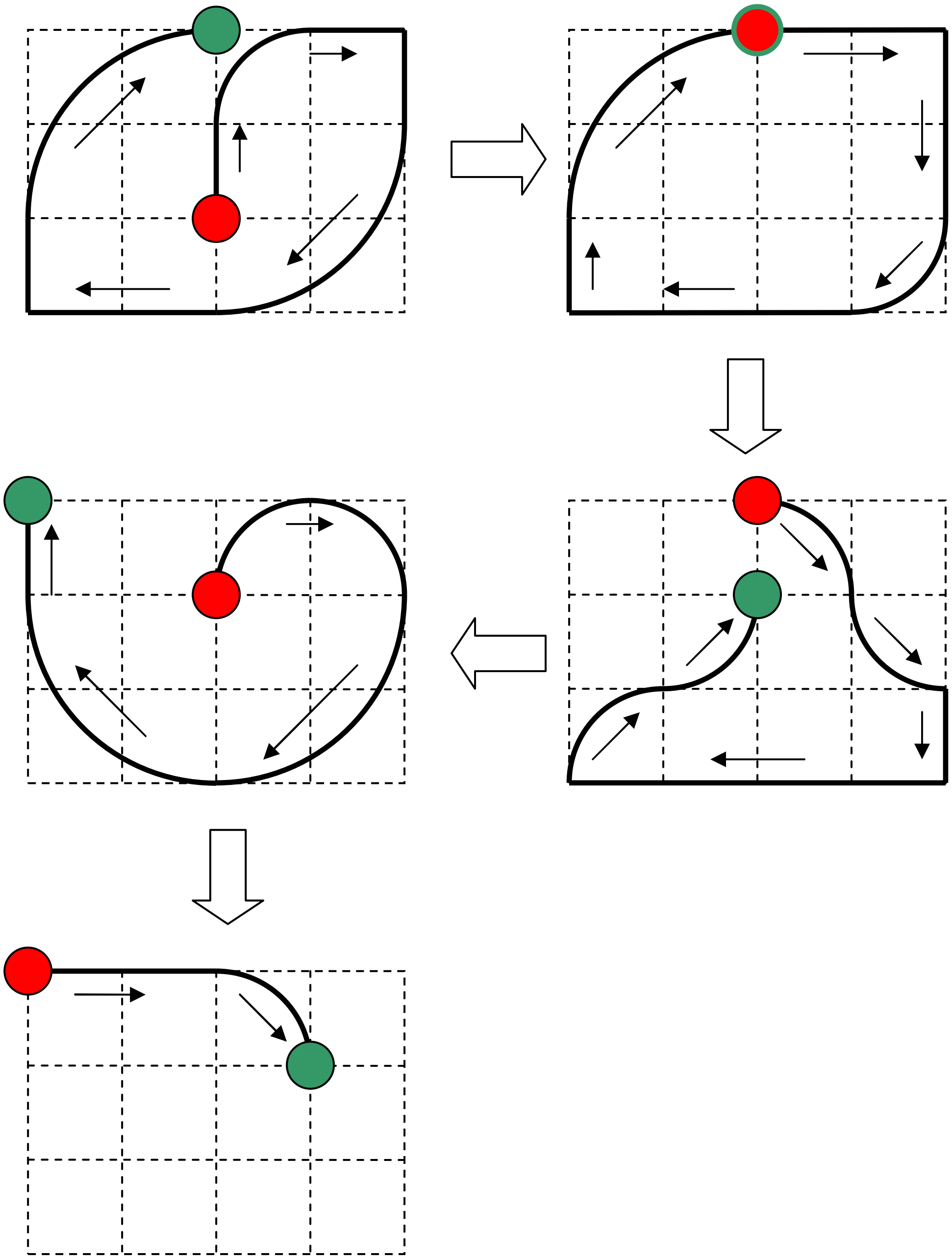
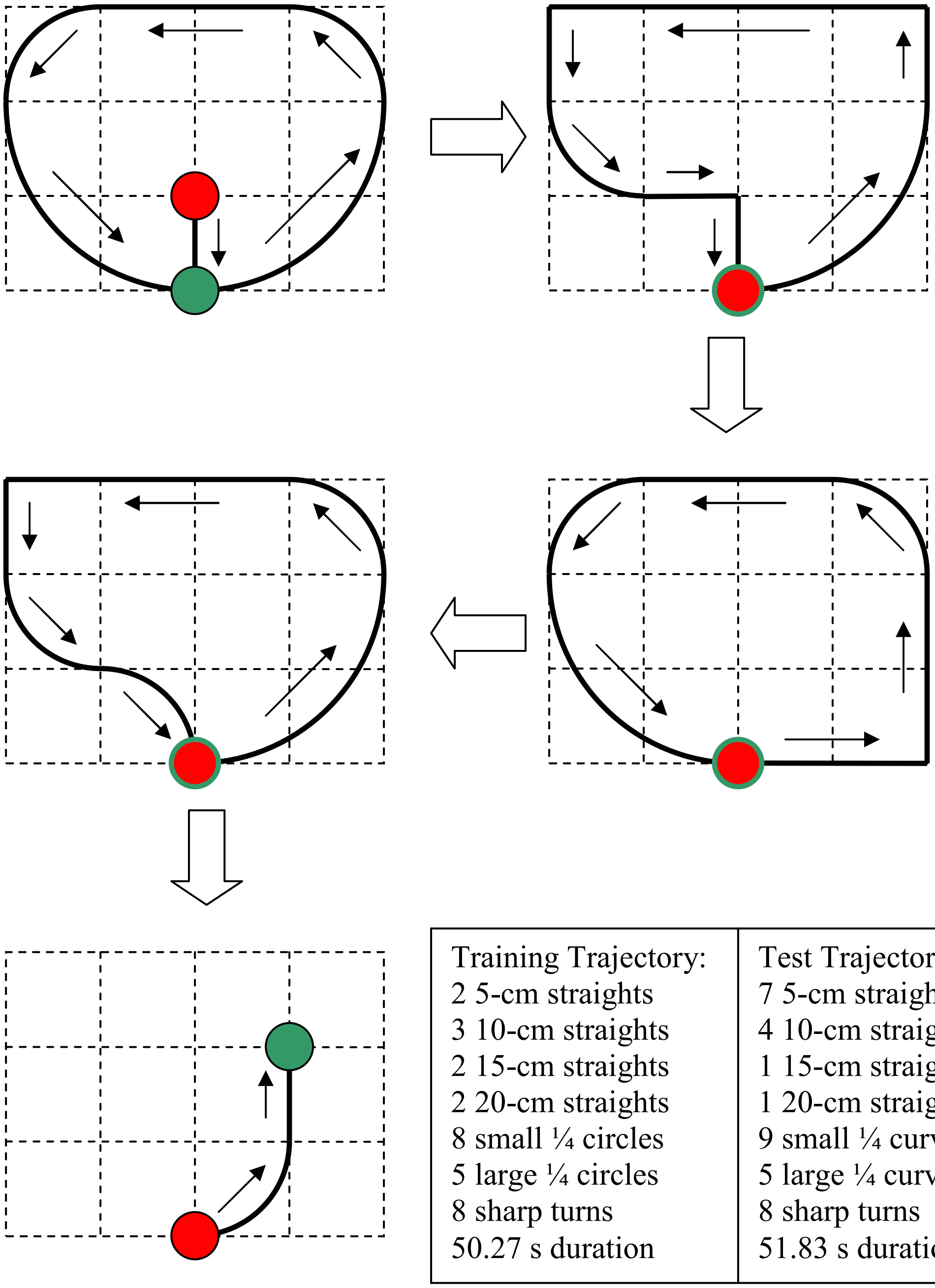


Figure A.3: Test Reference Trajectory B



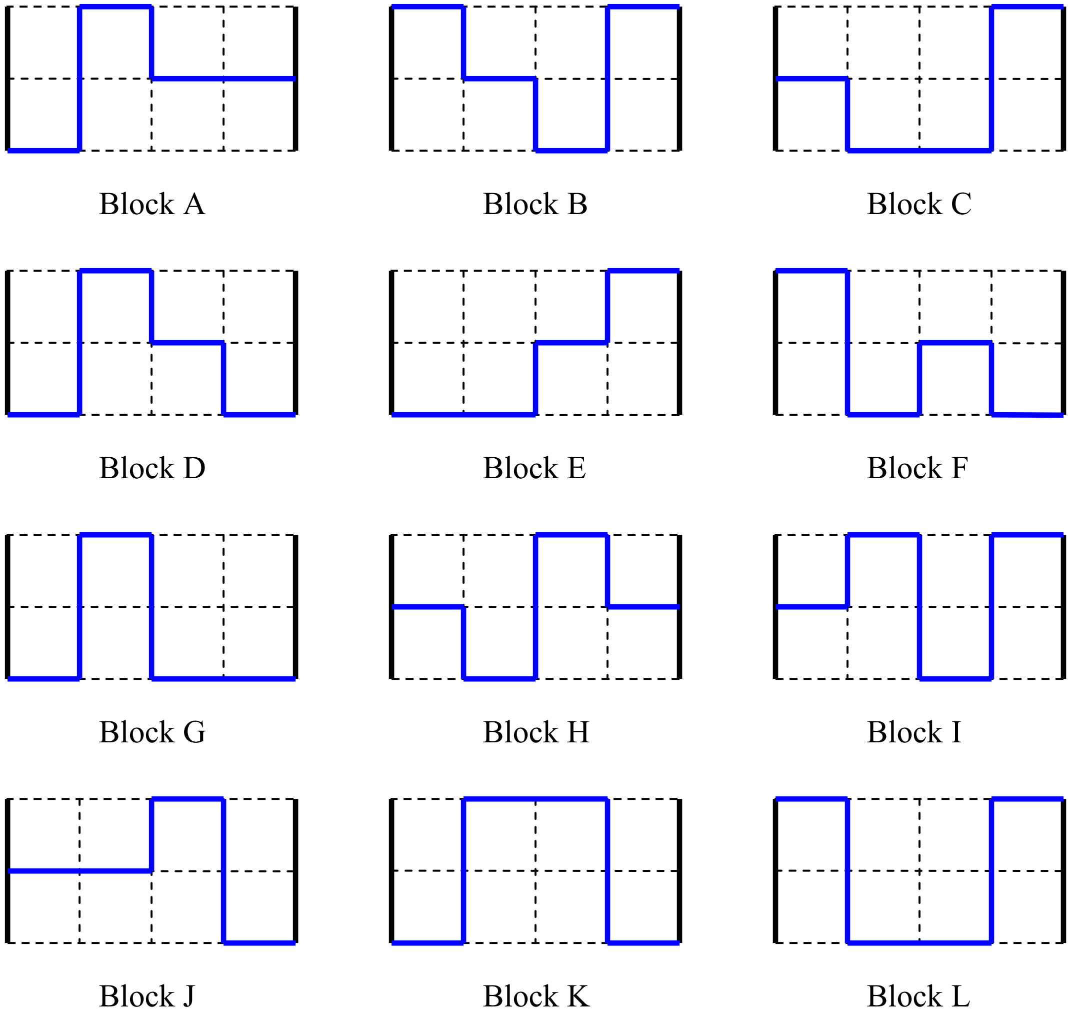
Training Trajectory:	Test Trajectories:
2 5-cm straights	7 5-cm straights
3 10-cm straights	4 10-cm straights
2 15-cm straights	1 15-cm straights
2 20-cm straights	1 20-cm straights
8 small ¼ circles	9 small ¼ curves
5 large ¼ circles	5 large ¼ curves
8 sharp turns	8 sharp turns
50.27 s duration	51.83 s duration

Figure A.4: Test Reference Trajectory C

APPENDIX B

SURFACE CONTOUR BLOCKS

For the surface contour identification, twelve blocks of different contour surfaces are used. They are listed in Fig. B.1. The contour surface is set up in a 2-by-4 grid, each grid being 2 in by 2 in. Notice that the blocks are as perceived by the operator via the master.



Block A

Block B

Block C

Block D

Block E

Block F

Block G

Block H

Block I

Block J

Block K

Block L

Figure B.1: Surface Contour Blocks

Figure B.2 shows the multiple choice questions used to test the subjects in the contour identification test. A total of 12 questions are used, one for each surface block and 4 questions per algorithm tested. For each subject a different combination of the question is used. The circled block indicates the correct surface.

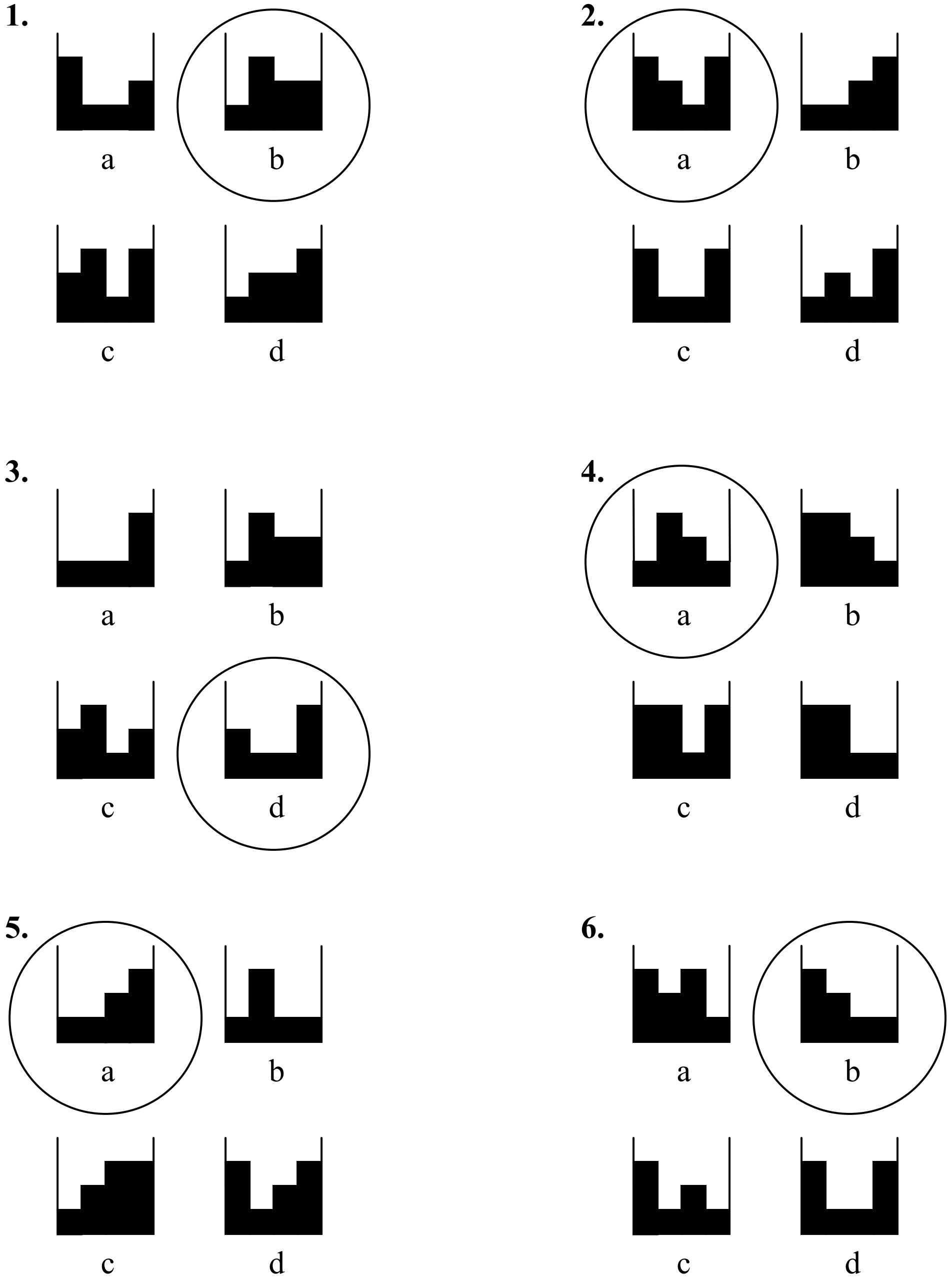


Figure B.2: Contour Identification Questions

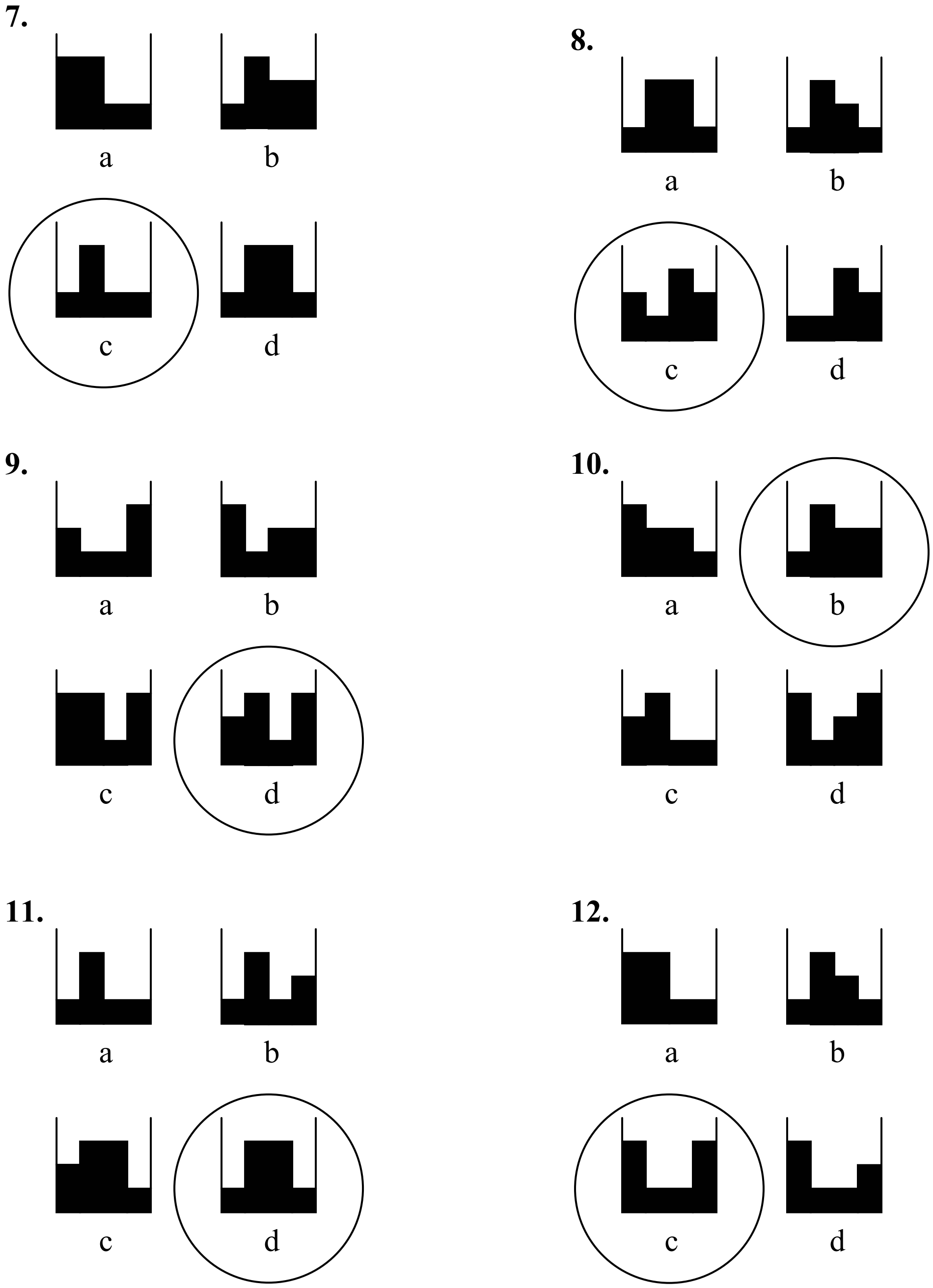
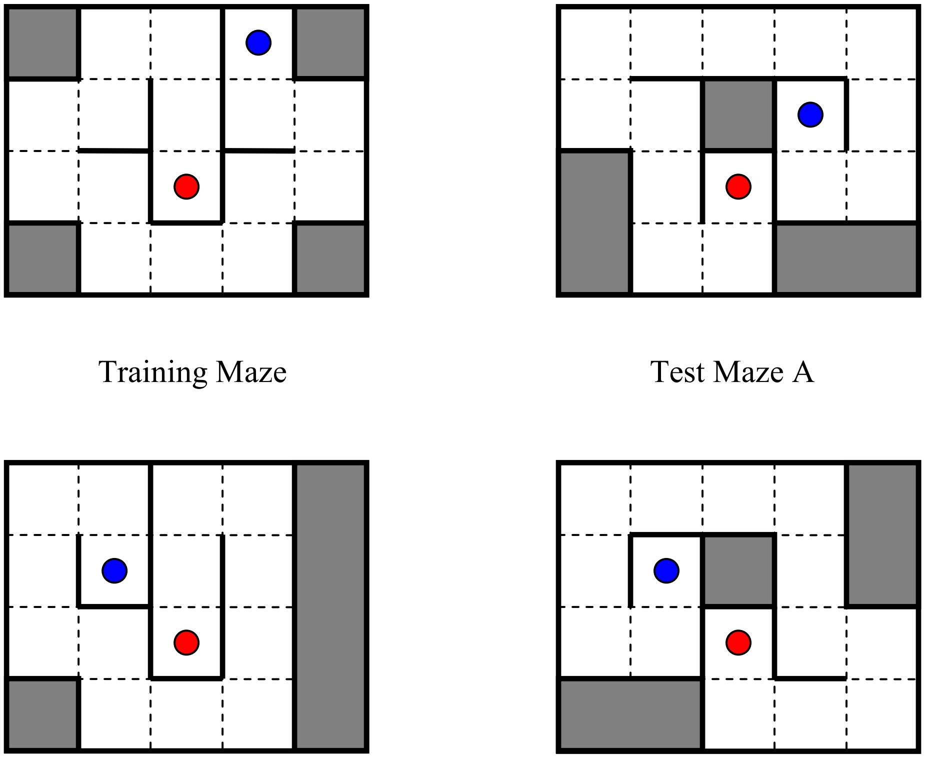


Figure B.2: Continued

APPENDIX C

MAZE BLOCKS

For the maze navigation, one training maze and three test mazes are used as shown in Fig. C.1. The maze is set up in a 4-by-5 grid pattern with each grid being 1.5 in by 1.5 in. The test mazes have similar difficulty because they have the same number of turns and cover the same distance. The red circle indicates the start point, and the blue circle indicates the end point. Notice the shape is that as viewed from the operator's perspective via the master.



Training Maze

Test Maze A

Test Maze B

Test Maze C

Figure C.1: ***Maze Blocks***

APPENDIX D

HUMAN EXPERIMENT RESULTS

Tables D.1 to D.25 show the results of the individual subjects with Table D.0 showing the benchmark scores used as a basis to reward bonuses. Under each subject there is information on his/her gender, handedness, and hours per week on video games. In trajectory tracking the performance index is the average error between the reference and slave trajectories. In contour identification the performance index is the number of multiple choice questions answered correctly. In maze navigation the performance index is the time taken for maze completion. The order of the algorithms tested is listed with the conventional case always the first for all tasks. The other two algorithms may be switched. In trajectory tracking and maze navigation tasks, the maze used for each algorithm is also listed. In contour identification the questions used in the multiple choice test is shown: bold capital letters indicate the correct answers and underlined letters indicates subjects' answers. The user ratings for each algorithm and task are also included with the conventional algorithm being standard at 10. Table D.26 shows the subjective user feedback.

Table D.0: Benchmark Task Results

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.5	3.75	1.75
Contour ID Results (Out of 4)	4	3	3
Maze Results (s)	15	40	30

Table D.1: Subject 1 Results

Male, Right-Handed, Video Game 5 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	2.22	3.44	2.63
Trajectory Algorithm Order	1	2	3
Trajectory Used	A	B	C
Trajectory Ratings	10	4	7
Contour ID Results (Out of 4)	3	3	4
Contour ID Algorithm Order	1	3	2
Contour ID Questions	9. a b c **D** 10. a **B** c d 7. a b **C** d 4. **A** b c d	1. a **B** c d 2. **A** b c d 11. a b c **D** 8. a b **C** d	5. **A** b c d 6. a **B** c d 3. a b c **D** 12. a b **C** d
Contour ID Ratings	10	4	6
Maze Results (s)	25	22	20
Maze Algorithm Order	1	3	2
Maze Used	A	B	C
Maze Ratings	10	5	7

Table D.2: Subject 2 Results

Male, Right-Handed, Video Game 10 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.41	3.48	1.91
Trajectory Algorithm Order	1	3	2
Trajectory Used	B	C	A
Trajectory Ratings	10	6	9
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour ID Questions	1. a **B** c d 10. a **B** c d 11. a b c **D** 4. **A** b c d	5. **A** b c d 2. **A** b c d 3. a b c **D** 8. a b **C** d	9. a b c **D** 6. a **B** c d 7. a b **C** d 12. a b **C** d
Contour ID Ratings	10	6	7
Maze Results (s)	14	31	25
Maze Algorithm Order	1	2	3
Maze Used	B	A	C
Maze Ratings	10	5	8

Table D.3: Subject 3 Results

Male, Right-Handed, Video Game 2 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.24	3.16	1.50
Trajectory Algorithm Order	1	2	3
Trajectory Used	B	A	C
Trajectory Ratings	10	5	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour ID Questions	5. **A** b c d 2. **A** b c d 11. a b c **D** 12. a b **C** d	9. a b c **D** 6. a **B** c d 3. a b c **D** 4. **A** b c d	1. a **B** c d 10. a **B** c d 7. a b **C** d 8. a b **C** d
Contour ID Ratings	10	6	7
Maze Results (s)	10	22	14
Maze Algorithm Order	1	3	2
Maze Used	B	C	A
Maze Ratings	10	6	7

Table D.4: Subject 4 Results

Male, Left-Handed, Video Game 0 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.30	3.79	2.88
Trajectory Algorithm Order	1	3	2
Trajectory Used	C	A	B
Trajectory Ratings	10	7	9
Contour ID Results (Out of 4)	4	2	3
Contour ID Algorithm Order	1	2	3
Contour ID Questions	1. a **B** c d 6. a **B** c d 7. a b **C** d 4. **A** b c d	5. **A** b c d 10. a **B** c d 11. a b c **D** 8. a b **C** d	9. a b c **D** 2. **A** b c d 3. a b c **D** 12. a b **C** d
Contour ID Ratings	10	3	5
Maze Results (s)	16	37	20
Maze Algorithm Order	1	3	2
Maze Used	B	A	C
Maze Ratings	10	4	6

Table D.5: Subject 5 Results

Male, Left-Handed, Video Game 8 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.64	3.92	1.61
Trajectory Algorithm Order	1	2	3
Trajectory Used	C	A	B
Trajectory Ratings	10	4	8
Contour ID Results (Out of 4)	4	3	4
Contour ID Algorithm Order	1	3	2
Contour Questions	5. **A** b c d 10. a **B** c d 3. a b c **D** 4. **A** b c d	9. a b c **D** 2. **A** b c d 7. a b **C** d 8. a b **C** d	1. a **B** c d 6. a **B** c d 11. a b c **D** 12. a b **C** d
Contour ID Ratings	10	3	6
Maze Results (s)	15	100	22
Maze Algorithm Order	1	2	3
Maze Used	B	A	C
Maze Ratings	10	4	7

Table D.6: Subject 6 Results

Male, Right-Handed, Video Game 1 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.41	2.67	1.74
Trajectory Algorithm Order	1	2	3
Trajectory Used	B	C	A
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	3	4
Contour ID Algorithm Order	1	3	2
Contour Questions	9. a b c **D** 10. a **B** c d 11. a b c **D** 12. a b **C** d	1. a **B** c d 2. **A** b c d 3. a b c **D** 4. **A** b c d	5. **A** b c d 6. a **B** c d 7. a b **C** d 8. a b **C** d
Contour ID Ratings	10	5	8
Maze Results (s)	7	29	18
Maze Algorithm Order	1	2	3
Maze Used	C	A	B
Maze Ratings	10	6	7

Table D.7: Subject 7 Results

Male, Right-Handed, Video Game 6 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.27	3.67	1.65
Trajectory Algorithm Order	1	3	2
Trajectory Used	C	B	A
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour Questions	1. a **B** c d 2. **A** b c d 3. a b c **D** 4. **A** b c d	5. **A** b c d 6. a **B** c d 7. a b **C** d 8. a b **C** d	9. a b c **D** 10. a **B** c d 11. a b c **D** 12. a b **C** d
Contour ID Ratings	10	8	8
Maze Results (s)	9	50	17
Maze Algorithm Order	1	2	3
Maze Used	A	B	C
Maze Ratings	10	5	8

Table D.8: Subject 8 Results

Female, Right-Handed, Video Game 0 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.55	3.69	1.67
Trajectory Algorithm Order	1	2	3
Trajectory Used	C	A	B
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	3	2
Contour Questions	5. **A** b c d 2. **A** b c d 7. a b **C** d 4. **A** b c d	9. a b c **D** 6. a **B** c d 11. a b c **D** 8. a b **C** d	1. a **B** c d 10. a **B** c d 3. a b c **D** 12. a b **C** d
Contour ID Ratings	10	6	8
Maze Results (s)	11	85	22
Maze Algorithm Order	1	2	3
Maze Used	B	C	A
Maze Ratings	10	5	8

Table D.9: Subject 9 Results

Male, Right-Handed, Video Game 3 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.38	3.76	1.32
Trajectory Algorithm Order	1	3	2
Trajectory Used	A	B	C
Trajectory Ratings	10	6	9
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour Questions	9. a b c **D** 10. a **B** c d 3. a b c **D** 8. a b **C** d	1. a **B** c d 2. **A** b c d 7. a b **C** d 12. a b **C** d	5. **A** b c d 6. a **B** c d 11. a b c **D** 4. **A** b c d
Contour ID Ratings	10	6	7
Maze Results (s)	14	20	17
Maze Algorithm Order	1	2	3
Maze Used	C	B	A
Maze Ratings	10	6	7

Table D.10: Subject 10 Results

Male, Right-Handed, Video Game 6 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.85	4.21	2.12
Trajectory Algorithm Order	1	3	2
Trajectory Used	A	B	C
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	3	3
Contour ID Algorithm Order	1	3	2
Contour Questions	9. a b c **D** 6. a **B** c d 7. a b **C** d 12. a b **C** d	1. a **B** c d 10. a **B** c d 11. a b c **D** 4. **A** b c d	5. **A** b c d 2. **A** b c d 3. a b c **D** 8. a b **C** d
Contour ID Ratings	10	8	7
Maze Results (s)	22	30	27
Maze Algorithm Order	1	3	2
Maze Used	A	B	C
Maze Ratings	10	8	9

Table D.11: Subject 11 Results

Female, Right-Handed, Video Game 0 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.33	4.15	1.36
Trajectory Algorithm Order	1	3	2
Trajectory Used	A	C	B
Trajectory Ratings	10	4	8
Contour ID Results (Out of 4)	4	3	4
Contour ID Algorithm Order	1	2	3
Contour Questions	1. a **B** c d 10. a **B** c d 3. a b c **D** 12. a b **C** d	5. **A** b c d 2. **A** b c d 7. a b **C** d 4. **A** b c d	9. a b c **D** 6. a **B** c d 11. a b c **D** 8. a b **C** d
Contour ID Ratings	10	3	7
Maze Results (s)	16	>300 (Fail)	14
Maze Algorithm Order	1	2	3
Maze Used	C	A	B
Maze Ratings	10	4	7

Table D.12: Subject 12 Results

Male, Right-Handed, Video Game 0 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.71	3.47	2.28
Trajectory Algorithm Order	1	2	3
Trajectory Used	B	A	C
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour Questions	1. a **B** c d 2. **A** b c d 11. a b c **D** 8. a b **C** d	5. **A** b c d 6. a **B** c d 3. a b c **D** 12. a b **C** d	9. a b c **D** 10. a **B** c d 7. a b **C** d 4. **A** b c d
Contour ID Ratings	10	4	5
Maze Results (s)	10	35	29
Maze Algorithm Order	1	3	2
Maze Used	C	B	A
Maze Ratings	10	5	5

Table D.13: Subject 13 Results			
Male, Right-Handed, Video Game 1 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.22	2.92	1.96
Trajectory Algorithm Order	1	2	3
Trajectory Used	A	B	C
Trajectory Ratings	10	4	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	3	2
Contour Questions	1. a **B** c d 2. **A** b c d 7. a b **C** d 12. a b **C** d	5. **A** b c d 6. a **B** c d 11. a b c **D** 4. **A** b c d	9. a b c **D** 10. a **B** c d 3. a b c **D** 8. a b **C** d
Contour ID Ratings	10	2	7
Maze Results (s)	11	24	17
Maze Algorithm Order	1	3	2
Maze Used	B	C	A
Maze Ratings	10	5	7

Table D.14: Subject 14 Results			
Female, Right-Handed, Video Game 2 Hours/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.63	4.12	1.81
Trajectory Algorithm Order	1	3	2
Trajectory Used	A	C	B
Trajectory Ratings	10	7	9
Contour ID Results (Out of 4)	4	1	3
Contour ID Algorithm Order	1	2	3
Contour Questions	1. a **B** c d 6. a **B** c d 11. a b c **D** 12. a b **C** d	5. **A** b c d 10. a **B** c d 3. a b c **D** 4. **A** b c d	9. a b c **D** 2. **A** b c d 7. a b **C** d 8. a b **C** d
Contour ID Ratings	10	4	8
Maze Results (s)	19	77	12
Maze Algorithm Order	1	3	2
Maze Used	C	A	B
Maze Ratings	10	6	10

Table D.15: Subject 15 Results			
Male, Right-Handed, Video Game 1 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.26	3.58	2.16
Trajectory Algorithm Order	1	2	3
Trajectory Used	A	C	B
Trajectory Ratings	10	6	8
Contour ID Results (Out of 4)	4	3	4
Contour ID Algorithm Order	1	3	2
Contour Questions	5. **A** b c d 10. a **B** c d 11. a b c **D** 8. a b **C** d	9. a b c **D** 2. **A** b c d 3. a b c **D** 12. a b **C** d	1. a **B** c d 6. a **B** c d 7. a b **C** d 4. **A** b c d
Contour ID Ratings	10	6	8
Maze Results (s)	14	74	33
Maze Algorithm Order	1	2	3
Maze Used	A	B	C
Maze Ratings	10	5	8

Table D.16: Subject 16 Results			
Male, Right-Handed, Video Game 0 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.14	2.72	1.23
Trajectory Algorithm Order	1	3	2
Trajectory Used	B	A	C
Trajectory Ratings	10	5	8
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour Questions	5. **A** b c d 2. **A** b c d 3. a b c **D** 8. a b **C** d	9. a b c **D** 6. a **B** c d 7. a b **C** d 12. a b **C** d	1. a **B** c d 10. a **B** c d 11. a b c **D** 4. **A** b c d
Contour ID Ratings	10	6	8
Maze Results (s)	8	25	13
Maze Algorithm Order	1	3	2
Maze Used	B	A	C
Maze Ratings	10	6	8

Table D.17: Subject 17 Results

Male, Right-Handed, Video Game 1 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.52	4.90	1.79
Trajectory Algorithm Order	1	3	2
Trajectory Used	C	A	B
Trajectory Ratings	10	3	7
Contour ID Results (Out of 4)	4	3	4
Contour ID Algorithm Order	1	3	2
Contour Questions	9. a b c **D** 6. a **B** c d 3. a b c **D** 4. **A** b c d	1. a **B** c d 10. a **B** c d 7. a b **C** d 8. a b **C** d	5. **A** b c d 2. **A** b c d 11. a b c **D** 12. a b **C** d
Contour ID Ratings	10	4	7
Maze Results (s)	9	66	29
Maze Algorithm Order	1	3	2
Maze Used	C	A	B
Maze Ratings	10	2	5

Table D.18: Subject 18 Results

Male, Left-Handed, Video Game 1 Hour/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.40	3.83	2.45
Trajectory Algorithm Order	1	2	3
Trajectory Used	C	B	A
Trajectory Ratings	10	5	8
Contour ID Results (Out of 4)	4	4	3
Contour ID Algorithm Order	1	2	3
Contour Questions	5. **A** b c d 6. a **B** c d 7. a b **C** d 8. a b **C** d	9. a b c **D** 10. a **B** c d 11. a b c **D** 12. a b **C** d	1. a **B** c d 2. **A** b c d 3. a b c **D** 4. **A** b c d
Contour ID Ratings	10	6	6
Maze Results (s)	9	29	14
Maze Algorithm Order	1	2	3
Maze Used	C	B	A
Maze Ratings	10	5	7

Table D.19: Subject 19 Results			
Male, Right-Handed, Video Game 6 Hours/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.65	2.80	1.24
Trajectory Algorithm Order	1	2	3
Trajectory Used	A	C	B
Trajectory Ratings	10	8	10
Contour ID Results (Out of 4)	4	3	0
Contour ID Algorithm Order	1	3	2
Contour Questions	9. a b c **D̲** 2. **A̲** b c d 7. a b **C̲** d 8. a b **C̲** d	1. a **B̲** c d 6. a **B̲** c d 11. a b c̲ **D** 12. a b **C̲** d	5. **A** b̲ c d 10. a **B** c d̲ 3. a b̲ c **D** 4. **A** b c̲ d
Contour ID Ratings	10	3	2
Maze Results (s)	20	31	24
Maze Algorithm Order	1	2	3
Maze Used	A	C	B
Maze Ratings	10	6	8

Table D.20: Subject 20 Results			
Male, Right-Handed, Video Game 8 Hours/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.11	3.18	1.57
Trajectory Algorithm Order	1	2	3
Trajectory Used	B	C	A
Trajectory Ratings	10	3	7
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	2	3
Contour Questions	5. **A̲** b c d 6. a **B̲** c d 3. a b c **D̲** 12. a b **C̲** d	9. a b c **D̲** 10. a **B̲** c d 7. a b **C̲** d 4. **A̲** b c d	1. a **B̲** c d 2. **A̲** b c d 11. a b c **D̲** 8. a b **C̲** d
Contour ID Ratings	10	6	7
Maze Results (s)	13	35	18
Maze Algorithm Order	1	3	2
Maze Used	A	C	B
Maze Ratings	10	3	6

Table D.21: Subject 21 Results			
Male, Right-Handed, Video Game 0 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.39	3.49	1.37
Trajectory Algorithm Order	1	3	2
Trajectory Used	B	A	C
Trajectory Ratings	10	5	9
Contour ID Results (Out of 4)	4	4	4
Contour ID Algorithm Order	1	3	2
Contour Questions	5. **A** b c d 6. a **B** c d 11. a b c **D** 4. **A** b c d	9. a b c **D** 10. a **B** c d 3. a b c **D** 8. a b **C** d	1. a **B** c d 2. **A** b c d 7. a b **C** d 12. a b **C** d
Contour ID Ratings	10	8	7
Maze Results (s)	13	50	31
Maze Algorithm Order	1	2	3
Maze Used	A	C	B
Maze Ratings	10	6	8

Table D.22: Subject 22 Results			
Female, Right-Handed, Video Game 0 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.98	4.22	2.04
Trajectory Algorithm Order	1	3	2
Trajectory Used	C	B	A
Trajectory Ratings	10	6	8.5
Contour ID Results (Out of 4)	4	2	4
Contour ID Algorithm Order	1	2	3
Contour Questions	9. a b c **D** 2. **A** b c d 3. a b c **D** 12. a b **C** d	1. a **B** c d 6. a **B** c d 7. a b **C** d 4. **A** b c d	5. **A** b c d 10. a **B** c d 11. a b c **D** 8. a b **C** d
Contour ID Ratings	10	4	5
Maze Results (s)	16	23	16
Maze Algorithm Order	1	3	2
Maze Used	C	B	A
Maze Ratings	10	6	8

Table D.23: Subject 23 Results			
Female, Right-Handed, Video Game 0 Hour/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.68	4.01	3.30
Trajectory Algorithm Order	1	2	3
Trajectory Used	C	B	A
Trajectory Ratings	10	4	6
Contour ID Results (Out of 4)	4	3	3
Contour ID Algorithm Order	1	3	2
Contour Questions	9. a b c **D** 6. a **B** c d 11. a b c **D** 8. a b **C** d	1. a **B** c d 10. a **B** c d 3. a b c **D** 12. a b **C** d	5. **A** b c d 2. **A** b c d 7. a b **C** d 4. **A** b c d
Contour ID Ratings	10	7	4
Maze Results (s)	21	19	26
Maze Algorithm Order	1	2	3
Maze Used	B	C	A
Maze Ratings	10	8	6

Table D.24: Subject 24 Results			
Male, Right-Handed, Video Game 3 Hours/Week			
	Conventional	Wave	WAPD
Trajectory Error (mm)	1.46	3.83	1.70
Trajectory Algorithm Order	1	3	2
Trajectory Used	B	C	A
Trajectory Ratings	10	2	7
Contour ID Results (Out of 4)	4	2	4
Contour ID Algorithm Order	1	3	2
Contour Questions	1. a **B** c d 10. a **B** c d 7. a b **C** d 8. a b **C** d	5. **A** b c d 2. **A** b c d 11. a b c **D** 12. a b **C** d	9. a b c **D** 6. a **B** c d 3. a b c **D** 4. **A** b c d
Contour ID Ratings	10	3	6
Maze Results (s)	26	60	36
Maze Algorithm Order	1	3	2
Maze Used	A	C	B
Maze Ratings	10	4	5

Table D.25: Subject 25 Results

Male, Right-Handed, Video Game 3 Hours/Week

	Conventional	Wave	WAPD
Trajectory Error (mm)	1.45	3.71	1.48
Trajectory Algorithm Order	1	2	3
Trajectory Used	A	C	B
Trajectory Ratings	10	3	9
Contour ID Results (Out of 4)	4	4	3
Contour ID Algorithm Order	1	2	3
Contour Questions	1. a **B** c d 10. a **B** c d 11. a b c **D** 4. **A** b c d	5. **A** b c d 2. **A** b c d 3. a b c **D** 8. a b **C** d	9. a b c **D** 6. a **B** c d 7. a b **C** d 12. a b **C** d
Contour ID Ratings	10	5	6
Maze Results (s)	14	34	18
Maze Algorithm Order	1	3	2
Maze Used	C	B	A
Maze Ratings	10	6	7

Table D.26: Subjective Feedback

Subject	Wave	WAPD
1	Feels wobbly in motion tracking. Tend to skip corners. Contact force is too soft to detect contact effectively. Contact force is not steady.	Not wobbly in motion tracking. Slave eventually goes where the master was with delay. Contact force is clearer but pushes back out a little too hard.
2	Impossible to track in free space. Can feel force feedback oscillations that disrupt user in tracking task. Can feel some oscillation in contact force too.	Slave follows the master well with no problem in motion tracking. Contact force is crisp.
3	Hard to track motion. Too much oscillation in slave trajectory. Hard to detect contact from oscillation in force feedback.	Good at free space tracking. Sometimes jerk master off hand if hit contact hard.
4	Poor motion tracking when moving fast. There always seem to be an error even when not moving in free space. Force feedback is not clear and confusing.	Better tracking regardless of speed. Clearer force feedback but tend to kick back pretty hard.
5	Motion tracking is bad. Not as much feedback under contact and more gradual contact force. Force feedback is not steady and jumps around a little.	Better for motion tracking. Clear indication when hit wall but still lacks accuracy in contact location. Contact location seems different from different directions.

Table D.26: Continued

6	Awful tracking especially when moving fast. Contact force feels mushy.	No problem in tracking. Lots of force feedback and more immediate. More solid indication of force good for bumping into walls and finding way around maze.
7	Bad motion tracking. Even error between master and slave when both are stationary. Contact force is too soft and wavy. Can't tell wall location.	Excellent tracking since the slave follows the master. Force feedback is abrupt which is good for wall detection.
8	Seems to skip too much in free space motion tracking and very unpredictable. Very loose under contact like hitting jelly. Force jumps around before settling.	Tracks well in free space. Force feedback is abrupt but good because easier to navigate and identify rigid contact.
9	Cut corners in free space motion tracking. Persistent steady state error between master and slave positions is annoying. Hard to get a definite force feedback under contact. Did manage to learn to deal with it.	No problem in free space motion tracking. Easiest to compensate for because of sharp feedback force.
10	Motion tracking is impossible. Slave does not follow and always error between master and slave even motionless. Contact is ok but too soft.	Motion tracking is much better. Do feel a clear bump under contact but can not get clear boundary. Contact location seems different when moving in from moving out.
11	Can not track in motion. Can not feel the force feedback in contact.	Follow the line better in motion tracking. Better at feeling contact force.
12	Rocky in motion tracking. Feels more rigid under contact force.	Motion is smoother for tracking. Delay seems more significant. Feels similar to wave variables.
13	Needs to go slow for good motion tracking. Very hard to confirm bumps of contact force because of softness. Sensation is also wavy in force.	Better motion tracking and can afford to go fast. Sharper force feedback for quick contact detection. Somewhat pushy in forcing master out of contact.
14	Always error in motion tracking between master and slave, even when standing still. Contact feels soft and ambiguous.	No problem in motion tracking except the delay but slave ultimately follows. Abrupt contact gives almost instant detection.

Table D.26: Continued

15	Horrible motion tracking. Slave moves all around master but never quite on top of it. Force reaction in contact is slow and ambiguous. Need to guess sometimes.	Feels a little viscous but the slave always tries to follow the master position via the most direct route. Force reaction in contact is very quick. Know the state quickly.
16	Motion tracking seems more random. Force feedback in contact also seems more random.	Easy to anticipate position despite delay. Also predictable in contact with clear force feedback. More abrupt but able to adapt to it quickly.
17	Does not track motion perfectly. The contact force is there but too soft hence hard to judge the level and location of contact. Take more time to figure out contact location.	Slave follows master position exactly. Sharp contact force and takes less time to figure out contact location. May kick back against the hand pretty hard when in contact.
18	Can hardly track motion. Error even in steady state. Delay seems to amplify size of contour. Contact force is very soft but more continuous which allows for contour tracking.	Much better at motion tracking with no error in master and slave positions. Contact force is clear but not continuous which confuses contact location. Better for maze because only need contact detection, and the clear contact force is a definite plus.
19	Position drift error at steady state is noticeable. Force feedback can get wavy and interferes with hand trying to track motion. Contact feels like jelly but still smooth and gradual with less distortion.	Slave tracks master better without overshoot. Too pushy under contact. Good for detection, but there is distortion in force feedback where contact location seems different going in from coming out. This can get confusing.
20	Got too much drift in motion tracking and need to artificially exaggerate movement. Contact force is sloppy and let you push too far into wall.	Still lags but tracks motion better. Sharper force feedback but once used to it provides immediate contact detection that is useful.
21	Very jumpy in motion tracking. Easier to determine contour shape because more continuous force feedback. Still contact force is too soft.	Trajectory control is smoother and more predictable. Force feedback is more responsive. Good for contact detection but hard to determine contact location. Force distorted in different directions.

Table D.26: Continued

22	Very jerky and unpredictable in tracking motion and force. Wobbly in following line. Force feedback is like mush and wavy.	Still feel the delay in motion and force but predictable. Force feedback is abrupt and clear. Easier to map contact walls.
23	Does not track motion well especially in corners. Contact force is soft. Pushes very deep into wall before feeling force.	Slave tracks master position well. Higher force is good at detecting but not tracking contour. Force is also jumpier and distorted in different directions. Tries hard to push hand out of contact. Very confusing.
24	Slave will not follow properly in motion tracking. Force feedback is sloppy and confusing because unsteady.	Good motion tracking. Force feedback is clearer and more immediate. Can be certain about contact sensation.
25	Motion tracking unpredictable and inaccurate. Drift is very noticeable even when not moving. Contact force is weak, unpredictable, inaccurate, and jumps. Takes some time for it to settle.	Motion tracking is good. Slave follows master trajectory well. Contact force is crisp and accurate.

BIBLIOGRAPHY

1. Etkin, B. and Reid, L. D., *Dynamics of Flight: Stability and Control*, 3rd ed., John Wiley & Sons, Inc., USA, 1996.

2. Sheridan, T. B., Telerobotics, *Automatica*, Volume 25, Issue 4, pp. 487-507, July 1989.

3. www.sensable.com, 9/06.

4. www.intuitivesurgical.com, 9/06.

5. www.novint.com, 9/06.

6. Ogata, K., *Modern Control Engineering*, 3rd ed., Prentice Hall, New Jersey, 1997.

7. Ferrell, W. R., Remote Manipulation with Transmission Delay, *IEEE Transactions on Human Factors in Electronics*, pp. 24-32, Sept. 1965.

8. Ferrell, W. R., Delay Force Feedback, *IEEE Transactions on Human Factors in Electronics*, pp. 449-455, Oct. 1966.

9. Taylor, K. and Dalton, B., Internet Robots: A New Robotics Niche, *IEEE Robotics and Automation Magazine*, pp. 27-34, Sept. 1999.

10. Book, W. J., Rouse, M. D., and Koeppen, K., Internet Access to a Fluid Power Mechatronics Laboratory, *Proceedings of International Conference on Advanced Intelligent Mechatronics*, Vol. 2, pp. 1112-1117, July 2001.

11. Backes, P. G., Tso, K. S., and Tharp, G. K., Mars Pathfinder Mission Internet-Based Operations Using WITS, *Proceedings of International Conference on Robotics and Automation*, pp. 284-291, May 1993.

12. Anderson, R. J. and Spong, M.W., Bilateral Control of Teleoperators with Time Delay, *IEEE Transaction on Automatic Control*, Vol. 34, Issue 5, pp. 494-501, May 1989.

13. Anderson, R. J. and Spong, M. W., Asymptotic Stability for Force Reflecting Teleoperators with Time Delay, *The International Journal of Robotics Research*, Vol. 11, Issue 2, pp. 135-149, April 1992.

14. Niemeyer, G. and Slotine, J. E., Stable Adaptive Teleoperation, *IEEE Transaction of Oceanic Engineering*, Vol. 16, Issue 1, pp. 152-162, Jan. 1991.

15. Niemeyer, G. and Slotine, J. E., Using Wave Variables for System Analysis and Robot Control, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1619-1625, April 1997.

16. Munir S., *Internet-Based Teleoperation*, Ph. D. Thesis, Department of Mechanical Engineering, Georgia Institute of Technology, March 2001.

17. Alise, M., Roberts, R. G., and Repperger, D. W., Time-Delayed Teleoperation Using Wave Variables on Multiple Degree-of-Freedom Systems, *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory*, pp. 253-257, March 2005.

18. Lawn, C. A. and Hannaford, B., Performance Testing of Passive Communication and Control in Teleoperation with Time Delay, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 776-783, May 1993.

19. Benedetti, C., Franchini, M., and Fiorini, P., Stable Tracking in Variable Time-Delay Teleoperation, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 4, pp. 2252-2257, Nov. 2001.

20. Lew, J. Y., Repperger, D. W., and Berlin, J., Wave Variable Based Teleoperation with Time Delay: Application to Space Based Laser Maintenance, *Aerospace Conference Proceedings*, Vol. 5, pp. 2912-2919, March 2004.

21. Munir, S. and Book, W. J., Wave-Based Teleoperation with Prediction, *Proceedings of the American Control Conference*, Vol. 6, pp. 4605-4611, June 2001.

22. Ganjetar, S., Momeni, H., and Janabi-Sharill, F., Teleoperation Systems Design Using Augmented Wave-Variables and Smith Predictor Method for Reducing Time-Delay Effects, *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 333-338, Oct. 2002.

23. Arioui, H., Kheddar, A., and Mammar, S., A Predictive Wave-Based Approach for Time Delayed Virtual Environments Haptics Systems, *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pp. 134-139, Sept. 2002.

24. Kosuge, K., Murayama, H., and Takeo, K., Bilateral Feedback Control of Telemanipulators via Computer Network, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 1380-1385, Nov. 1996.

25. Yokokohji, Y., Imaida, T., and Yoshikawa, T., Bilateral Teleoperation under Time-Varying Communication Delay, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 1854-1859, Oct. 1999.

26. Yokokohji, Y., Imaida, T., and Yoshikawa, T., Bilateral Control with Energy Balance Monitoring under Time-Varying Communication Delay, *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2684-2689, April 2000.

27. Mirfankhrai, T. and Payandeh, S., A Delay Prediction Approach for Teleoperation Over the Internet, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 2178-2183, May 2002.

28. Zhang T. and L. Y. C., 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*, pp. 273-278, Jan. 2006.

29. Carignan, C. R. and Olsson, P. A., Cooperative Control of Virtual Objects over the Internet Using Force-Reflecting Master Arms, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1221-1226, April 2004.

30. Massaquoi, S. G. and Slotine, J. E., The Intermediate Cerebellum May Function as a Wave Variable Processor, *Neuroscience Letters*, pp. 60-64, 1996.

31. Massaquoi, S. G. Sarma, S. V., and Dahleh, M., Reduction of a Wave-Variable Biological Arm Control Model, *Proceedings of the American Control Conference*, Vol. 4, pp. 2405-2409, June 2000.

32. Lee, J. H., Cho, C. H., Song, J. B., Hwang, C. S., and Kim, M. S., Haptic Interface Using Delayed Reflection Wave: Application to a Passive Haptic Device, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2471-2476, April 2005.

33. Kim, H. W., Suh, I. H., and Yi, B. J., A Stabilizing Control Technique for Bilateral Teleoperation with Time Delay, *IEEE Workshop on Advanced Robotics and Its Social Impacts*, pp. 103-108, June 2005.

34. Tanner, N. A. and Niemeyer, G., Improving Perception in Time Delayed Teleoperation, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 354-359, April 2005.

35. Tanner, N. A. and Niemeyer, G., High-Frequency Acceleration Feedback in Wave Variable Telerobotics, *ASME Transaction on Mechatronics*, Vol. 11, Issue 2, pp. 119-127, April 2006.

36. Tanner, N. A. and Niemeyer, G., Stabilization Through Gyration: A Wave Variable Approach to High Frequency Force Feedback in Telerobotics, *14th*

Symposium on Haptic Interfaces for Virtual Environment and Teleoperation Systems, pp. 153-159, March 2006.

37. Tanner, N. A. and Niemeyer, G., Practical Limitations of Wave Variable Controller in Teleoperation, *IEEE Conference on Robotics, Automation and Mechatronics*, Vol. 1, pp. 25-30, Dec. 2004.

38. Tanner, N. A. and Niemeyer, G., Online Tuning of Wave Impedance in Telerobotics, *IEEE Conference on Robotics, Automation, and Mechatronics*, Vol. 1, pp. 7-12, Dec. 2004.

39. Diolaiti, N. and Niemeyer, G., Wave Haptics: Providing Stiff Coupling to Virtual Environments, *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 185-192, March 2006.

40. Smith, J. M., Closer Control of Loops with Dead Time, *Chemical Engineering Progress*, Vol. 53, Issue 5, pp. 217-219, May 1957.

41. Bahill, A., A Simple Adaptive Smith-Predictor for Controlling Time-Delay Systems: A Tutorial, *IEEE Control Systems Magazine*, Vol. 3, Issue 2, pp. 16-22, May 1983.

42. He, S. Z., Xu, F. L., and Tan, S., A New Adaptive Smith Predictor Controller, *TENCON '92 Technology Enabling Tomorrow: Computers, Communication and Automation Towards the 21st Century*, Vol. 2, pp. 1038-1042, Nov. 1992.

43. Huang, J. J. and DeBra, D. B., Automatic Smith-Predictor Tuning Using Optimal Parameter Mismatch, *IEEE Transactions on Control Systems Technology*, Vol. 10, Issue 3, pp. 447-459, May 2002.

44. Wu, S., Watanabe, K., Muramatsu, E., Ariga, Y., and Endo, S., Robust Stability of Approximate Smith Predictor Control Systems, *SICE 2004 Annual Conference*, Vol. 2, pp. 1522-1527, Aug. 2004.

45. Hashtrudi-Zaad, K. and Salcudean, S. E., Adaptive Transparent Impedance Reflecting Teleoperation, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1369-1374, April 1996.

46. Gao, W., Li, Y. C., Liu, G. J., and Zhang, T., An Adaptive Fuzzy Smith Control of Time-Varying Processes with Dominant and Variable Delay, *Proceedings of the American Control Conference*, Vol. 1, pp. 220-224, June 2003.

47. Ivanova, E. and Hadjiski, M., Rules Based Adaptation of Smith Predictor, *1st International IEEE Symposium on Intelligent Systems*, Vol. 3, pp. 26-30, Sept. 2002.

48. Huang, J. Q. and Lewis, F. L., Neural-Network Predictive Control for Nonlinear Dynamic Systems with Time-Delay, *IEEE Transactions on Neural Networks*, Vol. 14, Issue 2, pp. 377-389, March 2003.

49. Wang, P. G., Feng, H. P., and Zong, X. P., Smith Predictive Control Based on NN, *Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 7, pp. 4179-4183, Aug. 2005.

50. Smith, A. C. and Hashtrudi-Zaad, K., Neural Network-Based Teleoperation Using Smith Predictors, *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Vol. 3, pp. 1654-1659, Aug. 2005.

51. Bejczy, A. K., Kim, W. S., and Vienema, S. C., The Phantom Robot: Predictive Display for Teleoperation with Time Delay, *Proceedings of the IEEE International Conference on Intelligent Robots and Automation*, Vol. 1, pp. 546-551, May 1990.

52. Kotoku, T., A Predictive Display with Force Feedback and Its Application to Remote Manipulation System with Transmission Time Delay, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 239-246, July 1992.

53. Funda, J. and Paul, R. P., Model-Based, Delay-Tolerant Teleoperation in Unstructured Environments, *Proceedings of the 6th Mediterranean Electrotechnical Conference*, Vol. 2, pp. 908-911, May 1991.

54. Tsumaki, Y., Hoshi, Y., Naruse, H., and Uchiyama, M., Virtual Reality Based Teleoperation Which Tolerates Geometrical Modeling Errors, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 1023-1030, Nov. 1996.

55. Penin, L. F. Matsumoto, K., and Wakabayashi, S., Force Reflection for Time-Delayed Teleoperation of Space Robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3120-3125, April 2000.

56. Llewellyn, F. B., Some Fundamental Properties of Transmission System, *Proceedings of IRE*, Vol. 40, No. 5, pp. 271-283, 1952.

57. Oboe, R. and Fiorini, P., A Design and Control Environment for Internet-Based Telerobotics, *International Journal of Robotics Research*, Vol. 17, No. 4, pp. 433-439, 1998.

58. Imaida, T., Yokokohji, Y., Doi, T., Oda, M., and Yoshikawa, T., Ground-Space Bilateral Teleoperation of ETS-VII Robot Arm by Direct Bilateral Coupling under 7-S Time Delay Condition, *IEEE Transaction in Robotics and Automation*, Vol. 20, No. 3, pp. 499-511, June 2004.

59. Kim, W. S, Shared Compliant Control: a Stability Analysis and Experiments, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 620-623, Nov. 1990.

60. Hannaford, B. and Kim, W.S., Force Reflection, Shared Control, and Time Delay in Telemanipulation, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, pp. 133-137, Nov. 1989.

61. Kim, W. S., Hannaford, B., and Bejczy, A.K., Force Reflection and Shared Compliant Control in Operating Telemanipulators with Time Delay, *IEEE Transactions on Robotics and Automation*, Vol. 8, Issue 2, pp. 176-185, April 1992.

62. Kim, W. S., Experiments with a Predictive Display and Shared Compliant Control for Time-Delayed Teleoperation, *Proceedings of the IEEE International Conference of the Engineering in Medicine and Biology Society*, pp. 1905-1906, Nov. 1990.

63. Hannaford, B. and Ryu, J. H., Time-Domain Passivity Control of Haptic Interface, *IEEE Transaction on Robotics and Automation*, Vol. 18, Issue 1, pp. 1-10, Feb. 2002.

64. Koshkouei, J. A. and Zinober, A. S. I., Sliding Mode Time-Delay System, *Proceedings of the IEEE International Workshop on Variable Structure Systems*, pp. 97-101, Dec. 1996.

65. Park, J. H. and Cho. H. C., Sliding-Mode Controller for Bilateral Teleoperation with Varying Time Delay, *Proceedings of the ASME/IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 311-316, Sept. 1999.

66. Leung, G. M. H., Francis, B. A., and Apkarian, J., Bilateral Controller for Teleoperators with Time Delay via μ-Synthesis, *IEEE Transactions on Robotics and Automation*, Vol. 11, Issue 1, pp. 105-116, Feb. 1995.

67. www.handshakevr.com, 9/06.

68. Massie, T. H. and Salisbury, J. K., The Phantom Haptic Interface: A Device for Probing Virtual Objects, *ASME Dynamical System and Control*, Vol. 55, Issue 1, pp. 295-299, Nov. 1994.

69. Kontz M., *Haptic Enhancement of Operator Capabilities in Hydraulic Equipment*, Master Thesis, Department of Mechanical Engineering, Georgia Institute of Technology, Nov. 2002.

70. Chipman, R., *Transmission Lines*, New York: McGraw-Hill, 1968.