STABILIZATION OF ASYMMETRIC BILATERAL TELEOPERATION SYSTEMS WITH TIME-VARYING DELAYS

by

Trent Hilliard

Submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2012

DALHOUSIE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "STABILIZATION OF ASYMMETRIC BILATERAL TELEOPERATION SYSTEMS WITH TIME-VARYING DELAYS" by Trent Hilliard in partial fulfillment of the requirements for the degree of Master of Applied Science.

Dated: August 10, 2012

Supervisor: ______________________________

Readers: ______________________________

____

DALHOUSIE UNIVERSITY

DATE: August 10, 2012

AUTHOR: Trent Hilliard

TITLE: STABILIZATION OF ASYMMETRIC BILATERAL TELEOPERATION SYSTEMS WITH TIME-VARYING DELAYS

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE: M.A.Sc. CONVOCATION: October YEAR: 2012

Signature of Author

To my parents, family, and friends
without your encouragement and support
this would not have been possible.

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations Used

PID - Proportional-Integral-Derivative

F_h - Human Input Force

F_e - Environmental Contact Force

x_m - Master States

x_s - Slave States

τ - Communication Time Delay

NCS - Network Control System

$U(t)$ - Control Signal

K_p - Proportional Control Gain

K_i - Integral Control Gain

K_d - Derivative Control Gain

SMC - Sliding Mode Control

LMI - Linear Matrix Inequality

$U_m(t)$ - Master Control Signal

$U_s(t)$ - Slave Control Signal

m_m - Master Mass

k_m - Master Stiffness

b_m - Master Damping

A_m,B_m - Master State Space Matrices

m_s - Slave Mass

k_s - Slave Stiffness

b_s - Slave Damping

A_s,B_s - Slave State Space Matrices

B_e - Environmental Damping

K_e - Environmental Stiffness

A_M,B_M - Master Controller State Space Matrices

M - Master Controller Mass

K - Master Controller Stiffness

B - Master Controller Damping

τ_s - Slave to Master Time Delay

e(t) - State Error

τ_m - Master to Slave Time Delay

K_c - Control Gain

V - Lyapunov Function

P,Q_1,Q_2,R_1,R_2,S - Positive Symmetric Definite Matrices

D - Disturbance Matrix

τ_1 - Minimum Time Delay

τ_2 - Maximum Time Delay

γ_1,γ_2 - Cross Term Isolation Constants

ERA/OKiD - Observer Kalman Filter Identification

DAQ - Data Acquisition

A/D - Analog to Digital Conversion

Abstract

A novel control scheme for asymmetric bilateral teleoperation systems is developed based on linear models of the hardware, with two scenarios considered: i) only an upper bound on the time delay and ii) an upper and lower bounds on the time delay. Lyapunov based methods are used with linear matrix inequalities to prove that the system error is bounded between the manipulators. To ensure stability, a master side impedance matching controller is used.

Simulations were conducted using the Matlab and Simulink platform to run the LMI code and simulate the system. Experiments were then conducted using actual hardware to verify the results, with deviations from simulation results. The variations were due to non-linearities in the hardware and model parameter approximation errors. Finally, suggestions for future work are made.

Acknowledgements

I would like to thank my supervisor Dr. Ya-Jun Pan of the Mechanical Engineering Department for her guidance and support. She has spent many hours developing this work with me, and it would not be possible without her guidance. I would also like to thank Dr. Robert Bauer and Dr. Jason Gu for their role on my supervisory committee, Peter Jones, Mark MacDonald, and Jon MacDonald for their technical assistance, and the Advanced Controls and Mechatronics research group for their suggestions and comments over the years (Balu, Xiang, and Leslie). A special thanks to Dr. Rishad Irani for his help with the system identification code. I would also like to thank the Natural Sciences and Engineering Research Council (NSERC) for their financial support. Last, but certainly not least, thank you to my family for all their love and support.

Chapter 1

Introduction

The following sections outline the background information for the research work, including an overview of bilateral teleoperation and the many control strategies available for use with bilateral teleoperation schemes.

1.1 Teleoperation

Teleoperation, also known as unilateral teleoperation, is the control of a remote system with a local system where there is no feedback from the remote system to the local system. An example of such a scheme is a remote control vehicle, where master hardware (controller) signals for direction and velocity are sent to slave hardware (vehicle), with a block diagram representation in Fig. 1.1. The slave system does not send any type of signal back to the master side to update its state, the only cues the user has to go on are what they can physically see and hear. The same technology can also be applied to robotic arms and manipulators, where a controller of some sort (either an identical arm or control box) is used to control the motion of the remote arm. While excellent position control can be achieved through methods such as Proportional-Integral-Derivative control [31], many tasks can require more than position tracking. Specific information such as forces and torques required for assembly of components in places where humans can't safely work (such as outer space and radioactive waste disposal), level of traction on a remote vehicle, or contact forces on remote objects may be necessary for research or to complete a task. Due to the limited amount of information provided from the slave system and the desire to improve performance of systems where extra information is important, research has shifted to bilateral teleoperation.

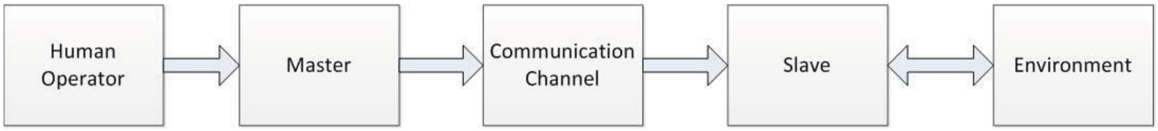


Figure 1.1: Unilateral Teleoperation System Layout

1.2 Bilateral Teleoperation

In contrast to normal teleoperation, bilateral teleoperation has communication from master to slave as well as communication from slave to master. The slave to master communication can be state variables [39], contact forces [32], or any other information that can be used to add an element of feel to the system. In [2], it has been shown that operator's have improved performance (shorter operating time for the same task) when provided with stable force feedback compared to only having audio and visual cues to perform the task. Unfortunately, having both forwards and backwards communication in the system adds levels of instability to the system as the communication itself can become unstable. A class of control law known as passivity control deals with this phenomenon and are discussed in Section 1.3.4. A typical teleoperation system block diagram can be found in Fig. 1.2, where f_h represents the human input, f_e the environmental input, x_m master states, x_s slave states, and τ communication delays.

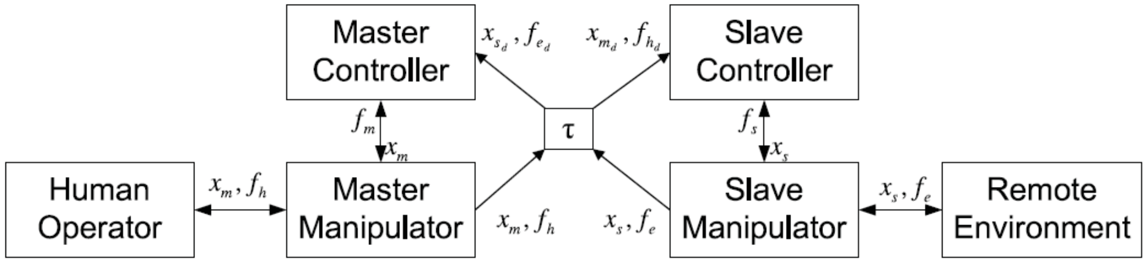


Figure 1.2: Bilateral Teleoperation System Layout [33] ©IEEE 1989

While bilateral teleoperation has added challenges in terms of gathering information from the slave side and added communication channels to the system, it provides necessary information for completion of specific tasks. Some current areas of application within bilateral teleoperation systems include the medical industry [8], dental industry [34], outer space [4], and waste removal [35]. In the medical and dental industries, teleoperation schemes are used on virtual patients [34] as a method of training students in these fields while reducing the risks to actual patients. An example from the medical industry is the da Vinci surgery system, as shown in Fig. 1.3.

Current system performance in multiple hardware setups limits the type of medical procedures able to be performed, as there is often a tradeoff between performance (level of position tracking and feedback) and stability (safe range of operation) based on the level of time delay in the system.

A second major platform that uses bilateral teleoperation is outer space exploration

Figure 1.3: da Vinci Surgical System [36]

and maintenance. Due to the hazards of outer space, it is desirable for robotics to be used to perform maintenance work when possible on platforms such as the International Space Station, and for exploration of remote celestial bodies such as the moon or Mars [37].

A common system that is considered similar to a bilateral teleoperation system is that of a network control system (NCS). The typical layout for a network control system is presented in Fig. 1.4, where communication delays and constraints are imposed by the network infrastructure. The dynamic modeling of a NCS can be as follows [53]

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + A_i\mathbf{x}(t-\tau_i) + A_j\mathbf{x}(t-\tau_j) + B\mathbf{u}(t) \\
\mathbf{y}(t) &= C\mathbf{x}(t) \\
\mathbf{x}(t) &= \phi(t), t\epsilon[-\tau_{max}, 0]
\end{aligned}$$

where the delay occurs on the state information as it passes through the network (τ_i and τ_j), but not on the control input $u(t)$. The function $\phi(t)$ represents the state behavior prior to motion as it is often required for controllers upon startup.

However, in a bilateral teleoperation system, the delay is often in the control input $u(t)$, as the control input is based on the states of the other manipulator, or other property that needs to be sent via network, and thus causes a delay in the control signal. The dynamic equations for a single side of bilateral teleoperation can be

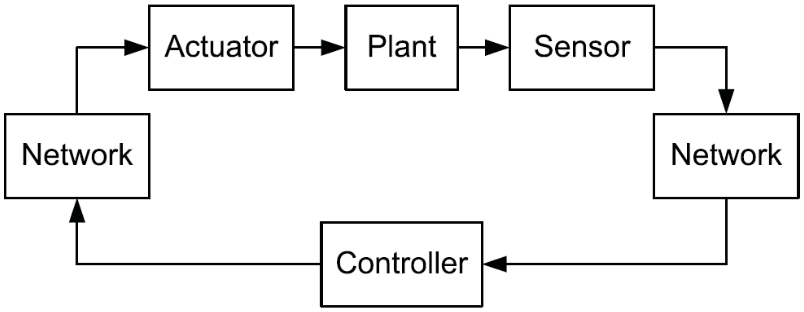


Figure 1.4: Typical Network Control System Layout

realized as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t-\tau) \\ \mathbf{y}(t) &= C\mathbf{x}(t) \\ \mathbf{x}(t) &= \phi(t), t\epsilon[-\tau_{max}, 0].\end{aligned}$$

Due to the differing nature of the delays, exportation of NCS control theory to bilateral teleoperation cannot directly happen. Therefore further work is needed to test if the varying theories can be used in bilateral teleoperation settings. The second major difference is that in a bilateral teleoperation system, two controllers are required (master and slave) compared to the single controller needed for a NCS.

1.3 Literature Review

In order to design controllers for a bilateral teleoperation system, it was necessary to review current control methodologies to determine thier strengths and weaknesses, along with if they are applicable to a teleoperation setup. A literature review was also necessary to help ensure novelty in the work being conducted, and to help with modeling choices and troubleshooting problems. While most of the following sections focus on control strategies, a broader overview of bilateral teleoperation specific control can be found in [14].

1.3.1 PID Control

Proportional integral derivative (PID) control is a control strategy that applies gains on the system error. System error is typically the desired or reference position minus the actual position. A PID controller consists of 3 main gains: K_p as the proportional gain on the error, K_i and the integral gain on the error, and K_d as the derivative gain on the error. The typical transfer function of a PID controller is as follows [13], [15]:

$$\begin{aligned} K(s) &= K_p + \frac{K_i}{s} + K_d s \\ K(s) &= K_p(1 + \frac{1}{T_i s} + T_d s) \end{aligned}$$

There are several methods for tuning a PID controller discussed in [13], but other methods also exist. It should be noted that other forms of controllers based on the PID principle exist, namely P, PI, and PD controllers. Each parameter of a PID or subsequent controller has a different effect on the controller performance. For example, a strictly P controller can improve its response time by increasing the gain K_p, but at the expense of increasing oscillatory motion and the peaks of the motion. Another important feature of pure proportional control is the existence of steady state error which is equal to $1/K_p$. In order to overcome the deficiencies of pure proportional control, integral and derivative terms are added to reduce the overshoots and eliminate the steady state error. More specifically, K_i is used to reach the steady state value and overcome the steady state error of purely proportional control. However, K_i can add to the oscillatory motion if not correctly set. The K_d term is used to limit the rate of change of the error, as to provide a smoother signal with less and smaller overshoots. As stated, the goals of the various gains compete with one another (speed of response for K_p and K_i versus rate limiting of K_d) so it is necessary to find the right balance for the specific application. Table 1.1 [15] provides a reference for closed loop behavior when independently tuning the PID parameters.

Table 1.1: Effects of independent K_p, K_i and K_d tuning [15]

Closed Loop Response	Rise Time	Overshoot	Settling Time	Steady State Error	Stability
Increasing K_p	Decrease	Increase	Small increase	Decrease	Degrade
Increasing K_i	Small decrease	Increase	Increase	Large decrease	Degrade
Increasing K_d	Small decrease	Decrease	Decrease	Minor change	Improve

In terms of bilateral teleoperation, PID controllers and their derivatives have been used in cases discussed in [16], [17], and [18]. In [16] a PD controller based solution is presented based on the assumption that the human and the environment are passive, the time delay is constant, and a joint space manipulator model is used. The other main assumption is that the input force is related to the positional error, in that the controller is trying to correct the error and the input is proportional to the error with a damping injection term added for stability. The work in [17] is an extension of [16], where the same joint based model is used along with the assumption of constant time delay, and again damping injection terms are added to maintain the passivity necessary to fit within the framework. The tracking performance in [17] is improved over [16], as is to be expected with the introduction of the integral term in the controller. [18] presents a PID approach with sliding mode control on a system with no delays, with the work more focused on dealing with modeling inaccuracies than time delay issues. One common trend in the work of PID controllers for bilateral teleoperation is the assumption of constant time delay, a factor that may not be realized on many lower quality networks such as the internet.

1.3.2 Sliding Mode Control

Sliding mode control (SMC) is a class of variable structure control used to deal with non-linear systems. Consider a second order system [19]

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= h(x) + g(x)u\end{aligned}$$

where $h(x)$ and $g(x)$ are unknown nonlinear functions and $g > g_0 > 0$ for all x. The idea is to develop a control law to keep the system on what is called the "sliding surface", and have the system reach this surface in finite time. In order to achieve these goals the controller is typically developed in two stages: the controller while on the surface, and the controller to get the system to the surface. A detailed proof along with a numerical example can be found in [19]. There are two main advantages for using sliding mode control(Pan, "Nonlinear Control Notes",unpublished): 1) The systems have robustness against a large class of model uncertainties, which enter in the same channel as the control inputs and 2) Less information is needed compared to classical control techniques.

Sliding mode control has been used in teleoperation systems in [20], [21], and [22]. In [20], a discrete time controller is developed for the scaled control of a microelectromechanical system subject to model uncertainty in parameters. Scaling is required between the master equipment and the micro sized slave hardware being controlled. The work does not consider time delays, as the focus is on the ability to overcome modeling issues between the micro slave and master hardware. The work of [21] deals with SMC under time varying delays for linear single degree of freedom manipulators. It uses and impedance matching controller on the master side and combines the principles of SMC with an impedance matching controller on the slave side to counteract the unknown time delay properties. In [22], another linear single degree of freedom system is considered with only mass for a property, modal decomposition, and no regard for time delay.

Most works do not deal with time delays due to added complexity in the stability proofs when the instantaneous feedback cannot be used.

1.3.3 H∞ Control

A commonly used class of robust controllers are known as H∞ controllers. In the case of H∞ controllers, the objective is to minimize the H∞-norm of the transfer function of the system $T_{zw}(s)$. A depiction of a general H∞ setup is shown in Figure 1.5 where $\mathbf{w}$ represents the uncontrolled input to the system, $\mathbf{z}$ the output to control, $\mathbf{y}$ the system measurements and $\mathbf{u}$ the controlled input. The dynamics of the system can be represented as (Pan, "Nonlinear Control Notes",unpublished)

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} + E\mathbf{w}\\ \mathbf{y} &= C_1\mathbf{x} + 0u + D_1\mathbf{w}\\ \mathbf{z} &= C_2\mathbf{x} + D_2\mathbf{u} + 0\mathbf{w}\\ \dot{\mathbf{v}} &= A_c\mathbf{v} + B_c\mathbf{y}\\ \mathbf{u} &= C_c\mathbf{v} + D_c\mathbf{y}\end{aligned}$$

where $\mathbf{x}$ is the plant states, and $\mathbf{v}$ the controller states. From the above relationships a generalized transfer function $T_{zw}(s) = C_{cl}(sI - A_{cl})^{-1}B_{cl} + D_{cl}$ can be realized where

$$\begin{aligned}A_{cl} &= \begin{bmatrix} A + BD_cC_1 & BC_c \\ B_cC_1 & A_c \end{bmatrix}\\ B_{cl} &= \begin{bmatrix} E + BD_cD_1 \\ B_cD_1 \end{bmatrix}\\ C_{cl} &= \begin{bmatrix} C_1 + D_2D_cC_1 & D_2C_c \end{bmatrix}\\ D_{cl} &= \begin{bmatrix} D_2D_cD_1 \end{bmatrix}\end{aligned}$$

for which the internal stability is achieved only when the eigenvalues of A_{cl} are in the left hand side of the s-plane (Pan, "Nonlinear Control Notes",unpublished). This criteria can be achieved as the matrices A_c, B_c, C_c and D_c are up to the discretion of the control system designer and can be appropriately tuned. The next step is to define the H∞-norm of the transfer function from $\mathbf{w}$ to $\mathbf{z}$ as that is what is to be minimized.

$$||T_{zw}||_\infty = \sup_{0\leq\omega\leq\infty} \sigma_{max}[T_{zw}(j\omega)]$$

With general theory covered, some examples that use the H∞ approach are found in [23]- [26]. In [23], the H∞ approach is used for sampled data measurements on

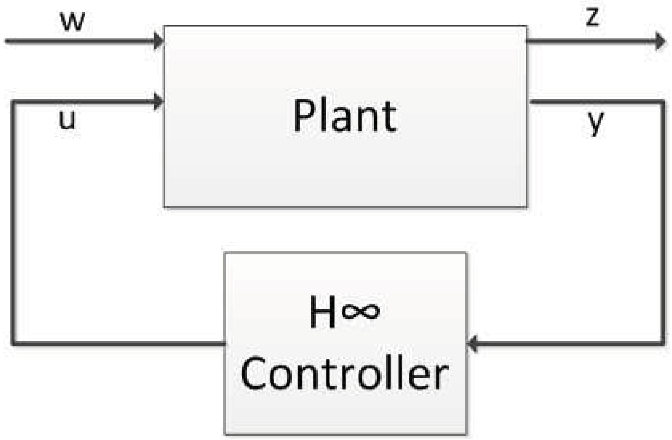


Figure 1.5: General H∞ control system layout

an active suspension for a car when using a quarter car model. The main element that they wish to minimize using the H∞ control is the body mass acceleration of the car, as to have a smoother ride. Lyapunov based criterion and linear matrix inequality (LMI) approaches are applied as part of the techniques to minimize the control objective. The work of [24] considers a discrete time based model of a NCS, for which delays and data dropout exist in the communication channels. The controller for this case is used to minimize the effects of delay and data dropout, as they are the main factors that can cause system degradation if the models are known. In [25], the H∞ approach is used as a method to deal with model uncertainties in a stochastic system, but without any time delay effects. Again, Lyapunov based LMIs are used to solve the objective minimization problem. In [26], the H∞ approach is applied to a linear system with uncertainties in the A and C matrices and a constant state delay. The controller is used as a state estimation device due to the uncertainties involved in the system parameters to help correct the parameters based on the bounds of fluctuation of the parameters in the system. As discussed, the H∞ approach has a broad range of uses, but the challenge with the setup is to properly construct the problem that the objective is measurable and can be minimized, along with the potential mathematical challenges to reach a form that can be solved, either with LMIs [23], [25] or other methods such as the Riccardi equation [26].

1.3.4 Passivity Control

Passivity based control is a method of control where the two power variables passed between the master and slave (typically force and velocity for mechanical systems) are monitored to ensure no increase in power level exists. If an increase in power

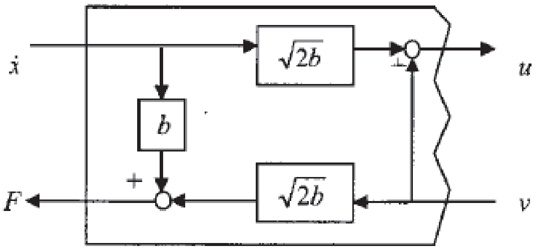


Figure 1.6: Wave variable flow [4] ©IEEE 2004

is noticed within the communication channel, energy is then dissipated through the control algorithm so that the communication channel remains passive. In order to apply this control method, one needs to know that all other elements of the system are passive, so that the only point in the system where passivity would be breached is the communication channel, and thus the control algorithm to eliminate the additional energy can be used, see [9], [10]. One issue with passivity based controllers is that while stable performance of the system is guaranteed, position tracking cannot be guaranteed without the addition of other control techniques, such as in [10].

An extension of passivity control is the wave variable method of teleoperation control [12]. Two new variables, u and v, are computed based on the original power flow variables (typically force and velocity), with u representing flow in one direction, while v represents flow in the opposite direction. The general definition of power flow is:

$$P = \dot{x}^T F = \frac{1}{2} u^T \cdot u - \frac{1}{2} v^T \cdot v$$

A diagram showing this layout can be seen as Fig. 1.6 courtesy of [4]. The parameter b, known as the characteristic wave impedance, is used to relate the actual variables to the flow variables through

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

The tuning of b can lead to either a faster responding system with lower force feedback, or a slower system with a higher level of force feedback. Depending upon the application, one method may be more desirable than another, and should be considered when working with the wave variable approach. Other examples using the wave variable approach can be found in [7].

An extension of passivity control is the idea of time based passivity control as introduced in [11], with more work on the subject presented in [9]. A further extension of the time domain passivity control is power based passivity control, with a teleoperation example given in [3].

1.3.5 Predictive Control

Predictive control approach to system control is typically applied to systems where the dynamics of the slave working in the environment are well known, and time delays can be quite large. By providing the operator with predictive responses from the slave, the overall task completion time can be reduced, as the operator need not wait for the response from the slave, which may take several minutes depending on distance and communication medium. A commonly used predictor model is known as the Smith predictor [5], where prior knowledge of the slave mechanics are used to help reduce the effects of time delay on the system. Other examples of predictive control for teleoperation can be found in [6] .

1.3.6 Lyapunov LMI-based State Feedback

As the title suggests, a state feedback controller uses the system states as the main variable of interest in the controller design. State feedback can exist in the form of a constant gain feedback, such as PID or other stabilization methods [27], or variable gain feedback based on state values such as fuzzy logic controllers [28], [29]. Many constant gain feedback systems are the results of LMI based solutions such as in [41] and [42], where Lyapunov functionals are proposed to deal with different types of network behavior such as weighted delay distributions [42] and interval time varying delay with system non-linearities [41]. LMI problems are formulated based on the modeling choices and solved to provide a static feedback control gain. Other works have covered these types of problems such as [30], [39].

1.4 Research Contributions

The work in this thesis contains the following contributions:

1. Formulates the bilateral teleoperation problem into a form similar to a network control system (NCS) problem with the use of a master side impedance controller and slave side stabilization by using Lyapunov based linear matrix inequalities.

2. Allows for asymmetrical teleoperation systems in terms of system parameters and range of motion characteristics. Based on the current work, as long as all asymmetrical terms are constant and lie within the same channel as the control signal (in this case

the velocity channel), the algorithm can account for the differences and adjust the slave side control law accordingly.

3. Derives a novel static state feedback control gain, where the state is the system error once the problem is reformulated as a NCS where the error is the state vector. A Lyapunov functional based on [42] is considered for the base of the proof. Additional disturbances, such as the human user input and environmental wall contact, are added. These terms appear in the error bound that may not exist in standard NCS dynamic structures (such as [42]).

4. A conference paper based on Case I (only an upper delay bound) has been accepted at ICIRA 2012 for presentation in October.

Chapter 2

System Modeling

In order to design a control law for a teleoperation system, it is important to determine the type of models to be used in the system, along with the general sign convention to be used. Fig. 2.1 shows the assumed directions of the forces acting on the system, with positive to the right. The following sections outline the modeling choices for the manipulators (linear single degree of freedom, similar to [47], [48], [12]), the human input force possibilities, the environmental force in the simulation case, and the communication channel behavior (internet based model without data dropout). Other modeling choices exist for the manipulators such as Euler-Lagrange [51], joint based models for linked manipulators [52], [58] and other non-linear methods.

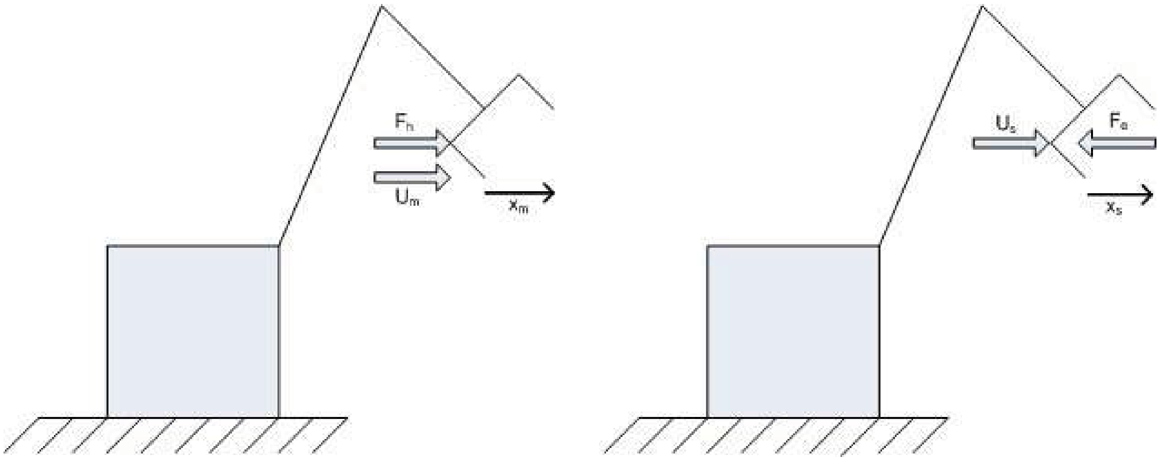


Figure 2.1: Diagram of forces and their assumed directions on 1-DOF manipulators

2.1 Master Side General Dynamics

Consider a single degree of freedom manipulator with linear properties of mass, stiffness, and damping. The general dynamic equation representing the system is

$$m_m \ddot{x}_m + b_m \dot{x}_m + k_m x_m = F_h + U_m \tag{2.1}$$

where m_m represents the mass of the system, b_m the damping, k_m the stiffness, x_m the position, F_h the human input force and U_m the control signal to the system. The system can be represented in the state-space form

$$\dot{\mathbf{x}}_m = A_m\mathbf{x}_m + B_m(F_h + U_m)$$

where

$$A_m = \begin{bmatrix} 0 & 1 \\ -k_m/m_m & -b_m/m_m \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ 1/m_m \end{bmatrix}.$$

2.2 Slave Side General Dynamics

Similar to the master dynamics, we consider the slave manipulator to be modeled as a single degree of freedom manipulator with mass, stiffness, and damping properties. The dynamic equation governing the system is:

$$m_s\ddot{x}_s + b_s\dot{x}_s + k_s x_s = U_s - F_e, \tag{2.2}$$

where m_s represents the mass of the system, b_s the damping, k_s the stiffness, x_s the position, F_e the environmental contact force and U_s the control signal to the system. The system can be represented in the state-space as:

$$\dot{\mathbf{x}}_s = A_s\mathbf{x}_s + B_s(U_s - F_e),$$

where

$$A_s = \begin{bmatrix} 0 & 1 \\ -k_s/m_s & -b_s/m_s \end{bmatrix}, \quad B_s = \begin{bmatrix} 0 \\ 1/m_s \end{bmatrix}.$$

2.3 Environmental Force

For simulation work it is necessary to model the expected environmental forces that would act upon the slave manipulator in an experimental setting. For this work, the chosen scenario was when the slave manipulator is positioned against a non-rigid surface such as foam. In this scenario motion into the foam would cause a reactionary force, and motion away from the foam would cause no environmental force (air drag is considered minimal at low speeds and thus negligible). The foam is modeled as a spring and damper system based on the position of the slave manipulator with coefficients of B_e and K_e as follows:

$$F_e = \begin{cases} B_e\dot{x}_s + K_e x_s & : x_s \geq 0 \\ 0 & : x_s < 0. \end{cases}$$

This approach has been previously used in [53] , and examples of filtering by frequency can be found in [54].

2.4 Human User Force

For the simulation environment it is necessary to model the various types of human user inputs possible for the system. The first input considered is a modified step input, as a human operator cannot physically apply a true step input since human motion is frequency limited [55]. To build the source, a ramp function is used to reach the step value and then the value is held constant. A second source that can be used an input to the system is a sinusoidal input, where once again the frequency is limited to the band of human motion. For reference and shown later in Section 4, the modified step input uses a slope of 0.2 N/s with a peak value of 1 second, while the sinusoidal input has an amplitude of 5 N and a period of 12 seconds.

2.5 Communication Channels

In order to develop appropriate theory, some assumptions on the nature of the communication channels are necessary. The channels in this work are treated as separate entities (asymmetric delays) in two different cases: one with just an upper bound on the channel variable time delay, and one with both an upper and lower bound on the channel variable time delay. In both scenarios the upper limit is set at 0.08 seconds and the lower limit at 0.02 seconds based on trans-continental internet delays [53]. While other works deal with phenomenon such as packet-loss ([45], [53], [56], [57]) and mode dependent time delays [50], this work only focuses on delays within a range, and future work could be done to incorporate the packet loss phenomenon. The delays are modeled as a random number between the upper and lower bounds to represent the random nature of delays when working on network infrastructure.

2.6 Summary

The linear single degree of freedom models used for both the master and slave manipulators are presented along with the modeling choice for the human and environmental forces for the simulation cases. Communication channel constraints are discussed and the assumptions that the channels will have a time varying delay are outlined.

Chapter 3

Controller Design

The design of the master and slave side control signals will be discussed in this chapter along with the stability proof based on the modeling choices outlined in Chapter 2. Two separate cases are considered for the stability proof: only an upper bound on the time delay, and both an upper and lower bound on the time delay.

3.1 Master Side Controller Design

For the master side controller an impedance matching approach is applied similar to work in [49] such that the master manipulator will have the same impedance as a designed system. A reference system would have the form

$$M\ddot{x}_m + B\dot{x}_m + Kx_m = F_h - F_e(t - \tau_s(t)), \tag{3.1}$$

where M, B and K represent the mass, damping and stiffness of the desired impedance system respectively. The desired impedance system can be represented in the state space form as

$$\dot{\mathbf{x}}_m = A_M\mathbf{x}_m + B_M(F_h - F_e(t - \tau_s(t))), \tag{3.2}$$

where

$$A_M = \begin{bmatrix} 0 & 1 \\ -K/M & -B/M \end{bmatrix}, \quad B_M = \begin{bmatrix} 0 \\ 1/M \end{bmatrix}.$$

For reference, the master system is modeled as

$$m_m\ddot{x}_m + B_M\dot{x}_m + k_mx_m = F_h + U_m. \tag{3.3}$$

The impedance matching approach is used to allow for a constant system control gain K_c to be designed later on to only have an impact on the slave side dynamics. By allowing the user to control the master side impedance dynamics through the choice of B, K, and M, the level of stability in the system can be tuned to match what is necessitated by the application. The final master controller is designed as

$$U_m = (B_M - \frac{m_m}{M}B)\dot{x}_m + (\frac{m_m}{M} - 1)F_h - \frac{m_m}{M}F_e(t - \tau_s(t)) + (k_m - \frac{m_m}{M}K)x_m, \tag{3.4}$$

which can realize the desired impedance model in (3.2) when substituted into (3.3).

3.2 Slave Side Controller Design

The following section deals with the design of the slave side controller by first introducing the error dynamics as required by the stability proof in Section 3.3 to allow for a method to deal with the asymmetrical terms. The control law is then derived to allow for positional tracking and to provide bounds to the external forces F_e and F_h.

3.2.1 Error Dynamics

It is necessary to derive the error dynamics of the system in order to properly substitute into Lyapunov candidate derivative terms in (3.10) and design the system control signal U_s. For most teleoperation applications, it is desirable for the slave to track the position of the master in real time i.e. $\mathbf{e}(t) = \mathbf{x}_s(t) - \gamma\mathbf{x}_m(t) + C$, where γ is a scaling factor and C an offset constant. While other choices exist for the error dynamics (such as $\mathbf{e}(t) = \mathbf{x}_s(t) - \gamma\mathbf{x}_m(t-\tau_m(t)) + C$ [44]) this work uses the definition of $\mathbf{e}(t) = \mathbf{x}_s(t) - \gamma\mathbf{x}_m(t) + C$. Hence

$$\begin{aligned}
\dot{\mathbf{e}} &= \dot{\mathbf{x}}_s - \gamma\dot{\mathbf{x}}_m \\
&= A_s\mathbf{x}_s + B_s(U_s - F_e) - \gamma[A_M\mathbf{x}_m + B_M(F_h - F_e(t-\tau_s(t))] \\
&= A_s\mathbf{x}_s + B_s(U_s - F_e) - \gamma[A_M\mathbf{x}_m + B_M(F_h - F_e(t-\tau_s(t))] \\
&\quad + A_M\mathbf{x}_s - A_M\mathbf{x}_s \\
&= A_M\mathbf{e} + (A_s - A_M)\mathbf{x}_s + \gamma B_M F_e(t-\tau_s(t)) - \gamma B_M F_h + B_s U_s - B_s F_e \qquad (3.5)
\end{aligned}$$

where $\tau_s(t)$ represents the time delay in communication from the slave side to the master side, and $\tau_m(t)$ represents the delay from the master side to slave side.

3.2.2 Controller Design

From (3.5), the term $(A_s - A_M)x_s$ needs to be eliminated using the slave side control law. The controller also needs to provide a method for bounding the effects of F_h and F_e. The term $(A_s - A_M)$ can be represented as:

$$A_s - A_M = \begin{bmatrix} 0 & 0 \\ \delta_1 & \delta_2 \end{bmatrix}, \quad \delta_1 = K/M - k_s/m_s, \quad \delta_2 = B/M - b_s/m_s \qquad (3.6)$$

where all the terms lie is the same channel the control input U_s, in the $\dot{x}_{2s}$ channel, which means that a portion of the U_s can be used to eliminate it.

Now design a control signal of the form

$$\begin{aligned} U_s &= -m_s[\delta_1 x_{1s} + \delta_2 x_{2s}] + K_c(\mathbf{x}_s(t - \tau_m(t)) - \mathbf{x}_m(t - \tau_m(t))) \\ &+ \alpha F_h(t - \tau_m(t)) - \beta F_e(t) + F_e(t) \end{aligned} \tag{3.7}$$

where the first term $(-m_s[\delta_1 x_{1s} + \delta_2 x_{2s}])$ of the control input U_s is to deal with the asymmetry of the system as represented in (3.6). The second term $(K_c(\mathbf{x}_s(t-\tau_m(t))-\mathbf{x}_m(t-\tau_1(t))))$ is the controller input on the delayed error, as the current value of the master states (x_m) is not available to the slave side. With an ability to record states and send time stamps, the delayed value of the slave states can be made available to allow for the delayed error to be realized. The remaining terms $(\alpha F_h(t - \tau_m(t)) - \beta F_e(t) + F_e(t))$ are used to provide a bound to the external disturbances F_h and F_e. Substituting (3.7) into (3.5) yields

$$\begin{aligned} \dot{\mathbf{e}} &= A_M\mathbf{e} + B_M F_e(t - \tau_s(t)) - B_M F_h(t) \\ &\quad +B_s[K_c(\mathbf{x}_s(t - \tau_m(t)) - \mathbf{x}_m(t - \tau_m(t))) - \alpha F_h(t - \tau_m(t)) \\ &\quad -\beta F_e(t)] + B_s F_e(t) - B_s F_e(t) \\ &= A_M\mathbf{e} + B_M F_e(t - \tau_s(t)) - B_M F_h(t) + B_s[K_c\mathbf{e}(t - \tau_m(t)) \\ &\quad -\alpha F_h(t - \tau_m(t)) - \beta F_e(t)] \\ &= A_M\mathbf{e} + B_M F_e(t - \tau_s(t)) - \beta B_s F_e(t) + \alpha B_s F_h(t - \tau_m(t)) \\ &\quad -B_M F_h(t) + B_s K_c\mathbf{e}(t - \tau_m(t)). \end{aligned}$$

From the error dynamics, design an $\alpha = \gamma m_s/M$ to combine the $\alpha B_s F_h(t - \tau_m(t)) - \gamma B_M F_h(t)$ terms from the error dynamics to a single difference term. Similarly, to bound $\gamma B_M F_e(t - \tau_s(t)) - \beta B_s F_e(t)$ to a single difference term, select $\beta = \gamma m_s/M$. Then the error dynamics becomes

$$\begin{aligned} \dot{\mathbf{e}} &= A_M\mathbf{e} + B_s K_c\mathbf{e}(t - \tau_m(t)) \\ &\quad +\gamma B_M[F_h(t - \tau_m(t)) - F_h(t)] + \gamma B_M[F_e(t - \tau_s(t)) - F_e(t)] \\ &= A_M\mathbf{e} + B_s K_c\mathbf{e}(t - \tau_m(t)) + B_M[d_1 + d_2] \\ &= A_M\mathbf{e} + B_s K_c\mathbf{e}(t - \tau(t)) + D \end{aligned} \tag{3.8}$$

where $D = B_M[d_1+d_2]$ with $d_1 = \gamma(F_h(t-\tau_m(t))-F_h(t))$, $d_2 = \gamma(F_e(t-\tau_s(t))-F_e(t))$, and for simplicity $\tau(t) = \tau_m(t)$.

3.3 Stability Analysis

The stability analysis is conducted for two separate communication channel cases: i) when only an upper bound exists on the delay, and ii) when there is both an upper and lower bound on the delay. Lyapunov based linear matrix inequalities (LMIs) are used to prove stability while providing error bounds. LMIs are a common tool used for teleoperation system stability such as in [46].

3.3.1 Case I: Upper Delay Bound Only

Theorem 1. *For given constant τ_2, if there exist matrices $X > 0$, $\hat{Q} > 0$, $\hat{R} > 0$ and Y with appropriate dimensions, such that the following LMI*

$$\begin{bmatrix} -\hat{Q}-\hat{R} & 0 & 0 & 0 & \hat{R} & 0 & 0 \\ * & \hat{Q}+A_MX+XA_M^T-\hat{R} & I & 0 & \hat{R}+B_sY & XA_M^T & \tau_2XA_M^T \\ * & * & -\gamma_1 I & X & 0 & 0 & 0 \\ * & * & * & -I & 0 & 0 & 0 \\ * & * & * & * & -2\hat{R} & Y^TB_s^T & \tau_2Y^TB_s^T \\ * & * & * & * & * & -\gamma_2 I & 0 \\ * & * & * & * & * & * & -X\hat{R}^{-1}X \end{bmatrix} < 0 \qquad (3.9)$$

holds, then systems (3.1) and (2.2) are asymptotically stable.

Proof: Choose a Lyapunov Candidate of the following form similar as in [41] and [42]

$$\begin{aligned} V &= V_1 + V_2 + V_3 \qquad (3.10) \\ V_1 &= \mathbf{e}(t)^TP\mathbf{e}(t) \\ V_2 &= \int_{t-\tau_2}^{t} \mathbf{e}^T(s)Q\mathbf{e}(s)ds \\ V_3 &= \int_{-\tau_2}^{0}\int_{t+s}^{t} \dot{\mathbf{e}}^T(v)\tau_2R\dot{\mathbf{e}}(v)dvds, \end{aligned}$$

where τ_2 is the maximum level of time delay of the time varying delay $\tau(t)$, i.e. $\tau_m(t)$. It is also noted that $P, Q, R > 0$. The time derivatives of each function are

$$\begin{aligned} \dot{V_1} &= \mathbf{e}(t)^TP\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}(t)^TP\mathbf{e}(t) \\ \dot{V_2} &= \mathbf{e}^T(t)Q\mathbf{e}(t) - \mathbf{e}^T(t-\tau_2)Q\mathbf{e}(t-\tau_2) \\ \dot{V_3} &= \dot{\mathbf{e}}^T(t)\tau_2^2R\dot{\mathbf{e}}(t) - \int_{t-\tau_2}^{t} \dot{\mathbf{e}}^T(v)\tau_2R\dot{\mathbf{e}}(v)dv. \end{aligned}$$

From $\dot{V}$ we have an integral term of $-\int_{t-\tau_2}^{t}\dot{\mathbf{e}}^T(v)\tau_2R\dot{\mathbf{e}}(v)dv$ that needs to be converted to a different form to fit into an LMI-based solution. By applying Jensen's Inequality ([41], [43]) to the term the result is

$$-\int_{t-\tau_2}^{t}\dot{\mathbf{e}}^T(v)\tau_2R_2\dot{\mathbf{e}}(v)dv \leq \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix}^T \begin{bmatrix} -R & R & 0 \\ * & -2R & R \\ * & * & -R \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix}. \quad (3.11)$$

Other methods exist to deal with this term such as adding "zero" terms which can be used to eliminate the integral and leave terms that can be easily worked into a LMI, similar to [38], [39], [45]. Then by substituting (3.8) into the V_1 portion of (3.10),

$$\begin{aligned}
\dot{V} &= \mathbf{e}(t)^T P\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}(t)^T P\mathbf{e}(t) + \mathbf{e}^T(t)Q\mathbf{e}(t) \\
&\quad -\mathbf{e}^T(t-\tau_2)Q\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)\tau_2^2 R\dot{\mathbf{e}}(t) \\
&= \mathbf{e}(t)^T P(A_M\mathbf{e}(t) + B_sK_c\mathbf{e}(t-\tau(t)) + D) \\
&\quad +(A_M\mathbf{e}(t) + B_sK_c\mathbf{e}(t-\tau(t)) + D)^T P(\mathbf{e}(t)) \\
&\quad -\mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)(\tau_2^2 R)\dot{\mathbf{e}}(t) \\
&= \mathbf{e}^T(t)(Q + PA_M + A_M^T P + \gamma_1^{-1}PP)\mathbf{e}(t) + \mathbf{e}^T(t)PB_sK_c\mathbf{e}(t-\tau(t)) \\
&\quad +\mathbf{e}(t-\tau(t))K_c^T B_s^T P\mathbf{e}(t) - \mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) \\
&\quad +\dot{\mathbf{e}}^T(t)(\tau_2^2 R)\dot{\mathbf{e}}(t) + \gamma_1 D^T D
\end{aligned}$$

Ignoring the $\dot{\mathbf{e}}^T(t)(\tau_2^2 R)\dot{\mathbf{e}}(t)$ term temporarily and converting to an LMI form with the previous term from (3.11)

$$\begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix}^T \begin{bmatrix} \Pi_1 & \Pi_2 & 0 \\ * & -2R & R \\ * & * & -Q-R \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix} + \gamma_1 D^T D,$$

where

$$\begin{aligned}
\Pi_1 &= Q + PA_M + A_M^T P + \gamma_1^{-1}PP - R \\
\Pi_2 &= R + PB_sK_c.
\end{aligned}$$

Define $\zeta(t) = \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \end{bmatrix}$, and rewrite the LMI of this portion as

$$\begin{bmatrix} \mathbf{e}(t-\tau_2) \\ \zeta \end{bmatrix}^T \begin{bmatrix} -Q-R & \Pi_4 \\ * & \Pi_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}(t-\tau_2) \\ \zeta \end{bmatrix} + \gamma_1 D^T D, \tag{3.12}$$

where

$$\Pi_3 = \begin{bmatrix} \Pi_1 & \Pi_2 \\ * & -2R \end{bmatrix}, \qquad \Pi_4 = \begin{bmatrix} 0 & R \end{bmatrix}.$$

The next term that needs to be examined is $\dot{\mathbf{e}}^T(t)(\tau_2^2 R)\dot{\mathbf{e}}(t)$. By substituting $\dot{\mathbf{e}}$ into the term and setting $E(t) = A_M\mathbf{e} + B_sK_c\mathbf{e}(t-\tau(t))$, it becomes

$$\begin{aligned}
& [E(t) + D]^T(\tau_2^2 R)[E(t) + D] \\
={}& E^T(t)(\tau_2^2 R)E(t) + E^T(t)(\tau_2^2 R)D + D^T(\tau_2^2 R)E(t) + D^T(\tau_2^2 R)D \\
={}& E^T(t)(\tau_2^2 R + \gamma_2^{-1} I)E(t) + D^T(\tau_2^2 R)D + \gamma_2 D^T(\tau_2^2 R^T R\tau_2^2)D
\end{aligned}$$

Then substituting back in $E(t)$ in a matrix form, the previous term yields

$$\begin{aligned}
&\begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \end{bmatrix}^T \begin{bmatrix} A_M & B_sK_c \end{bmatrix}^T (\tau_2 R \tau_2 + \gamma_2^{-1} I) \begin{bmatrix} A_M & (B_sK_c) \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \end{bmatrix} \\
&+D^T(\tau_2^2 R)D + \gamma_2 D^T(\tau_2^2 R^T R \tau_2^2)D \\
&\begin{bmatrix} \zeta \end{bmatrix}^T \begin{bmatrix} A_M & B_sK_c \end{bmatrix}^T (\tau_2 R \tau_2 + \gamma_2^{-1} I) \begin{bmatrix} A_M & (B_sK_c) \end{bmatrix} \begin{bmatrix} \zeta \end{bmatrix} \\
&+D^T(\tau_2^2 R)D + \gamma_2 D^T(\tau_2^2 R^T R \tau_2^2)D \qquad (3.13)
\end{aligned}$$

Substituting (3.13) into (3.12) results in

$$\begin{bmatrix} -Q-R & \Pi_4 & 0 \\ * & \Pi_3 + \begin{pmatrix} A_M^T \\ (B_sK_c)^T \end{pmatrix} \gamma_2^{-1} I \begin{pmatrix} A_M & (B_sK_c) \end{pmatrix} & \tau_2 \begin{pmatrix} A_M^T \\ (B_sK_c)^T \\ -R^{-1} \end{pmatrix} \\ * & * & \end{bmatrix} < 0$$

Apply the Schur Complement on the term $\Pi_3 + \begin{pmatrix} A_M^T \\ (B_sK_c)^T \end{pmatrix} \gamma_2^{-1} I \begin{pmatrix} A_M & (B_sK_c) \end{pmatrix}$ and expand the rows and columns to achieve

$$\begin{aligned}
\dot{V} \;&\leq \begin{bmatrix} -Q-R & 0 & R & 0 & 0 \\ * & Q + PA_M + A_M^T P + \gamma_1^{-1} PP - R & R + B_sK_c & A_M^T & \tau_2 A_M^T \\ * & * & -2R & Y^T B_s^T & \tau_2 Y^T B_s^T \\ * & * & * & -\gamma_2 & 0 \\ * & * & * & * & -R^{-1} \end{bmatrix} \\
&+ \; \gamma_1 D^T D + D^T(\tau_2^2 R)D + \gamma_2 D^T(\tau_2^2 R^T R \tau_2^2)D \\
&\leq \; \Theta - \mathbf{e}^T\mathbf{e} + D^T(\gamma_1 I + \tau_2^2 R + \gamma_2 \tau_2^2 R^T R \tau_2^2)D
\end{aligned}$$

where

$$\Theta = \begin{bmatrix} -Q-R & 0 & R & 0 & 0 \\ * & Q + PA_M + A_M^T P + \gamma_1^{-1} PP - R + I & R + B_sK_c & A_M^T & \tau_2 A_M^T \\ * & * & -2R & Y^T B_s^T & \tau_2 Y^T B_s^T \\ * & * & * & -\gamma_2 & 0 \\ * & * & * & * & -R^{-1} \end{bmatrix}.$$

If $\Theta < 0$ holds, then

$$\dot{V} \leq \; -\|\mathbf{e}(t)\|^2 + \|D^T(\gamma_1 I + \tau_2^2 R + \gamma_2 \tau_2^2 R^T R \tau_2^2)D\| \qquad (3.14)$$

which shows that the error is bounded and that the bound is determined by the magnitude of the terms γ_1, γ_2, $\|D\|$, the eigenvalues of R, and the upper bound of the delay τ_2.

The next step is to linearize the above system Θ by pre- and post-multiplying by $diag(P^{-1}, P^{-1}, P^{-1}, I, I)$ and then making the following substitutions

$$\begin{aligned} X &= P^{-1} \\ Y &= KX \\ \hat{Q} &= XQX \\ \hat{R} &= XRX, \end{aligned}$$

which results in the following LMI for Θ,

$$\begin{bmatrix} -\hat{Q}-\hat{R} & 0 & \hat{R} & 0 & 0 \\ * & \hat{Q}+A_MX+XA_M^T+\gamma_1^{-1}I-\hat{R}+XIX & \hat{R}+B_sY & XA_M^T & \tau_2XA_M^T \\ * & * & -2\hat{R} & Y^TB_s^T & \tau_2Y^TB_s^T \\ * & * & * & -\gamma_2 & 0 \\ * & * & * & * & -R^{-1} \end{bmatrix} < 0$$

Substituting

$$\hat{R} = XRX, \quad R^{-1} = X\hat{R}^{-1}X,$$

to get

$$\begin{bmatrix} -\hat{Q}-\hat{R} & 0 & \hat{R} & 0 & 0 \\ * & \hat{Q}+A_MX+XA_M^T+\gamma_1^{-1}I-\hat{R}+XIX & \hat{R}+B_sY & XA_M^T & \tau_2XA_M^T \\ * & * & -2\hat{R} & Y^TB_s^T & \tau_2Y^TB_s^T \\ * & * & * & -\gamma_2 & 0 \\ * & * & * & * & -X\hat{R}^{-1}X \end{bmatrix} < 0.$$

We can now apply the Schur Complement twice to isolate the γ_1^{-1} and XIX terms in the LMI thus arriving at a nearly linear LMI that can be coded in Matlab

$$\begin{bmatrix} -\hat{Q}-\hat{R} & 0 & 0 & 0 & \hat{R} & 0 & 0 \\ * & \hat{Q}+A_MX+XA_M^T-\hat{R} & I & 0 & \hat{R}+B_sY & XA_M^T & \tau_2XA_M^T \\ * & * & -\gamma_1I & X & 0 & 0 & 0 \\ * & * & * & -I & 0 & 0 & 0 \\ * & * & * & * & -2\hat{R} & Y^TB_s^T & \tau_2Y^TB_s^T \\ * & * & * & * & * & -\gamma_2I & 0 \\ * & * & * & * & * & * & -X\hat{R}^{-1}X \end{bmatrix} < 0.$$

This completes the proof. ■

It is necessary to find a method to deal with the non-linear term $X\hat{R}^{-1}X$ in (3.9). To do so, replace $X\hat{R}^{-1}X$ with a new variable G, and apply additional constraints to LMI system

$$\begin{bmatrix} -\hat{Q}-\hat{R} & 0 & 0 & 0 & \hat{R} & 0 & 0 \\ * & \hat{Q}+A_MX+XA_M^T-\hat{R} & I & 0 & \hat{R}+B_sY & XA_M^T & \tau_2XA_M^T \\ * & * & -\gamma_1I & X & 0 & 0 & 0 \\ * & * & * & -I & 0 & 0 & 0 \\ * & * & * & * & -2\hat{R} & Y^TB_s^T & \tau_2Y^TB_s^T \\ * & * & * & * & * & -\gamma_2I & 0 \\ * & * & * & * & * & * & -G \end{bmatrix} \tag{3.15}$$

where $G = X\hat{R}^{-1}X$. Define $J = G^{-1}$, $P = X^{-1}$ and $L = R^{-1}$, G can be translated into

$$\begin{bmatrix} J & P \\ * & L \end{bmatrix} \geq 0. \tag{3.16}$$

To help enforce the inverse relationships between variables define the following:

$$\begin{aligned} &\text{Minimize Trace}(XP + JG + RL) \\ &\text{subject to } X > 0, P > 0, G > 0, J > 0, L > 0, R > 0 \\ &\begin{bmatrix} X & I \\ * & P \end{bmatrix} > 0; \begin{bmatrix} G & I \\ * & J \end{bmatrix} > 0; \begin{bmatrix} R & I \\ * & L \end{bmatrix} > 0 \end{aligned} \tag{3.17}$$

The new LMI problem can be solved by applying the cone complementarity algorithm ([40], [41]) in the following manner:
Step 1: Find a set of feasible matrices $(X, Y, G, R, L, Q, P, J)^0$ that satisfies (3.15),(3.16) and (3.17).
Step 2: Solve the following LMI minimization problem:

$$\begin{aligned} &\text{Minimize Trace}(XP^0 + PX^0 + GJ^0 + JG^0 + RL^0 + LR^0) \\ &\text{subject to (3.15),(3.16) and (3.17)} \end{aligned}$$

Step 3: Substitute the new matrix variables from the previous step into (3.9). If the result is feasible, stop and set $K_c = YX^{-1}$. If not feasible, set the newly acquired matrices to $(X, Y, G, R, L, Q, P, J)^0$ and go to step 2.
It may be necessary to set a maximum number of iterations depending upon available computing power, time, and stability of the given system matrices. The algorithm does not guarantee a solution, but any solution will guarantee stability and tracking.

3.3.2 Case II: Upper and Lower Bound on Time Delay

Theorem 2. *For given constants τ_1, τ_2, if there exist matrices $X > 0$, $\hat{Q}_1 > 0$, $\hat{Q}_2 > 0$, $\hat{R}_1 > 0$, $\hat{R}_2 > 0$, $\hat{S} > 0$ and Y with appropriate dimensions, such that the following LMI*

$$\begin{bmatrix} -\hat{S} - \hat{R}_1 - \hat{Q}_1 & 0 & \hat{R}_1 & 0 & 0 & \hat{S} & 0 & 0 & 0 & 0 \\ * & -\hat{S} - \hat{R}_2 - \hat{Q}_2 & 0 & 0 & 0 & \hat{S} + \hat{R}_2 & 0 & 0 & 0 & 0 \\ * & * & \hat{Q}_1 + \hat{Q}_2 + A_M X + XA_M^T - \hat{R}_2 - \hat{R}_1 & X & 0 & \hat{R}_2 + B_s Y & XA_M^T & \tau_1 XA_M^T & \tau_2 XA_M^T & (\tau_2 - \tau_1)XA_M^T \\ * & * & * & -I & I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & -\gamma_1 I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & -2\hat{S} - 2\hat{R}_2 & Y^T B_s^T & \tau_1 Y^T B_s^T & \tau_2 Y^T B_s^T & (\tau_2 - \tau_1)Y^T B_s^T \\ * & * & * & * & * & * & -\gamma_2 I & 0 & 0 & 0 \\ * & * & * & * & * & * & * & -R_1^{-1} & 0 & 0 \\ * & * & * & * & * & * & * & * & -R_2^{-1} & 0 \\ * & * & * & * & * & * & * & * & * & -S^{-1} \end{bmatrix} \tag{3.18}$$

holds, then systems (3.1) and (2.2) are asymptotically stable.

Proof: Consider a Lyapunov Candidate of the following form [41], [42]:

$$\begin{aligned} V &= V_1 + V_2 + V_3 + V_4 \\ V_1 &= \mathbf{e}(t)^T P \mathbf{e}(t) \\ V_2 &= \sum_{i=1}^{2} \int_{t-\tau_i}^{t} \mathbf{e}^T(s) Q_i \mathbf{e}(s) ds \\ V_3 &= \sum_{i=1}^{2} \int_{-\tau_i}^{0} \int_{t+s}^{t} \dot{\mathbf{e}}^T(v) \tau_i R_i \dot{\mathbf{e}}(v) dv ds \\ V_4 &= \int_{-\tau_2}^{-\tau_1} \int_{t+s}^{t} \dot{\mathbf{e}}^T(v)(\tau_2 - \tau_1) S \dot{\mathbf{e}}(v) dv ds \end{aligned}$$

Where τ_1 is the minimum level of time delay and τ_2 is the maximum level of time delay of $\tau(t)$. It is also noted that $R_i, Q_i, S > 0$. For the proof it is necessary to have the time derivative of each function,

$$\begin{aligned} \dot{V_1} &= \mathbf{e}(t)^T P \dot{\mathbf{e}}(t) + \dot{\mathbf{e}}(t)^T P \mathbf{e}(t) \\ \dot{V_2} &= \sum_{i=1}^{2} (\mathbf{e}^T(t) Q_i \mathbf{e}(t) - \mathbf{e}^T(t-\tau_i) Q_i \mathbf{e}(t-\tau_i)) \\ &= \mathbf{e}^T(t) Q_1 \mathbf{e}(t) - \mathbf{e}^T(t-\tau_1) Q_1 \mathbf{e}(t-\tau_1) + \mathbf{e}^T(t) Q_2 \mathbf{e}(t) - \mathbf{e}^T(t-\tau_2) Q_2 \mathbf{e}(t-\tau_2) \\ \dot{V_3} &= \sum_{i=1}^{2} (\dot{\mathbf{e}}^T(t) \tau_i^2 R_i \dot{\mathbf{e}}(t) - \int_{t-\tau_i}^{t} \dot{\mathbf{e}}^T(v) \tau_i R_i \dot{\mathbf{e}}(v) dv) \\ &= \dot{\mathbf{e}}^T(t) \tau_1^2 R_1 \dot{\mathbf{e}}(t) - \textstyle\int_{t-\tau_1}^{t} \dot{\mathbf{e}}^T(v) \tau_1 R_1 \dot{\mathbf{e}}(v) dv + \dot{\mathbf{e}}^T(t) \tau_2^2 R_2 \dot{\mathbf{e}}(t) - \textstyle\int_{t-\tau_2}^{t} \dot{\mathbf{e}}^T(v) \tau_2 R_2 \dot{\mathbf{e}}(v) dv \\ \dot{V_4} &= \dot{\mathbf{e}}^T(t)(\tau_2 - \tau_1)^2 S \dot{\mathbf{e}}(t) - \int_{t-\tau_2}^{t-\tau_1} \dot{\mathbf{e}}^T(v)(\tau_2 - \tau_1) S \dot{\mathbf{e}}(v) dv. \end{aligned}$$

Similar to Case I, apply Jensen's Inequality to reduce the following terms $-\int_{t-\tau_2}^{t-\tau_1} \dot{\mathbf{e}}^T(v)(\tau_2 - \tau_1) S \dot{\mathbf{e}}(v) dv$, $-\int_{t-\tau_1}^{t} \dot{\mathbf{e}}^T(v) \tau_1 R_1 \dot{\mathbf{e}}(v) dv$, $-\int_{t-\tau_2}^{t} \dot{\mathbf{e}}^T(v) \tau_2 R_2 \dot{\mathbf{e}}(v) dv$.

$$-\textstyle\int_{t-\tau_2}^{t-\tau_1} \dot{\mathbf{e}}^T(v)(\tau_2 - \tau_1) S \dot{\mathbf{e}}(v) dv \leq \begin{bmatrix} \mathbf{e}(t-\tau_1) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix}^T \begin{bmatrix} -S & S & 0 \\ * & -2S & S \\ * & * & -S \end{bmatrix} \begin{bmatrix} \mathbf{e}(t-\tau_1) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix} \quad (3.19)$$

$$-\int_{t-\tau_1}^{t} \dot{\mathbf{e}}^T(v) \tau_1 R_1 \dot{\mathbf{e}}(v) dv \leq \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau_1) \end{bmatrix}^T \begin{bmatrix} -R_1 & R_1 \\ * & -R_1 \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau_1) \end{bmatrix} \quad (3.20)$$

$$-\textstyle\int_{t-\tau_2}^{t} \dot{\mathbf{e}}^T(v) \tau_2 R_2 \dot{\mathbf{e}}(v) dv \leq \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix}^T \begin{bmatrix} -R_2 & R_2 & 0 \\ * & -2R_2 & R_2 \\ * & * & -R_2 \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t-\tau(t)) \\ \mathbf{e}(t-\tau_2) \end{bmatrix} \quad (3.21)$$

The remaining portions of $\dot{V}$ are then

$$\begin{aligned}\dot{V} &= \mathbf{e}(t)^T P\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}(t)^T P\mathbf{e}(t) + \mathbf{e}^T(t)Q_1\mathbf{e}(t) - \mathbf{e}^T(t-\tau_1)Q_1\mathbf{e}(t-\tau_1) + \mathbf{e}^T(t)Q_2\mathbf{e}(t)\\ &- \mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)\tau_1^2 R_1\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}^T(t)\tau_2^2 R_2\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}^T(t)(\tau_2-\tau_1)^2 S\dot{\mathbf{e}}(t).\end{aligned}$$

Then combine terms to arrive at

$$\begin{aligned}\dot{V} &= \mathbf{e}(t)^T P\dot{\mathbf{e}}(t) + \dot{\mathbf{e}}(t)^T P\mathbf{e}(t) + \mathbf{e}^T(t)(Q_1+Q_2)\mathbf{e}(t) - \mathbf{e}^T(t-\tau_1)Q_1\mathbf{e}(t-\tau_1)\\ &- \mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)(\tau_1^2 R_1 + (\tau_2-\tau_1)^2 S + \tau_2^2 R_2)\dot{\mathbf{e}}(t).\end{aligned}$$

Next substitute $\dot{\mathbf{e}}$ into the V_1 portion of the Lyapunov Candidate

$$\begin{aligned}\dot{V} &= \mathbf{e}(t)^T P(A_M\mathbf{e}(t) + B_sK_c\mathbf{e}(t-\tau(t)) + D) + (A_M\mathbf{e}(t) + B_sK_c\mathbf{e}(t-\tau(t))\\ &+ D)^T P(\mathbf{e}(t)) + \mathbf{e}^T(t)(Q_1+Q_2)\mathbf{e}(t) - \mathbf{e}^T(t-\tau_1)Q_1\mathbf{e}(t-\tau_1)\\ &- \mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)(\tau_1^2 R_1 + (\tau_2-\tau_1)^2 S + \tau_2^2 R_2)\dot{\mathbf{e}}(t)\\ \dot{V} &= \mathbf{e}^T(t)(Q_1+Q_2+PA_M+A_M^TP+\gamma_1^{-1}PP)\mathbf{e}(t) - \mathbf{e}^T(t-\tau_1)Q_1\mathbf{e}(t-\tau_1)\\ &- \mathbf{e}^T(t-\tau_2)Q_2\mathbf{e}(t-\tau_2) + \dot{\mathbf{e}}^T(t)(\tau_1^2 R_1 + (\tau_2-\tau_1)^2 S + \tau_2^2 R_2)\dot{\mathbf{e}}(t) + \gamma_1 D^T D.\end{aligned}$$

Converting into an LMI form with (3.19)-(3.21)

$$\begin{bmatrix}\mathbf{e}(t)\\ \mathbf{e}(t-\tau_1)\\ \mathbf{e}(t-\tau(t))\\ \mathbf{e}(t-\tau_2)\end{bmatrix}^T \begin{bmatrix}\Pi_1 & R_1 & \Pi_2 & 0\\ R_1 & -S-R_1-Q_1 & S & 0\\ \Pi_2^T & S & -2S-2R_2 & S+R_2\\ 0 & 0 & S+R_2 & -S-R_2-Q_2\end{bmatrix}\begin{bmatrix}\mathbf{e}(t)\\ \mathbf{e}(t-\tau_1)\\ \mathbf{e}(t-\tau(t))\\ \mathbf{e}(t-\tau_2)\end{bmatrix}$$

where

$$\begin{aligned}\Pi_1 &= Q_1+Q_2+PA_M+A_M^TP+\gamma_1^{-1}PP-R_2-R_1\\ \Pi_2 &= R_2+PB_sK_c.\end{aligned}$$

Let $\zeta = \begin{bmatrix}\mathbf{e}(t)\\ \mathbf{e}(t-\tau(t))\end{bmatrix}$ and rearrange the LMI resulting in

$$\begin{bmatrix}\mathbf{e}(t-\tau_1)\\ \mathbf{e}(t-\tau_2)\\ \zeta\end{bmatrix}^T \begin{bmatrix}-S-R_1-Q_1 & 0 & \Pi_4\\ * & -S-R_2-Q_2 & \Pi_5\\ * & * & \Pi_3\end{bmatrix}\begin{bmatrix}\mathbf{e}(t-\tau_1)\\ \mathbf{e}(t-\tau_2)\\ \zeta\end{bmatrix} \tag{3.22}$$

where

$$\begin{aligned}\Pi_3 &= \begin{bmatrix}\Pi_1 & \Pi_2\\ * & -2S-2R_2\end{bmatrix}\\ \Pi_4 &= \begin{bmatrix}R_1 & S\end{bmatrix}\\ \Pi_5 &= \begin{bmatrix}0 & S+R_2\end{bmatrix}.\end{aligned}$$

The next term that needs to be examined is $\dot{\mathbf{e}}^T(t)(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)\dot{\mathbf{e}}(t)$. By substituting $\dot{\mathbf{e}} = E(t) + D$ into the term it becomes

$$\begin{aligned}
& [E + D]^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)[E + D] \\
= \; & E^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)E + E^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)D \\
& + D^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)D + D^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)E \\
= \; & E^T(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2 + \gamma_2^{-1} I)E \\
& + D^T(\gamma_2(\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2)^2 + (\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2))D. \qquad (3.23)
\end{aligned}$$

By defining the following matrix relationships

$$\begin{aligned}
\tau_1^2 R_1 + (\tau_2 - \tau_1)^2 S + \tau_2^2 R_2 &= \begin{bmatrix} \tau_1 & \tau_2 & \tau_2 - \tau_1 \end{bmatrix} \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & S \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_2 - \tau_1 \end{bmatrix} \\
E &= \begin{bmatrix} A_M & B_s K_c \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}(t - \tau) \end{bmatrix}
\end{aligned}$$

And substituting into (3.23) resulting in

$$\begin{bmatrix} \zeta \end{bmatrix}^T \begin{bmatrix} A_M & B_s K_c \end{bmatrix}^T \left(\begin{bmatrix} \tau_1 & \tau_2 & \tau_2 - \tau_1 \end{bmatrix} \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & S \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_2 - \tau_1 \end{bmatrix} + \gamma_2^{-1} I \right) \begin{bmatrix} A_M & B_s K_c \end{bmatrix} \begin{bmatrix} \zeta \end{bmatrix}.$$

Then combining the ζ term to the previous results of the LMI in (3.22)

$$\begin{bmatrix} \zeta \end{bmatrix}^T \Pi_3 + \begin{bmatrix} A_M & B_s K_c \end{bmatrix}^T \left(\begin{bmatrix} \tau_1 & \tau_2 & \tau_2 - \tau_1 \end{bmatrix} \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & S \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_2 - \tau_1 \end{bmatrix} + \gamma_2^{-1} I \right) \begin{bmatrix} A_M & B_s K_c \end{bmatrix} \begin{bmatrix} \zeta \end{bmatrix}. \qquad (3.24)$$

Using the Schur Complement on (3.24) to arrive at

$$\begin{bmatrix} A & B \\ B^T & F \end{bmatrix}$$

where

$$\begin{aligned}
A &= \Pi_3 + \begin{bmatrix} A_M^T \\ (B_s K_c)^T \end{bmatrix} \gamma_2^{-1} I \begin{bmatrix} A_M & B_s K_c \end{bmatrix} \\
B &= \begin{bmatrix} A_M^T \\ (B_s K_c)^T \end{bmatrix} \begin{bmatrix} \tau_1 & \tau_2 & \tau_2 - \tau_1 \end{bmatrix} \\
F &= \begin{bmatrix} -R_1^{-1} & 0 & 0 \\ 0 & -R_2^{-1} & 0 \\ 0 & 0 & -S^{-1} \end{bmatrix}.
\end{aligned}$$

Substituting back into (3.22) yields

$$\begin{bmatrix} -S-R_1-Q_1 & 0 & \Pi_4 & 0 \\ * & -S-R_2-Q_2 & \Pi_5 & 0 \\ * & * & \Pi_3+\begin{bmatrix} A_M^T \\ (B_sK_c)^T \end{bmatrix}\gamma_2^{-1}I\begin{bmatrix} A_M & B_sK_c \end{bmatrix} & \begin{bmatrix} A_M^T \\ (B_sK_c)^T \end{bmatrix}\begin{bmatrix} \tau_1 & \tau_2 & \tau_2-\tau_1 \end{bmatrix} \\ * & * & * & \begin{bmatrix} -R_1^{-1} & 0 & 0 \\ 0 & -R_2^{-1} & 0 \\ 0 & 0 & -S^{-1} \end{bmatrix} \end{bmatrix} < 0. \quad (3.25)$$

Applying the Schur Complement again on the term $\Pi_3 + \begin{bmatrix} A_M^T \\ (B_sK_c)^T \end{bmatrix}\gamma_2^{-1}I\begin{bmatrix} A_M & B_sK_c \end{bmatrix}$ and expanding the rows and columns to acquire

$$\begin{bmatrix} -S-R_1-Q_1 & 0 & R_1 & S & 0 & 0 & 0 & 0 \\ * & -S-R_2-Q_2 & 0 & S+R_2 & 0 & 0 & 0 & 0 \\ * & * & Q_1+Q_2+PA_M+A_M^TP+\gamma_1 - 1PP-R_2-R_1 & R_2+PB_sK_c & A_M^T & \tau_1A_M^T & \tau_2A_M^T & (\tau_2-\tau_1)A_M^T \\ & * & * & -2S-2R_2 & (B_sK_c)^T & \tau_1(B_sK_c)^T & \tau_2(B_sK_c)^T & (\tau_2-\tau_1)A_M^T \\ * & * & * & * & \gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & -R_1^{-1} & 0 & 0 \\ * & * & * & * & * & * & -R_2^{-1} & 0 \\ * & * & * & * & * & * & * & -S^{-1} \end{bmatrix} < 0 \quad (3.26)$$

Then

$$\begin{aligned} \dot{V} &\leq [\zeta]^T \begin{bmatrix} -S-R_1-Q_1 & 0 & R_1 & S & 0 & 0 & 0 & 0 \\ * & -S-R_2-Q_2 & 0 & S+R_2 & 0 & 0 & 0 & 0 \\ * & * & Q_1+Q_2+PA_M+A_M^TP+\gamma_1^{-1}PP-R_2-R_1 & R_2+PB_sK_c & A_M^T & \tau_1A_M^T & \tau_2A_M^T & (\tau_2-\tau_1)A_M^T \\ & * & * & -2S-2R_2 & K_c^TB_s^T & \tau_1K_c^TB_s^T & \tau_2K_c^TB_s^T & (\tau_2-\tau_1)K_c^TB_s^T \\ * & * & * & * & -\gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & -R_1^{-1} & 0 & 0 \\ * & * & * & * & * & * & -R_2^{-1} & 0 \\ * & * & * & * & * & * & * & -S^{-1} \end{bmatrix} [\zeta] \\ &\quad +\gamma_1D^TD + D^T(\gamma_2(\tau_1^2R_1+(\tau_2-\tau_1)^2S+\tau_2^2R_2)^2+\tau_1^2R_1+(\tau_2-\tau_1)^2S+\tau_2^2R_2)D \\ &\leq \zeta^T\Theta\zeta - \mathbf{e}^T\mathbf{e} + D^T(\gamma_1I+\gamma_2(\tau_1^2R_1+(\tau_2-\tau_1)^2S+\tau_2^2R_2)^2+\tau_1^2R_1+(\tau_2-\tau_1)^2S+\tau_2^2R_2)D \end{aligned}$$

where

$$\Theta = \begin{bmatrix} -S-R_1-Q_1 & 0 & R_1 & S & 0 & 0 & 0 & 0 \\ * & -S-R_2-Q_2 & 0 & S+R_2 & 0 & 0 & 0 & 0 \\ * & * & Q_1+Q_2+PA_M+A_M^TP+\gamma_1^{-1}PP-R_2-R_1+I & R_2+PB_sK_c & A_M^T & \tau_1A_M^T & \tau_2A_M^T & (\tau_2-\tau_1)A_M^T \\ & * & * & -2S-2R_2 & K_c^TB_s^T & \tau_1K_c^TB_s^T & \tau_2K_c^TB_s^T & (\tau_2-\tau_1)K_c^TB_s^T \\ * & * & * & * & -\gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & -R_1^{-1} & 0 & 0 \\ * & * & * & * & * & * & -R_2^{-1} & 0 \\ * & * & * & * & * & * & * & -S^{-1} \end{bmatrix}.$$

Due to some of the non-convex terms, such as PB_sK_c, it is necessary to linearize by pre- and post- multiplying by $diag(P^{-1}, P^{-1}, P^{-1}, P^{-1}, I, I, I, I)$

$$\begin{bmatrix} -\hat{S}-\hat{R}_1-\hat{Q}_1 & 0 & \hat{R}_1 & \hat{S} & 0 & 0 & 0 & 0 \\ * & -\hat{S}-\hat{R}_2-\hat{Q}_2 & 0 & \hat{S}+\hat{R}_2 & 0 & 0 & 0 & 0 \\ * & * & \hat{Q}_1+\hat{Q}_2+A_MX+XA_M^T+\gamma_1^{-1}I-\hat{R}_2-\hat{R}_1+XIX & \hat{R}_2+B_sY & XA_M^T & \tau_1XA_M^T & \tau_2XA_M^T & (\tau_2-\tau_1)XA_M^T \\ & * & * & -2\hat{S}-2\hat{R}_2 & Y^TB_s^T & \tau_1Y^TB_s^T & \tau_2Y^TB_s^T & (\tau_2-\tau_1)Y^TB_s^T \\ * & * & * & * & -\gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & -R_1^{-1} & 0 & 0 \\ * & * & * & * & * & * & -R_2^{-1} & 0 \\ * & * & * & * & * & * & * & -S^{-1} \end{bmatrix} \quad (3.27)$$

where,

$$\begin{aligned}
X &= P^{-1}\\
Y &= KX\\
\hat{Q}_1 &= XQ_1X\\
\hat{Q}_2 &= XQ_2X\\
\hat{R}_1 &= XR_1X\\
\hat{R}_2 &= XR_2X\\
\hat{S} &= XSX.
\end{aligned}$$

If $\Theta < 0$ holds, then

$$\dot{V} \leq \; -\|\mathbf{e}(t)\|^2 + \|D^T(\gamma_1 I + \gamma_2(\tau_1^2 R_1 + (\tau_2-\tau_1)^2 S + \tau_2^2 R_2)^2 + \tau_1^2 R_1 + (\tau_2-\tau_1)^2 S + \tau_2^2 R_2)D\| \tag{3.28}$$

which shows that the error is bounded and that the bound is determined by the magnitude of the terms γ_1 and γ_2, $\|D\|$, the eigenvalues of R_1,R_2, and S, the upper bound of the delay τ_2, the lower bound of the delay τ_1, and the range of the time delay $\tau_2 - \tau_1$.

Applying the Schur Complement again to Θ removes the XIX and $\gamma_1^{-1}I$ terms

$$\begin{bmatrix}
-\hat{S}-\hat{R}_1-\hat{Q}_1 & 0 & \hat{R}_1 & 0 & 0 & \hat{S} & 0 & 0 & 0 & 0\\
* & -\hat{S}-\hat{R}_2-\hat{Q}_2 & 0 & 0 & 0 & \hat{S}+\hat{R}_2 & 0 & 0 & 0 & 0\\
* & * & \hat{Q}_1+\hat{Q}_2+A_M X+XA_M^T-\hat{R}_2-\hat{R}_1 & X & 0 & \hat{R}_2+B_s Y & XA_M^T & \tau_1 XA_M^T & \tau_2 XA_M^T & (\tau_2-\tau_1)XA_M^T\\
* & * & * & -I & I & 0 & 0 & 0 & 0 & 0\\
* & * & * & * & -\gamma_1 I & 0 & 0 & 0 & 0 & 0\\
* & * & * & * & * & -2\hat{S}-2\hat{R}_2 & Y^T B_s^T & \tau_1 Y^T B_s^T & \tau_2 Y^T B_s^T & (\tau_2-\tau_1)Y^T B_s^T\\
* & * & * & * & * & * & -\gamma_2 I & 0 & 0 & 0\\
* & * & * & * & * & * & * & -R_1^{-1} & 0 & 0\\
* & * & * & * & * & * & * & * & -R_2^{-1} & 0\\
* & * & * & * & * & * & * & * & * & -S^{-1}
\end{bmatrix}. \tag{3.29}$$

Unfortunately, Θ has 3 non-convex terms ($-R_1^{-1}$,$-R_2^{-1}$,$-S^{-1}$) that need to be examined further. The first step is to convert the variables to their linearized equivalents (i.e. $\hat{R}_1$ as opposed to R_1) through the following relationships

$$\hat{R}_1 = XR_1X, \quad R_1^{-1} = X\hat{R_1}^{-1}X$$

$$\hat{R}_2 = XR_2X, \quad R_2^{-1} = X\hat{R_2}^{-1}X$$

$$\hat{S} = XSX, \quad S^{-1} = X\hat{S}^{-1}X.$$

Substituting the above transformations we arrive at

$$\begin{bmatrix} -\hat{S}-\hat{R}_1-\hat{Q}_1 & 0 & \hat{R}_1 & 0 & 0 & \hat{S} & 0 & 0 & 0 & 0 \\ * & -\hat{S}-\hat{R}_2-\hat{Q}_2 & 0 & 0 & 0 & \hat{S}+\hat{R}_2 & 0 & 0 & 0 & 0 \\ * & * & \hat{Q}_1+\hat{Q}_2+A_MX+XA_M^T-\hat{R}_2-\hat{R}_1 & X & 0 & \hat{R}_2+B_sY & XA_M^T & \tau_1XA_M^T & \tau_2XA_M^T & (\tau_2-\tau_1)XA_M^T \\ * & * & * & -I & I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & -\gamma_1I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & -2\hat{S}-2\hat{R}_2 & Y^TB_s^T & \tau_1Y^TB_s^T & \tau_2Y^TB_s^T & (\tau_2-\tau_1)Y^TB_s^T \\ * & * & * & * & * & * & -\gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & * & * & -X\hat{R}_1^{-1}X & 0 & 0 \\ * & * & * & * & * & * & * & * & -X\hat{R}_2^{-1}X & 0 \\ * & * & * & * & * & * & * & * & * & -X\hat{S}^{-1}X \end{bmatrix}. \tag{3.30}$$

This completes the proof. ■

It is necessary to find a method to deal with the non-linear terms $X\hat{R}_1{}^{-1}X$,$X\hat{R}_2{}^{-1}X$, and $X\hat{S}^{-1}X$ in (3.18). To do so, replace $X\hat{R}_1{}^{-1}X$,$X\hat{R}_2{}^{-1}X$, and $X\hat{S}^{-1}X$ with a new variables G_1,G_2, and G_3 while apply additional constraints to the LMI

$$\begin{bmatrix} -\hat{S}-\hat{R}_1-\hat{Q}_1 & 0 & \hat{R}_1 & 0 & 0 & \hat{S} & 0 & 0 & 0 & 0 \\ * & -\hat{S}-\hat{R}_2-\hat{Q}_2 & 0 & 0 & 0 & \hat{S}+\hat{R}_2 & 0 & 0 & 0 & 0 \\ * & * & \hat{Q}_1+\hat{Q}_2+A_MX+XA_M^T-\hat{R}_2-\hat{R}_1 & X & 0 & \hat{R}_2+B_sY & XA_M^T & \tau_1XA_M^T & \tau_2XA_M^T & (\tau_2-\tau_1)XA_M^T \\ * & * & * & -I & I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & -\gamma_1I & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * & -2\hat{S}-2\hat{R}_2 & Y^TB_s^T & \tau_1Y^TB_s^T & \tau_2Y^TB_s^T & (\tau_2-\tau_1)Y^TB_s^T \\ * & * & * & * & * & * & -\gamma_2I & 0 & 0 & 0 \\ * & * & * & * & * & * & * & -G1 & 0 & 0 \\ * & * & * & * & * & * & * & * & -G2 & 0 \\ * & * & * & * & * & * & * & * & * & -G3 \end{bmatrix} \tag{3.31}$$

Define $J_1 = G_1^{-1}$, $P = X^{-1}$ and $L_1 = R_1^{-1}$, therefore G_1 can be translated into:

$$\begin{bmatrix} J_1 & P \\ * & L_1 \end{bmatrix} \geq 0. \tag{3.32}$$

Define $J_2 = G_2^{-1}$, $P = X^{-1}$ and $L_2 = R_2^{-1}$, therefore G_2 can be translated into:

$$\begin{bmatrix} J_2 & P \\ * & L_2 \end{bmatrix} \geq 0. \tag{3.33}$$

Define $J_3 = G_3^{-1}$, $P = X^{-1}$ and $L_3 = S^{-1}$, therefore G_3 can be translated into:

$$\begin{bmatrix} J_3 & P \\ * & L_3 \end{bmatrix} \geq 0. \tag{3.34}$$

To help enforce the inverse relationships between variables define the following:

$$\begin{aligned} &\text{Minimize Trace}(XP + J_1G_1 + R_1L_1 + J_2G_2 + R_2L_2 + J_3G_3 + SL_3) \\ &\text{subject to } X > 0, P > 0, G_1 > 0, J_1 > 0, L_1 > 0, R_1 > 0, G_2 > 0, J_2 > 0, \\ &L_2 > 0, R_2 > 0, , G_3 > 0, J_3 > 0, L_3 > 0, S > 0 \\ &\begin{bmatrix} X & I \\ * & P \end{bmatrix} > 0; \begin{bmatrix} G_1 & I \\ * & J_1 \end{bmatrix} > 0; \begin{bmatrix} G_2 & I \\ * & J_2 \end{bmatrix} > 0; \begin{bmatrix} G_3 & I \\ * & J_3 \end{bmatrix} > 0; \\ &\begin{bmatrix} R_1 & I \\ * & L_1 \end{bmatrix} > 0; \begin{bmatrix} R_2 & I \\ * & L_2 \end{bmatrix} > 0; \begin{bmatrix} S & I \\ * & L_3 \end{bmatrix} > 0 \end{aligned} \tag{3.35}$$

The new LMI problem can be solved by applying the cone complementarity algorithm [41] in the following manner:
Step 1: Find a set of feasible matrices
$(X, Y, Q_1, Q_2, P, G_1, R_1, L_1, J_1, G_2, R_2, L_2, J_2, G_3, S, L_3, J_3)^0$
that satisfies (3.31),(3.32),(3.33),(3.34) and (3.35).
Step 2: Solve the following LMI minimization problem

$$\begin{aligned}&\text{Minimize Trace}(XP^0 + PX^0 + G_1J_1^0 + J_1G_1^0 + R_1L_1^0 + L_1R_1^0 + G_2J_2^0 + J_2G_2^0\\&+R_2L_2^0 + L_2R_2^0 + G_3J_3^0 + J_3G_3^0 + SL_3^0 + L_3S^0)\\&\text{subject to (3.31)(3.32)(3.33)(3.34) and (3.35)}\end{aligned}$$

Step 3: Substitute the new matrix variables from the previous step into (3.18). If the result is feasible set $K_c = YX^{-1}$. If not feasible, set the newly acquired matrices to $(X, Y, G, R, L, Q, P, J)^0$ and go to step 2.
While the above theorem and algorithm are guaranteed to provide a stable result, they are not guaranteed to find a result. Due to the fact that no solution may exist, a loop counter and break command should be used to prevent an infinite loop.

3.4 Summary

The section outlines the designed control algorithms for the two cases considered: only an upper bound on time delay (Case I), and an upper and lower bound on the delay (Case II). The same master impedance controller is used for both cases, and the error dynamics remains unchanged for each case. Both cases are LMI based solutions from similar Lyapunov functions, with extra terms in Case II to account for the added lower level of delay. The LMIs are formulated using techniques such as Jensen's Inequality, Schur Complement, and the cone complementarity algorithm to eliminate inherent non-linearities in the LMIs in Theorems 1 and 2.

Chapter 4

Simulation Results

The following sections deal with the simulation work done using Matlab and Simulink. Matlab was used to solve the LMI problems as described in Section 3.2 to obtain the control gain K_c, while Simulink was used to perform the system simulation and achieve results.

4.1 System Parameters

The following are a list of base system parameters used for computing the LMI and running the simulations:

$$\begin{aligned}
m_m &= 3\text{kg} \\
b_m &= 3\text{Ns/m} \\
k_m &= 0\text{N/m} \\
m_s &= 2\text{kg} \\
b_s &= 2\text{Ns/m} \\
k_s &= 0\text{N/m} \\
M &= 1\text{kg} \\
B &= 4\text{Ns/m} \\
K &= 4\text{N/m} \\
B_e &= 0\text{Ns/m} \\
K_e &= 4\text{N} \\
\tau_2 &= 0.08\text{s} \\
\tau_1 &= 0.02\text{s.}
\end{aligned}$$

It should be noted that for Case I τ_1 is not used, as no lower bound on the time delay was considered.

4.2 Case I: Only Upper Delay Bound

The first case considered was with only an upper bound on the delay of τ_2, with the delay variable between 0 and τ_2.

4.2.1 Base LMI Results

Using the system parameters from Section 4.1, the LMI code ((3.15)(3.16) and (3.17)) was run to generate the following LMI variables

$$\begin{aligned}
X &= \begin{bmatrix} 0.1755 & -0.1352 \\ -0.1352 & 0.1725 \end{bmatrix} \\
Y &= 1 * 10^{-3} \begin{bmatrix} -0.4836 & -0.8777 \end{bmatrix} \\
\hat{R} &= 1 * 10^{-3} \begin{bmatrix} 0.384 & 0.0116 \\ 0.0116 & 0.2907 \end{bmatrix} \\
\hat{Q} &= 1 * 10^{-5} \begin{bmatrix} 0.2991 & -0.1156 \\ -0.1156 & 0.0909 \end{bmatrix} \\
\gamma_1 &= 5 \\
\gamma_2 &= 0.9
\end{aligned}$$

which led to the static feedback control gain of $K_c = \begin{bmatrix} -0.0168 & -0.0183 \end{bmatrix}$ from the LMI stabilization algorithm. It should be noted that the values of γ_1 and γ_2 were pre selected to the approximate minimum values that would allow the algorithm to find a solution. This was done to help reduce computational load as minimum values for γ_1 and γ_2 are desired based on the derived error bound expression (3.14).

4.2.2 Modified Step Response

The first input type tested on the system was that of a modified step response. The input forcing function F_h was modeled as a ramp function up to a value of 1 N with a slope of 0.2 N/s. After reaching a value of 1 N the force was held constant as in Fig. 4.1 (a). The corresponding control signals are displayed in Fig. 4.1 (b), and show relatively smooth behavior with no extreme control inputs or discontinuities present, which is a desired trait for smooth teleoperation. The net force seen by the manipulators as in Fig. 4.1 (c), shows slightly noisier signals being sent to the manipulator, but with low net values of force (all less than 0.2 N). The net force is calculated as

$$\begin{aligned}
F_{net,m} &= U_m + F_h \\
F_{net,s} &= U_s - F_e
\end{aligned}$$

Also shown is the fact that equilibrium with the environment is achieved after approximately 7 seconds, when the net manipulator forces going to zero. It should be noted that F_h and F_e need not be equal for equilibrium to be reached due to the asymmetry in the system and the scaling of F_h for the impedance matching controller.

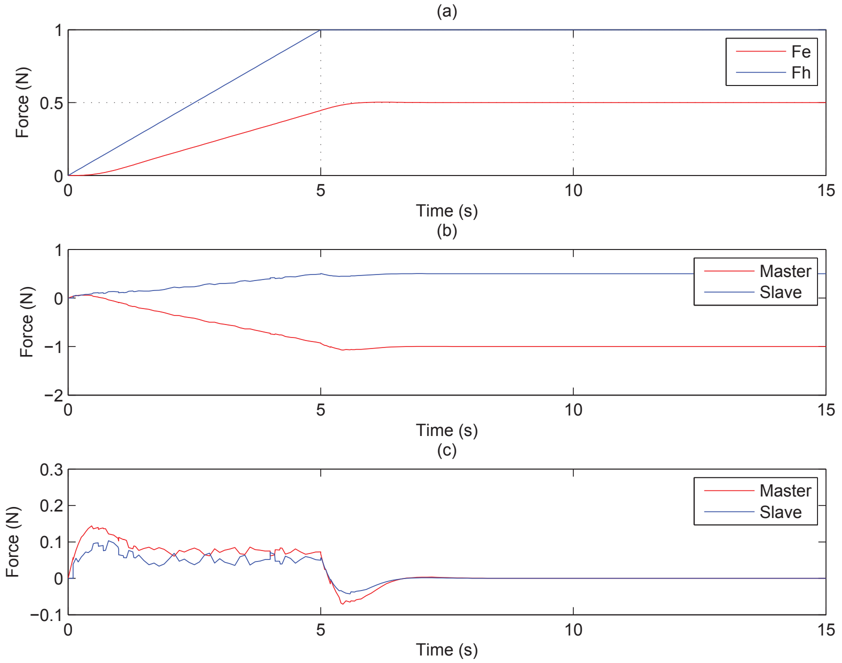


Figure 4.1: (a) External forces, (b) Control signals and (c) Net manipulator forces under a modified step input with only upper delay bound τ_2

Similar to the forces plotted in Fig. 4.1, the state information is displayed in Fig. 4.2. The positions of the manipulators are shown in Fig. 4.2 (a) which displays the desired tracking between master and slave with a final position of 0.125 m. Fig. 4.2 (b) displays the velocities of both manipulators with the expected ramping to an approximately constant velocity to match the input, and then deceleration back to no movement when force equilibrium is occurring on the manipulators. Fig. 4.2 (c) displays the positional error between the master and slave manipulators which peaks at 3.4 mm, giving a maximum error relative to final position of 2.72 %.

Lastly, Fig. 4.3 plots the positional error with reference to maximum error bound as derived in (3.14). As shown, the error clearly lies within the bounding limits of the algorithm, further verifying its ability to provide a stable bilateral teleoperation setup. Only the maximum error bound is displayed due to the extremely noisy behavior of the error bound. The error bound varies with the external forces F_h and F_e and the true level of time delay τ in the system, causing the noisy behavior. If τ were constant, an accurate plot of the error bound as a function of time could be obtained.

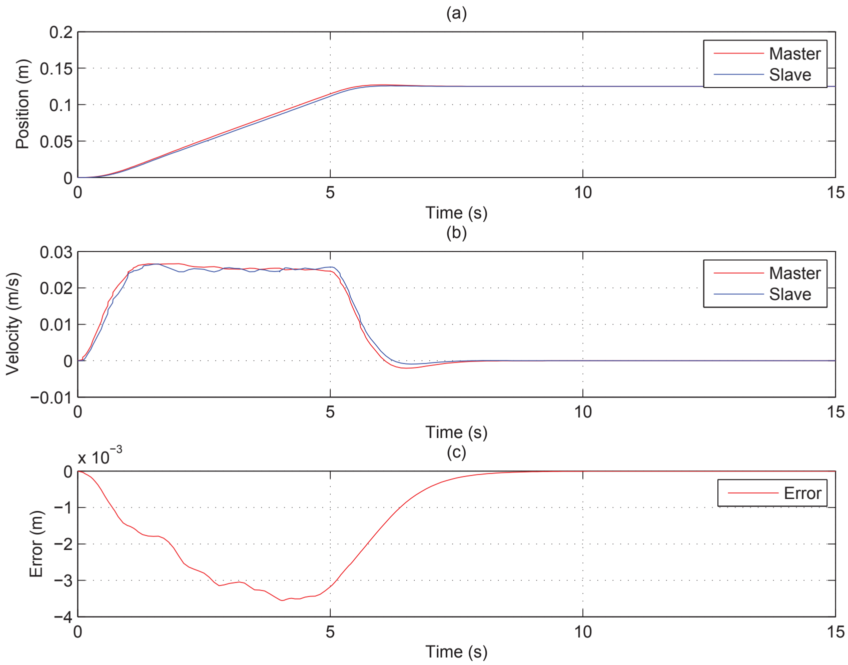


Figure 4.2: (a) Position, (b) Velocity and (c) Positional error under a modified step input with only upper delay bound τ_2

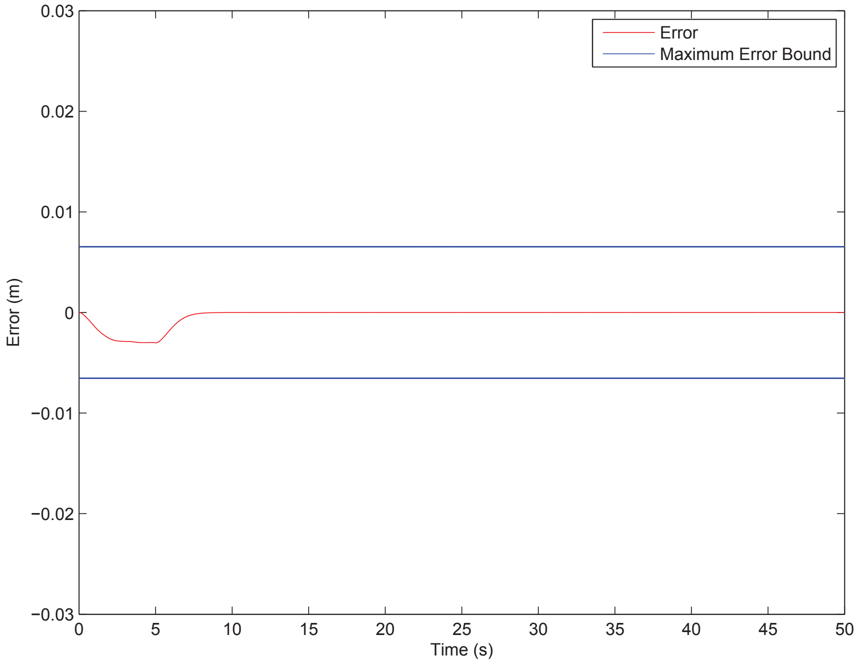


Figure 4.3: Positional error and its maximum bound under a modified step input with only upper delay bound τ_2

4.2.3 Sinusoid Response

A sinusoid was considered as a second human input type. For simulation testing, a sinusoid with a magnitude of 5 N and a frequency of 0.5 Hz was used, which falls within the 6 Hz limit of voluntary human motion [55]. The input is shown in Fig. 4.4 (a), along with the wall contact forces that occur only when the position of the system is positive. This is due to the assumption that the slave manipulator is resting against the object of interest in the environment. Displayed in Fig. 4.4 (b) are the control signals from the master side impedance matching controller and slave side stability controller respectively. Similar to the case with the modified step input, the signals are relatively smooth with no discontinuities or extreme force values as all values fall within ±7 N. Lastly, Fig. 4.4 (c) displays the net force as seen by the manipulators, and once again the signal is noisier than the control signal but again is bounded, this time by a magnitude of 2 N.

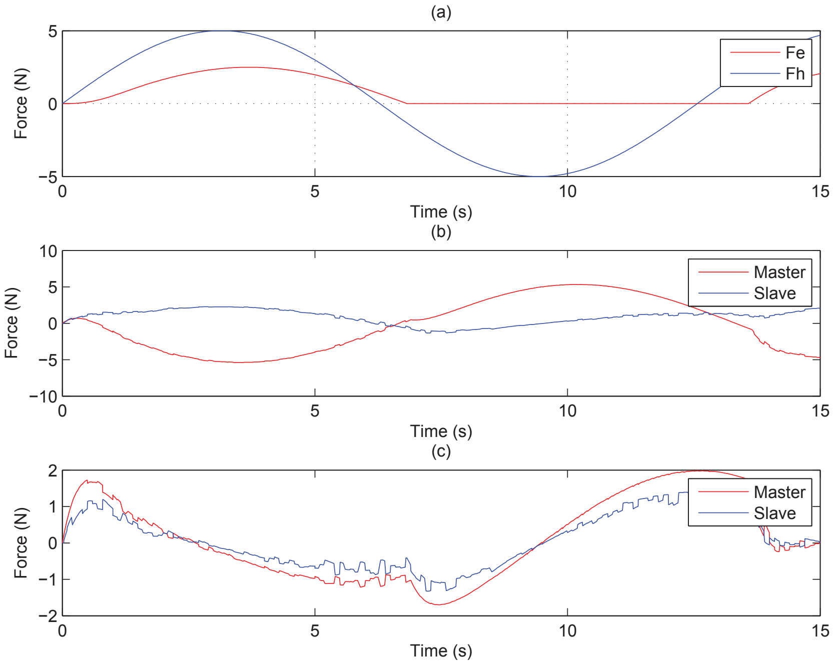


Figure 4.4: (a) External forces, (b) Control signals and (c) Net manipulator forces under a sinusoidal input with only upper delay bound τ_2

Fig. 4.5 (a) plots the positions of both the master and slave manipulator and displays the high level of tracking available in the system as it is difficult to discern the level of error from Fig. 4.5 (a) alone. Displayed in Fig. 4.5 (b) are the respective velocities that show relatively smooth motion. Smooth motion is preferable as it would be undesirable for high levels of overshoot in the slave tracking performance leading to unnecessary fluctuations in the contact force F_e. Fig. 4.5 (c) plots the error in position of the slave from the master. As shown, a peak error of 0.03 m is realized. When compared to a maximum distance of travel from equilibrium of 1.2 m,a percent error relative to the maximum travel of 2.5 % is realized.

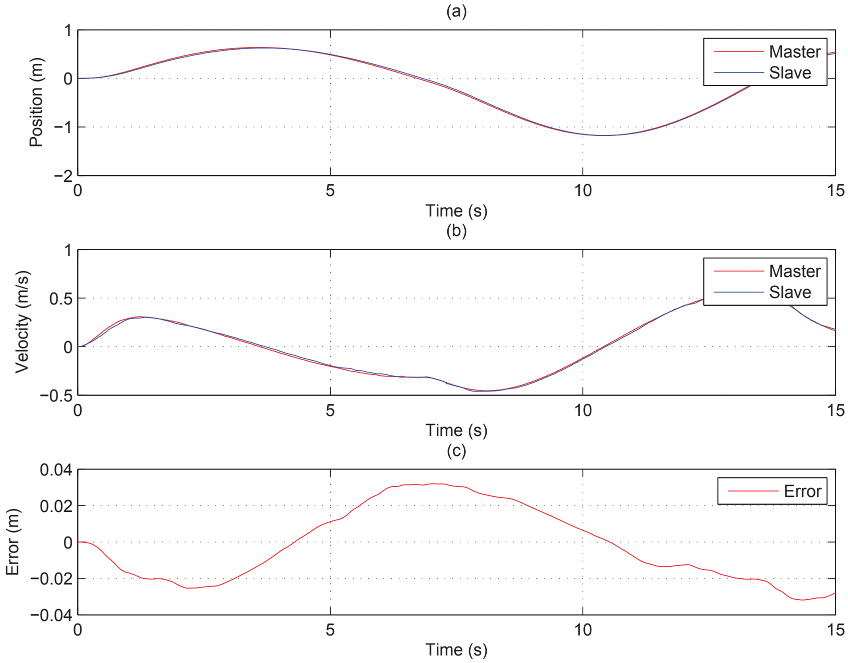


Figure 4.5: (a) Position, (b) Velocity and (c) Positional error under a sinusoidal input with only upper delay bound τ_2

Lastly, Fig. 4.6 plots the positional error with reference to maximum error bound as derived in (3.14). The plot shows that the error clearly lies within the bounding limits of the algorithm.

4.2.4 Parametric Study

It was determined that a parametric study on the fixed variables of τ_2, γ_1, and γ_2 be conducted as to explore their effects on the error. It was expected that increasing these values would increase the overall system error based on (3.14), although variations in the LMI solutions may cause other behavior. All of the system parameters are the same as in Section 4.1, with the input being a similar sinusoidal function as in Section 4.2.3, except the magnitude was reduced to 3 N.

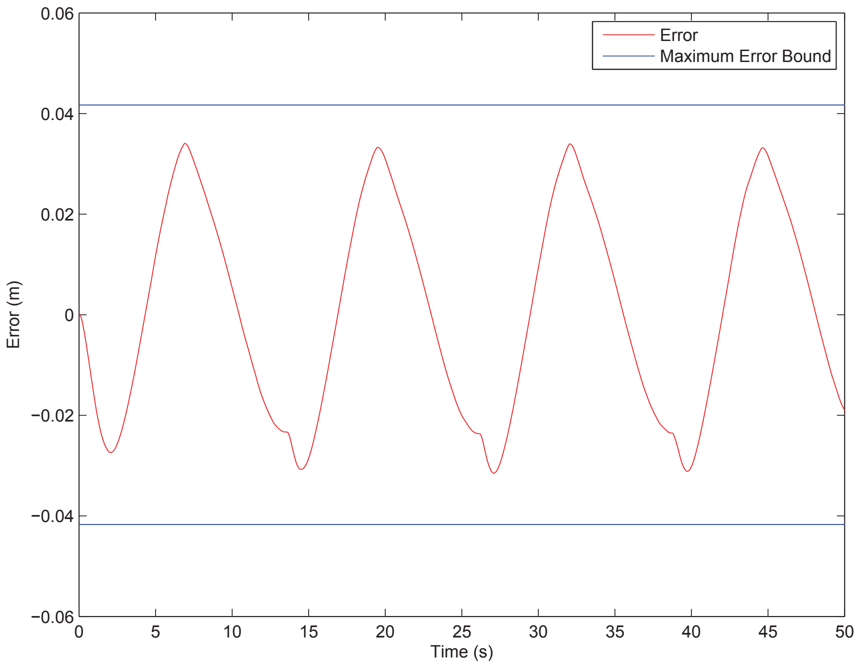


Figure 4.6: Positional error and its maximum bound under a sinusoidal input with only upper delay bound τ_2

τ_2 variations

The first variable considered was τ_2, as knowing the maximum allowable time delay that the algorithm can handle, along with the effects of the delay on the system error are desired. As shown in Fig. 4.7 (a), the maximum level of delay that the LMI can handle is 0.46 seconds, which provides a maximum error of 0.118 m, a considerable increase from the 0.02 m maximum error when τ_2 is set to 0.08 seconds. Fig. 4.7 (b) shows the maximum position of the master and that it remains relatively unchanged in the face of the delay. This results in the system percent error increasing from 2.8 % up to 17 % as τ_2 grows from 0.08 seconds to 0.46 seconds. It is important to monitor the control signals for undesirable maximum values so as to ensure safe operation of the hardware. As shown in Fig. 4.8 (a), the master's control signal maximum value stays within the bound of 3.213 N, a reasonable level considering the 3 N peak input. Similarly, the slave peak signal reaches limits of 1.53 N, which indicates stability as τ_2 grows until the algorithm no longer can compute a solution.

γ_1 variations

The next variable considered was the disturbance isolation constant γ_1 in order to consider the effects it has on the error and control signals. As displayed in Fig. 4.9

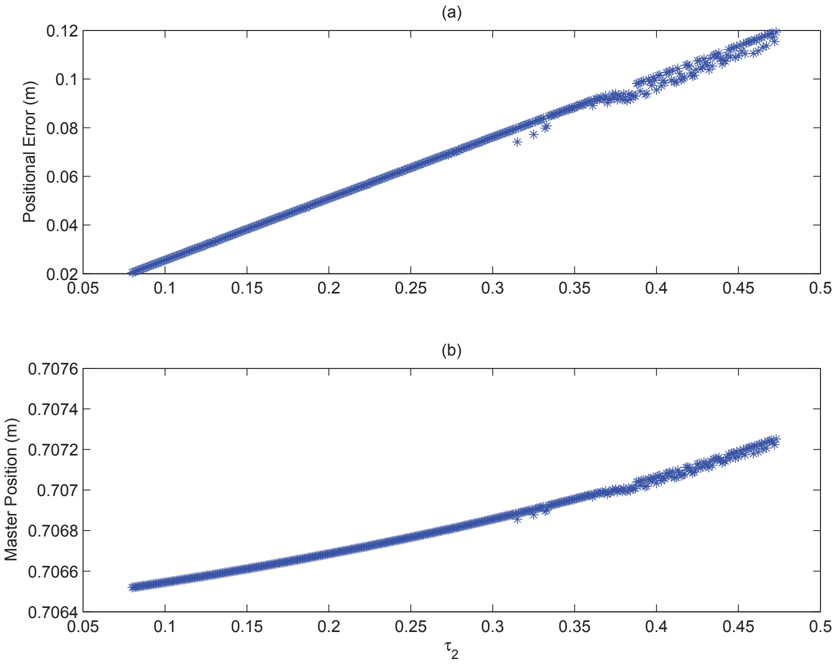


Figure 4.7: (a) Maximum positional error and (b) Maximum master position as τ_2 varies

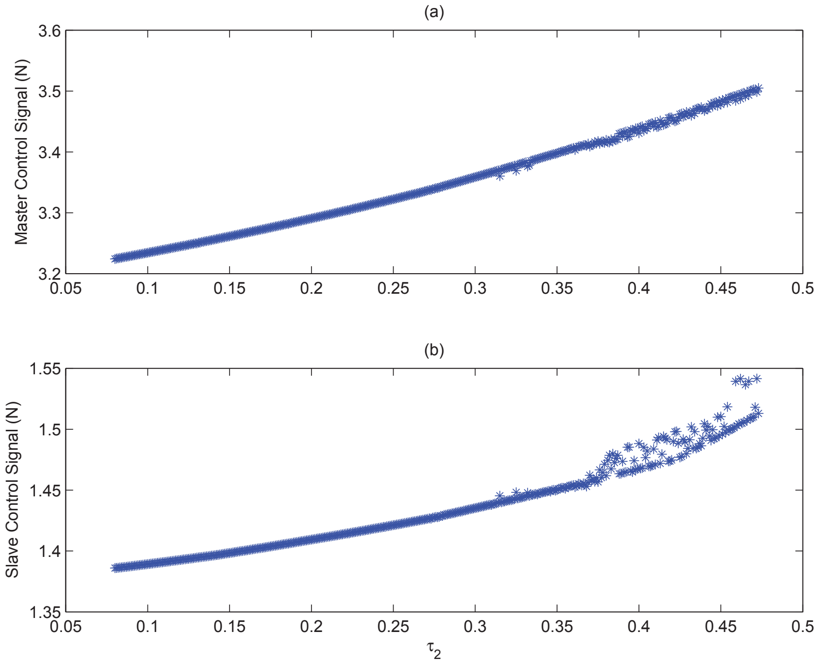


Figure 4.8: (a) Maximum master control signal and (b) Maximum slave control signal as τ_2 varies

(a), the maximum error is relatively unchanged by the changes in γ_1, with an overall trend of decline as γ_1 increases. Once again the position is unaffected by the choice of γ_1, as shown in Fig. 4.9 (b). Plotted in Fig. 4.10 (a) and (b) are the maximum master and slave control signals respectively. As shown, the changes in γ_1 have little impact on the magnitude of the signals, with peaks of 3.213 N for the master and 0.95 N for the slave. These peaks are comparable to the maximum input force of 3 N.

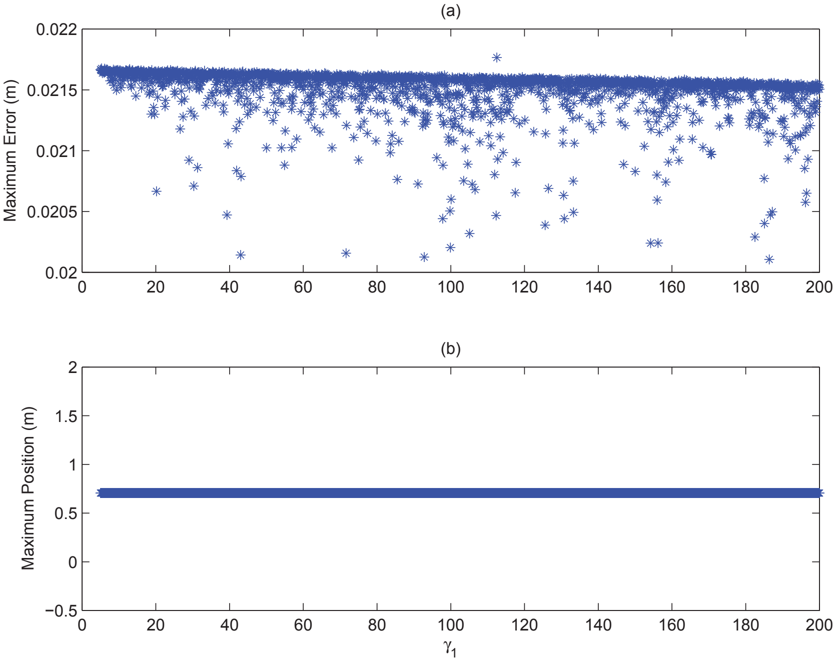


Figure 4.9: (a) Maximum positional error and (b) Maximum master position as γ_1 varies

γ_2 variations

The final variable that was in the LMIs was γ_2, and sweeps were conducted on the parameter value similar to γ_1 and τ_2. Fig. 4.11 (a) shows the maximum value of the error and an odd step response behavior. The low initial values of error are expected based on (3.14), but the peak at shortly thereafter indicates that the behavior of the other LMI variable must be more dependent on γ_2 than γ_1. This is because there were no real variations in the γ_1 maximum error values. Fig. 4.9 (b) displays the maximum position reached by master manipulator, which remains constant at 0.7 m.

Plotted in Fig. 4.12 (a) and (b) are the maximum master and slave control signal

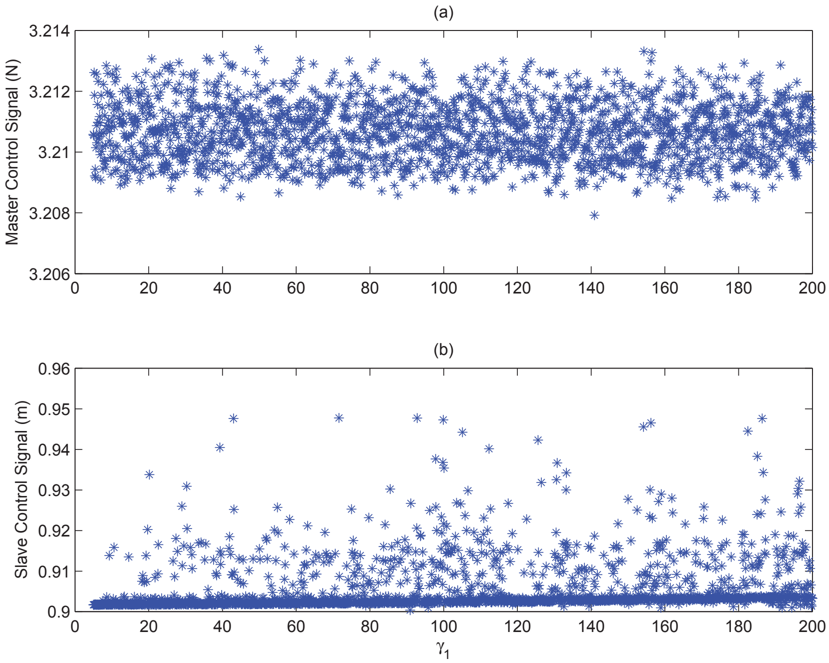


Figure 4.10: (a) Maximum master control signal and (b) Maximum slave control signal as γ_1 varies

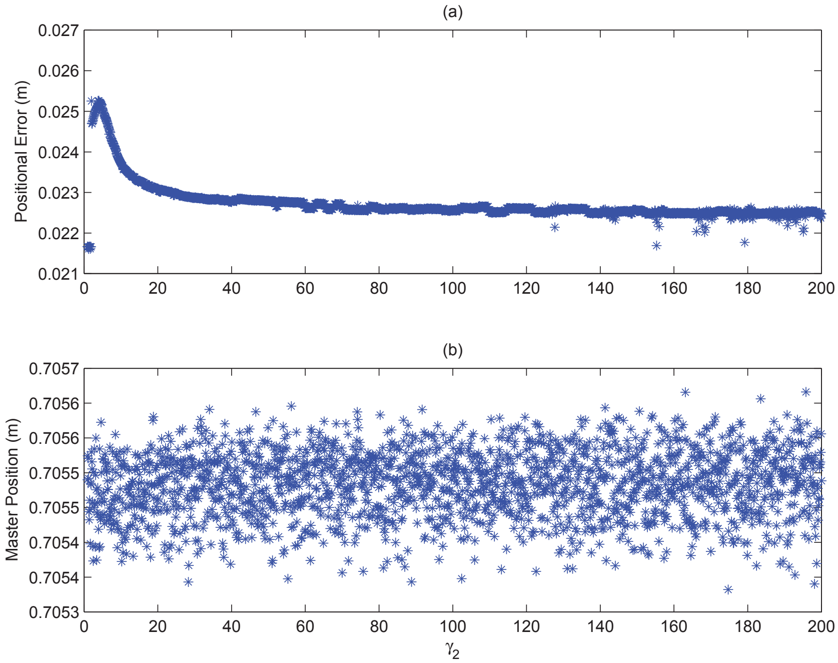


Figure 4.11: (a) Maximum positional error and (b) Maximum master position as γ_2 varies

values. The master signal appears to be unaffected by variations in γ_2 at a value around 3.213 N. However, the values of the slave control signal appear to be dependent on γ_2, with a general first order step response appearing as γ_2 increases.

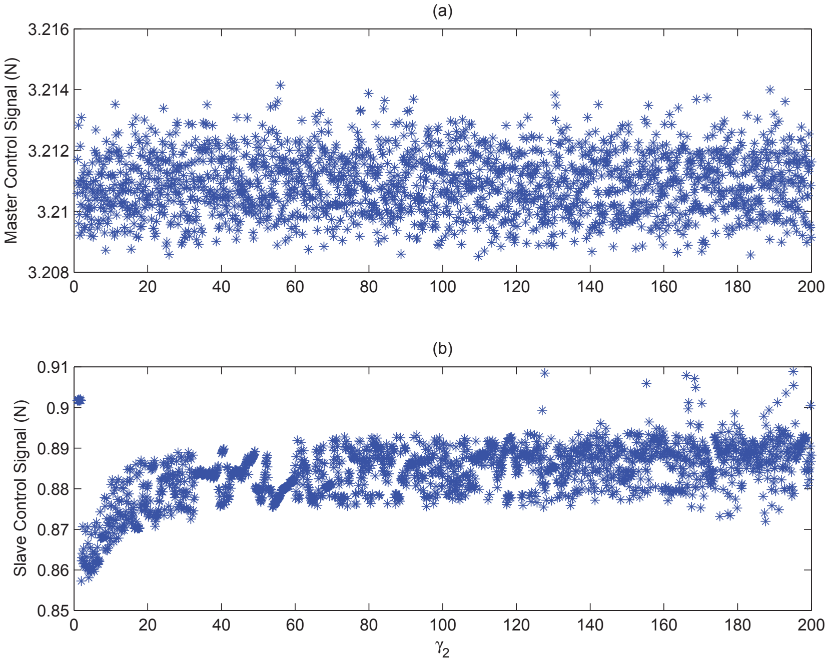


Figure 4.12: (a) Maximum master control signal and (b) Maximum slave control signal as γ_2 varies

A study was done to relate the maximum positional error to various combinations of γ_1 and γ_2, see Fig. 4.13. The results indicate a much stronger relationship between the error and γ_2 based on the curvature mimicking the previous results for γ_2 variations. The variations of γ_1 produce a small effect, as made evident by the gradual tapering slope. The optimal point is located when both γ_1 and γ_2 are minimized.

4.3 Case II: Upper and Lower Bound on Time Delays

The second case considered contains both an upper bound of τ_2 and a lower bound of τ_1 on the time delay, with the delay variable between τ_1 and τ_2. It was expected that the results will vary due to the increased level of communication modeling.

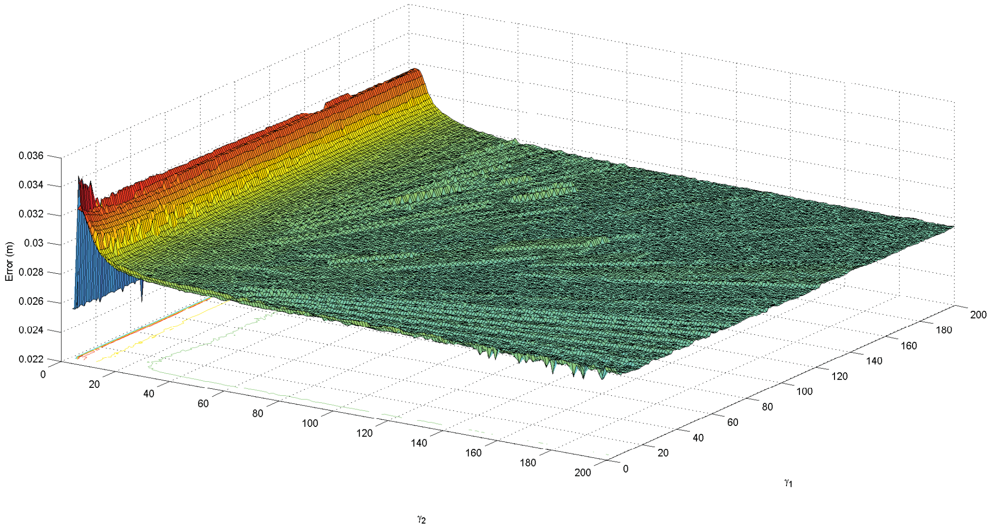


Figure 4.13: Error as γ_1 and γ_2 varies

4.3.1 Base LMI Results

Using the system parameters from Section 4.1, the LMI code((3.31)-(3.35)) was run to achieve the following LMI variables

$$\begin{aligned}
X &= 1 * 10^{-4} \begin{bmatrix} 0.6359 & -0.1280 \\ -0.1280 & 0.6458 \end{bmatrix} \\
Y &= 1 * 10^{-3} \begin{bmatrix} 0.2113 & -0.0983 \end{bmatrix} \\
\hat{R}_1 &= 1 * 10^{-4} \begin{bmatrix} 0.3628 & -0.0995 \\ -0.0995 & 0.3717 \end{bmatrix} \\
\hat{R}_2 &= 1 * 10^{-4} \begin{bmatrix} 0.5812 & 0.0322 \\ 0.0322 & 0.4078 \end{bmatrix} \\
\hat{S} &= 1 * 10^{-4} \begin{bmatrix} 0.3230 & -0.0868 \\ -0.0868 & 0.3309 \end{bmatrix} \\
\hat{Q}_1 &= 1 * 10^{-3} \begin{bmatrix} 0.0183 & 0.0206 \\ 0.0206 & 0.1406 \end{bmatrix} \\
\hat{Q}_2 &= 1 * 10^{-3} \begin{bmatrix} 0.0169 & 0.0220 \\ 0.0220 & 0.1435 \end{bmatrix} \\
\gamma_1 &= 1.1 \\
\gamma_2 &= 0.1
\end{aligned}$$

which led to a static feedback control gain of $K_c = \begin{bmatrix} 3.1411 & -0.8997 \end{bmatrix}$.

4.3.2 Modified Step Response

The first input type tested on the system was a modified step response. The input forcing function F_h was modeled as a ramp function up to a value of 1 N with a slope of 0.2 N/s. After reaching a value of 1 N the force was held constant as in Fig. 4.14 (a). The corresponding control signals are displayed in Fig. 4.14 (b), and show relatively smooth behavior with no extreme control inputs or discontinuities present which is a desired trait for smooth teleoperation. The net force seen by the manipulators are in Fig. 4.14 (c). The plot shows slightly noisier signals being sent to the manipulator, but a low net value of force (all less than 0.2 N). Also shown is the fact that equilibrium with the environment is achieved after approximately 7 seconds, when the net manipulator forces go to zero. It should be noted that it is not necessary for F_h and F_e to be equal for equilibrium to be reached due to the asymmetry in the system and the scaling for F_h for the impedance matching controller.

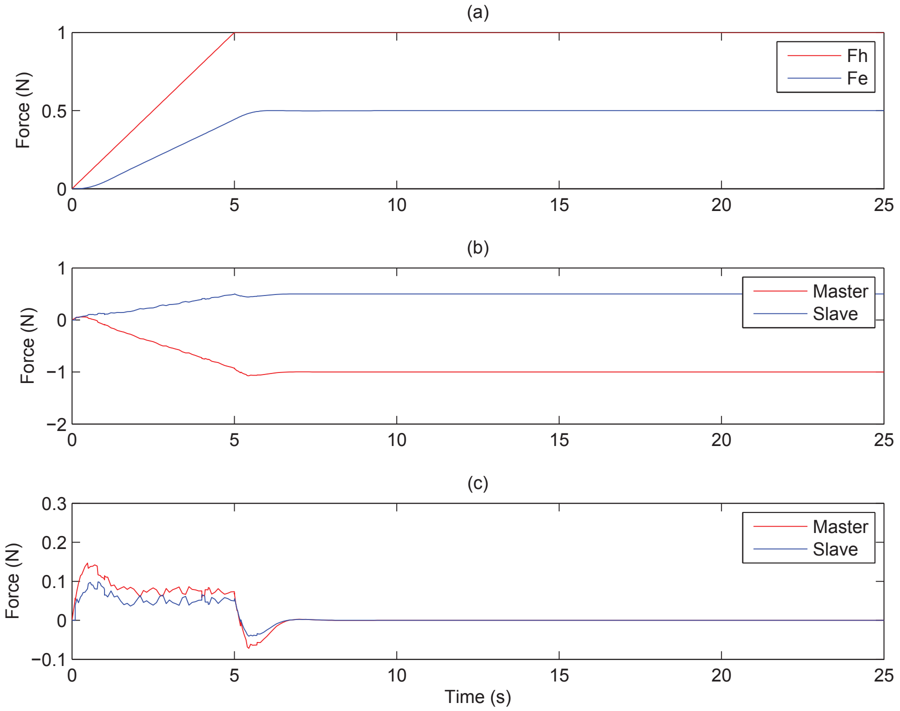


Figure 4.14: (a) External forces, (b) Control signals and (c) Net manipulator forces under a modified step input with lower delay bound τ_1 and upper delay bound τ_2

Similar to the forces plotted in Fig. 4.14, the state information is displayed in Fig. 4.15. The positions of the manipulators are shown in Fig. 4.15 (a) and displays the desired tracking between master and slave. Also shown is a final equilibrium position of 0.125 m. Fig. 4.15 (b) displays the velocities of both manipulators with the expected ramping to an approximately constant velocity to match the input, and then deceleration back to no movement when force equilibrium is occurring on the manipulators. Finally, Fig. 4.15 (c) displays the positional error between the master and slave manipulators which peaks at 4.2 mm, giving a maximum error of 3.36 % relative to the final position.

The results of Case II are are slightly worse than Case I as there is more error and a higher percent error, however the increased level of modeling lends more confidence in the results. The level of error increasing from 3.4 mm to 4.2 mm represents a percent increase of 23.5 %, a significant jump, but an increase compared to maximum travel of 0.56%, a less significant and more manageable increase. The better performance in Case I without the lower bound can be attributed to the packets that were received before the minimum delay level in Case II.

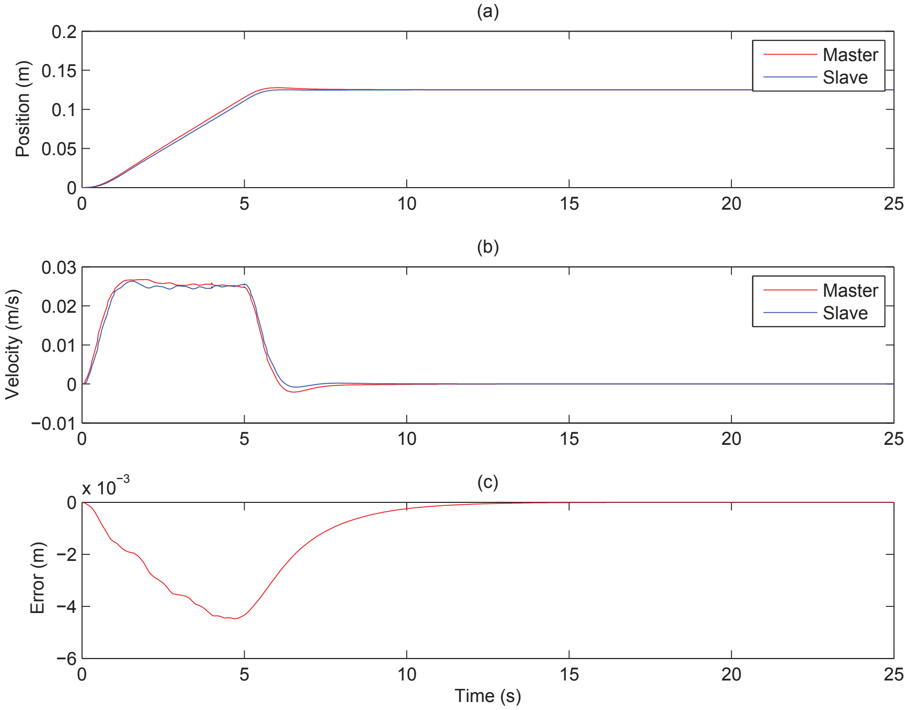


Figure 4.15: (a) Position, (b) Velocity and (c) Positional error under a modified step input with lower delay bound τ_1 and upper delay bound τ_2

Lastly, Fig. 4.16 plots the positional error with reference to maximum error bound as derived in (3.28). The error lies within the calculated bound limits of the algorithm, further verifying the ability to provide a stable bilateral teleoperation setup. Only the maximum error bound is displayed due to the extremely noisy behavior of the bound. The bound varies with the external forces F_h and F_e, as well as the true level of time delay τ in the system. The noisy behavior is due to the randomly varying delay values that cause variations in the differences of $F_h(t-\tau) - F_h(t)$ and $F_e(t-\tau) - F_e(t)$.

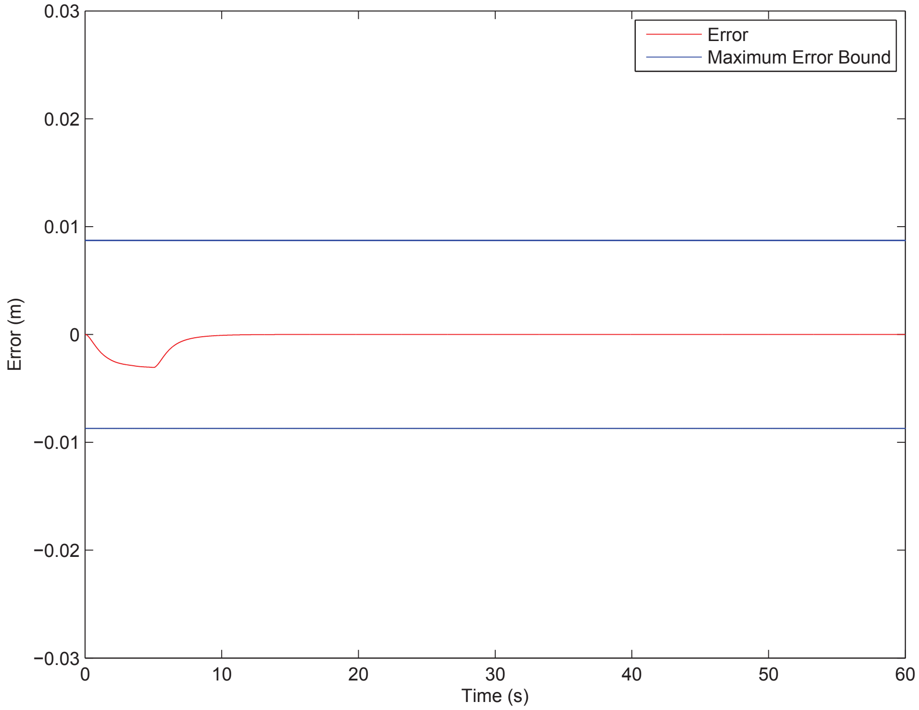


Figure 4.16: Positional error and its maximum bound under a modified step input with lower delay bound τ_1 and upper delay bound τ_2

4.3.3 Sinusoid Response

A sinusoid was chosen as a second human input for testing as it has a constantly changing force. For the simulations, a sinusoid with a magnitude of 3 N and a frequency of 0.5 Hz was used, which falls within the 6 Hz limit of voluntary human motion [55]. The input is shown in Fig. 4.17 (a), along with the wall contact force. The contact force occurs only when the position of the system is positive as it is assumed the slave manipulator is resting against the object of interest. Fig. 4.17 (b) shows the control signals from the master side impedance matching controller and slave side stability controller. The control signals are relatively smooth with no discontinuities or extreme force values, as all values falling within ±4 N. Fig. 4.17 (c) displays the net force as seen by the manipulators, and once again the signal is noisier than the control signal but again is bounded, this time by a magnitude of 1.5 N.

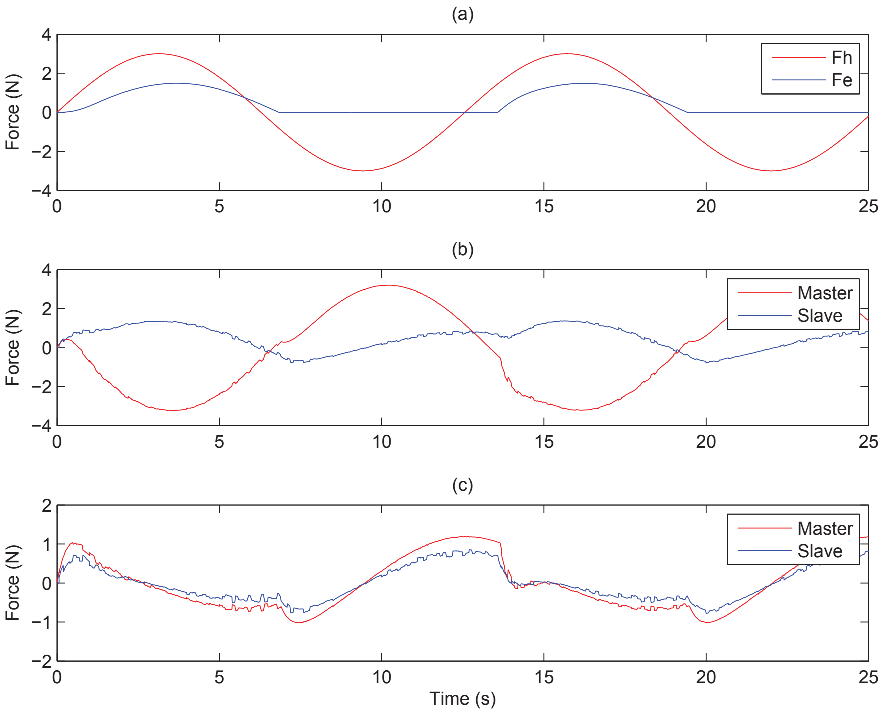


Figure 4.17: (a) External forces, (b) Control signals and (c) Net manipulator forces under a sinusoidal input with lower delay bound τ_1 and upper delay bound τ_2

Fig. 4.18 (a) plots the positions of both the master and slave manipulators and displays the level of tracking available in the system. It is noted that the level of error is difficult to discern from Fig. 4.18 (a) alone. Displayed in Fig. 4.18 (b) are the respective velocities that show relatively smooth motion. Smoothness is desirable to avoid high levels of overshoot in the slave tracking performance that would cause unnecessary fluctuations in the contact force F_e. Lastly, Fig. 4.18 (c) plots the error in position of the slave from the master. As shown, a peak error of 0.03 m is realized compared to a maximum distance from equilibrium of 0.7 m, yielding a percent error relative to maximum travel of 4.3 %.

When compared to Case I, the percent error of Case II is higher, primarily due to the packets that previously would have arrived prior to 0.02 s being delayed to 0.02 s or higher.

Fig. 4.16 plots the positional error with reference to maximum error bound as derived in (3.28). The error lies within the calculated bound limits of the algorithm, further verifying the algorithm's ability to provide a stable bilateral teleoperation setup. The maximum error bound is displayed due to the extremely noisy behavior of the bound. The bound varie with the external forces F_h and F_e, along with the true level of time

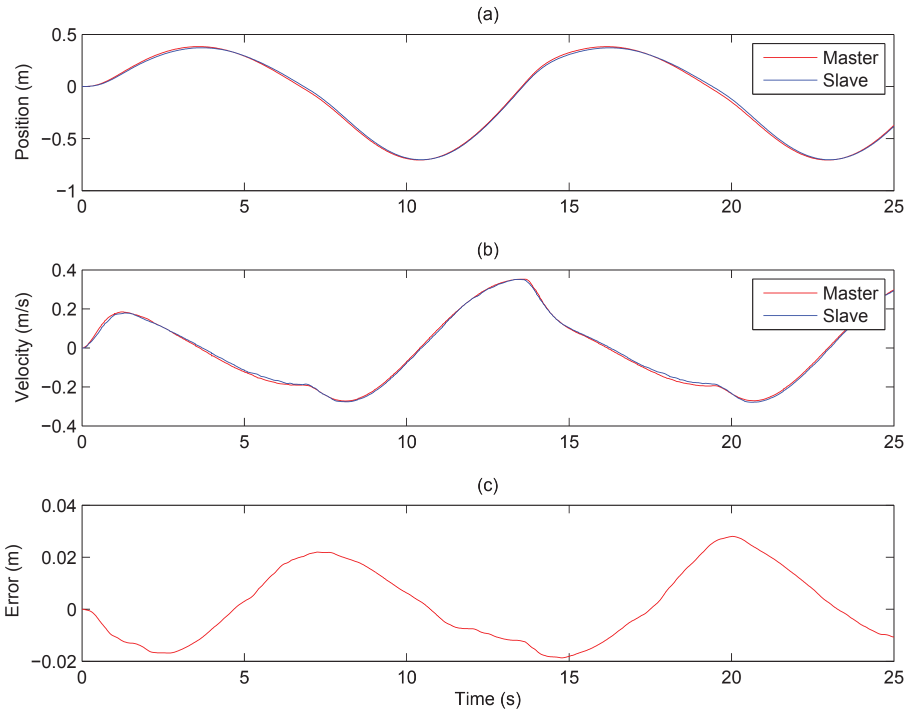


Figure 4.18: (a) Position, (b) Velocity and (c) Positional error under a sinusoidal input with lower delay bound τ_1 and upper delay bound τ_2

delay τ in the system. The noisy behavior is due to the randomly varying delay values that cause variations in the differences of $F_h(t-\tau) - F_h(t)$ and $F_e(t-\tau) - F_e(t)$.

4.3.4 Parametric Study

It was determined that a parametric study on the fixed variables τ_1, τ_2, γ_1, and γ_2 should be conducted as to explore their effects on the error and control signals. It was expected that increasing these values would increase the overall system error based on (3.28), although variations in the LMI solutions would possibly cause other behavior. All of the system parameters are the same as in Section 4.1, with the input being the same sinusoidal function as in Section 4.3.3.

τ_1 variations

The lower delay bound τ_1 was the first variable considered during the parametric study. It was expected, based on the previous results from Sections 4.3.3 and 4.2.3, that as τ_1 increased under a constant τ_2 that the level of error would increase. As shown in Fig. 4.20 (a), the maximum error did indeed increase as τ_1 increased. A second factor considered was the positional change due to the variation of τ_1. As

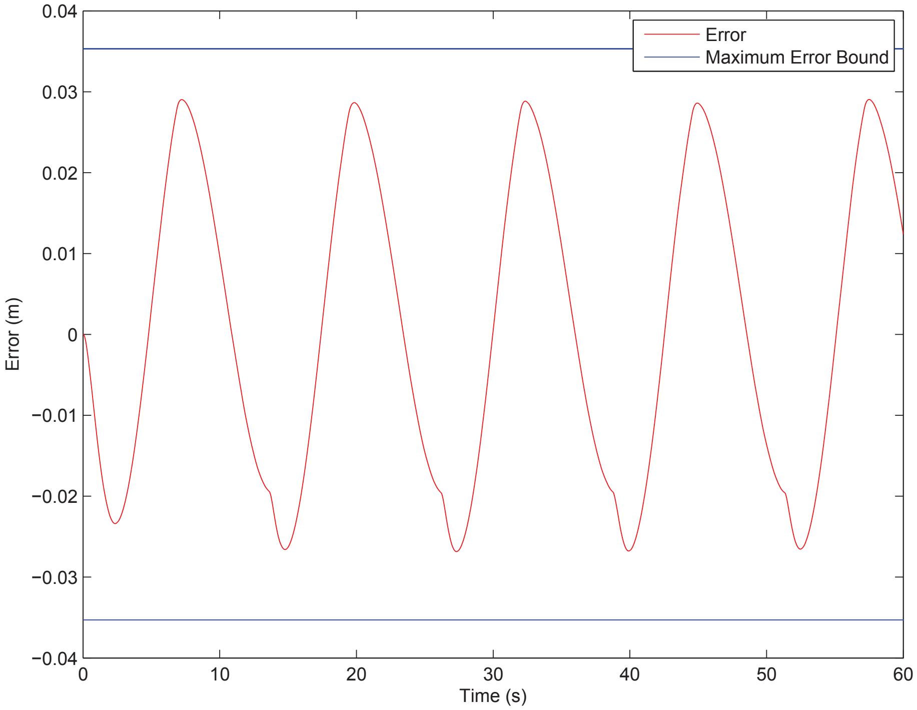


Figure 4.19: Positional error and its maximum bound under a sinusoidal input with lower delay bound τ_1 and upper delay bound τ_2

shown in Fig. 4.20(b), the behavior does not exist as while there was only a slight change in value, however the peak is essentially the same.

The second set of factors to explore is the control signals to ensure they are stable. If the position is stable then the magnitude of F_e is likewise stable due to its dependence on position, and F_h is guaranteed to be stable as it is defined by the user, therefore the only source of force spikes are the control signals. Fig. 4.21 shows the maximum resulting forces for (a) master and (b) slave. Based on the plots, it can be seen that the forces are bounded by similar bounds, and that trends do exist where the master signal increases with τ_1 and the slave signal decreases with τ_1.

τ_2 variations

Recalling that the maximum level of delay from Case I was 0.46 s, in finding the maximum allowable delay for Case II the hope was it would be larger due to the more in depth modeling of the communication channels. As displayed in Fig. 4.22 (a), the maximum allowable delay has been increased to 1.236 s, while the maximum error shows a similar trend, as it grew along with τ_2 from 0.03 m to over 0.4 m. This is an increase in percent error from 2.85 percent to 57 percent. Unlike Case I, the

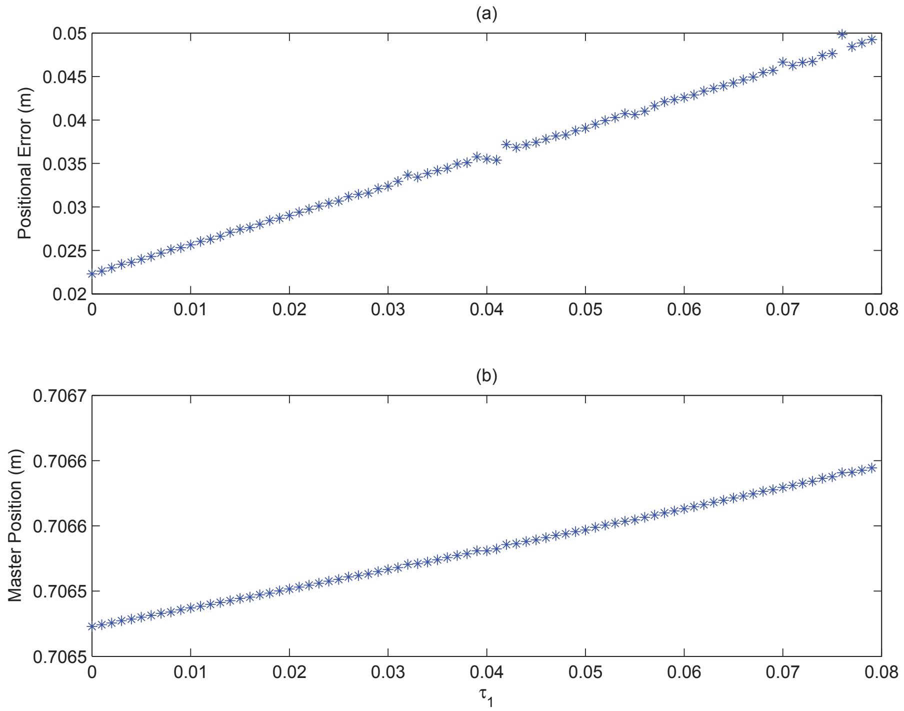


Figure 4.20: (a) Maximum positional error and (b) Maximum master position as τ_1 varies

maximum master position varies slightly with τ_2, since it can be seen to grow with τ_2 in Fig. 4.22(b).

It is necessary to check the control signals to ensure there are no unwanted control signal peaks. Fig. 4.23 shows the maximum master control signal (a) and the maximum slave control signal (b). It can be seen that for both cases, the control signal maximum is kept within a reasonable bound. The master maximum control signal was bounded by 4.4 N while the slave by 1.6 N, which are comparable to the 3 N input maximum. The patterns shown by the signals are of interest, as the master signal continues to grow as τ_2 grows, while the slave signal is somewhat parabolic with a peak occurring at 0.73 s.

γ_1 variations

After concluding the testing of τ_1 and τ_2, similar testing was done on the gain γ_1. As shown in Fig. 4.24 (a), other than in the first few values ($\gamma_1 < 10$) there is little effect on the error as γ_1 varied. Overall the error increased by 0.004 m, or 12 %, before it leveled off. Based on the fact the error bound is a function of γ_1, ot was expected that as γ_1 increased the error would increase. However, instead the LMI compensates for

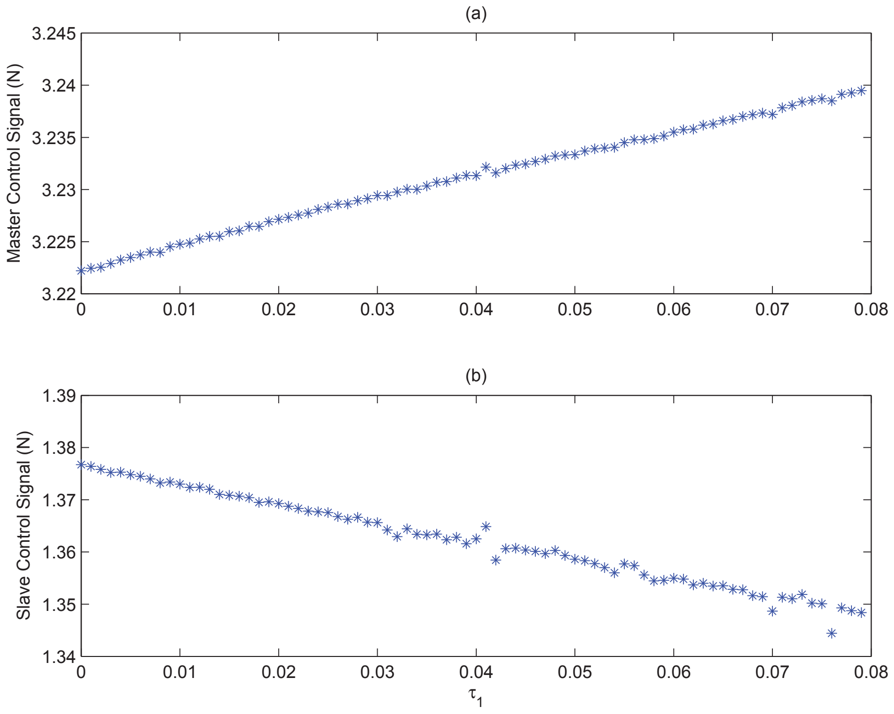


Figure 4.21: (a) Maximum master control signal and (b) Maximum slave control signal as τ_1 varies

the changes in γ_1 by adjusting other parameters such as R1, R2 and S. Secondly, it can be seem that the maximum position that the manipulator reaches is independent of γ_1, and that all values are bounded by 0.707 m.

The second factor to consider are the control signals generated when γ_1 changes. The maximum values can be seen in Fig. 4.25. As shown, both control signals' maximums appear to decrease as γ_1 increases and then level off. Leveling occurs at 3.224N for the master and at 1.35N for the slave.

γ_2 variations

The final parameter to explore the effects of its variation was γ_2. As shown in Fig. 4.26 (a), the maximum error is dependent on the value of γ_2, with the maximum occurring when γ_2 is minimized. This indicates that other LMI parameters are increased to cause the additional levels of error. It appears that the ideal value of γ_2 is approximately 10, assuming that the other parameters are kept constant. It is also important to note that the value of maximum error appears to level off as γ_2 increases. As plotted in Fig. 4.26 (b), the maximum position achieved by the master manipulator remains unchanged at 0.706 m.

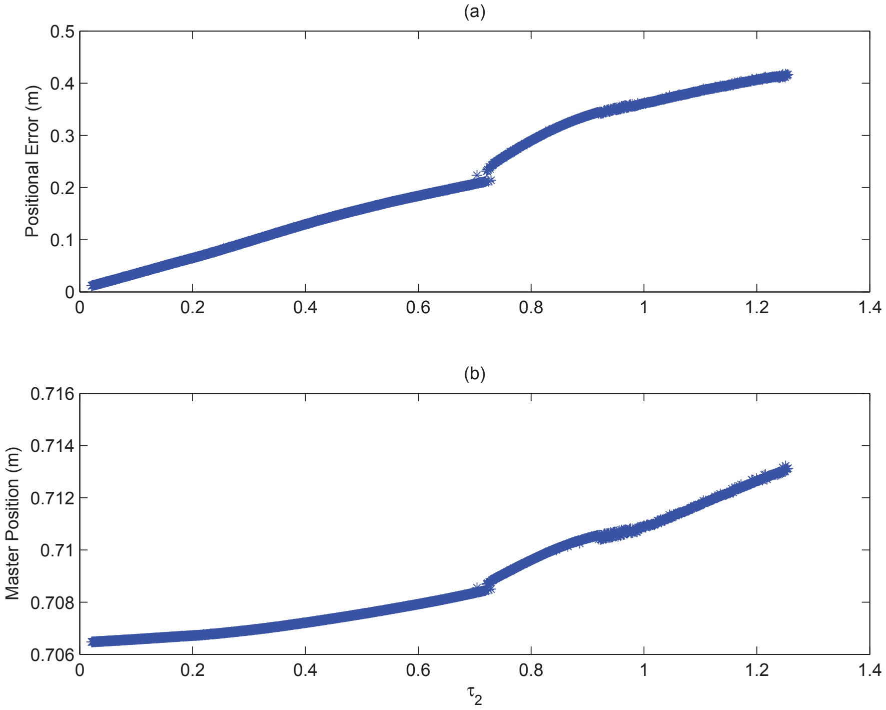


Figure 4.22: (a)Maximum positional error and (b) Maximum master position as τ_2 varies

The final factor to consider was the control signal peaks, which are plotted in Fig. 4.27. Fig. 4.27 (a) shows the master control signal which has no trends to it as far as γ_2 is concerned, fluctuating 3.235N. Shown in Fig. 4.27 (b) has little change based on γ_2, only a slight ramp effect when the values of γ_2 are quite small, and only over a range of 0.05N.

A study was conducted to relate the maximum positional error to various combinations of γ_1 and γ_2 and is displayed in Fig. 4.28. The plotted results indicate that both γ_1 and γ_2 have a strong effect on the error, particularly at low values. For γ_1, the error remains large, until a value of approximately 7, and then drops down to a much smoother plane. In the case of γ_2, the error remains large, until a value of approximately 36, and then drops off dramatically. Once on the smoother lower slope, the values of maximum error continue to lower as both γ_1 and γ_2 increase. A final point of interest is that when both γ_1 and γ_2 are small (γ_1=1 and γ_2=8) the error is minimized.

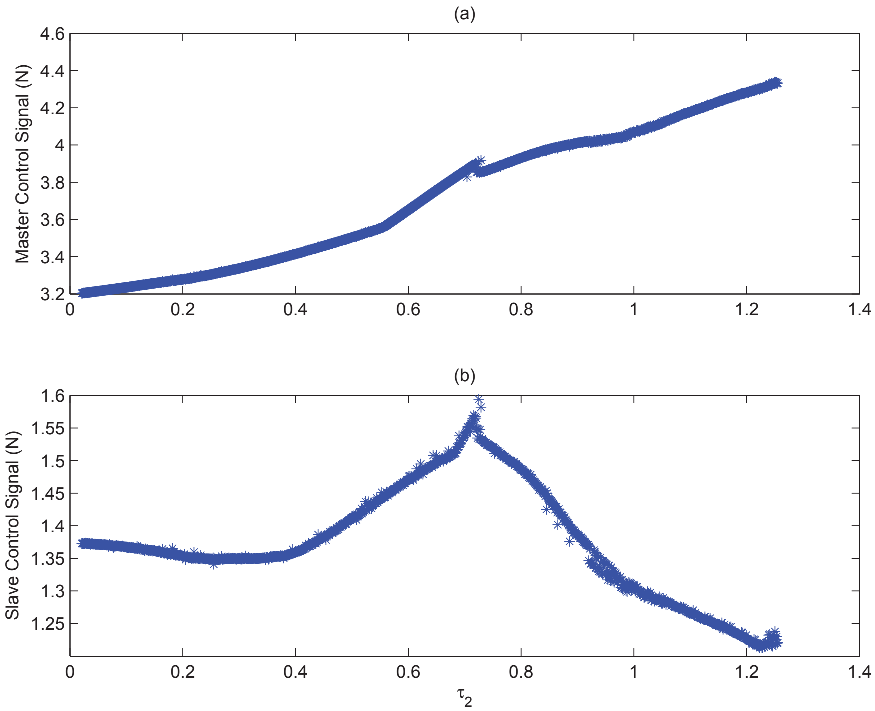


Figure 4.23: (a) Maximum master control signal and (b) Maximum slave control signal as τ_2 varies

4.4 Summary

In summary, the simulation results for a sample test system were explored and discussed for the two test cases: where only an upper bound on delay was considered, and where both an upper and lower bound were considered. The simulation results demonstrate tracking capabilities, with the maximum error for Case I at 0.03 m for a 5 N sinusoid input, while in Case II the maximum error was 0.03 m with an input of a 3 N sinusoid. A scenario with a modified step input leading to contact with a deformable surface was also explored, and showed that an equilibrium position occurred where no error existed. A parametric study was conducted for both cases in which the values of τ_1, τ_2, γ_1, and γ_2 were varied while the others were held constant in order to evaluate their effects on the system. It was shown that τ_2 had the greatest effect on the error followed by τ_1 for the case with the lower bound. The other parameters had a smaller effect, although γ_2 had a greater influence than γ_1 on the error and control signals.

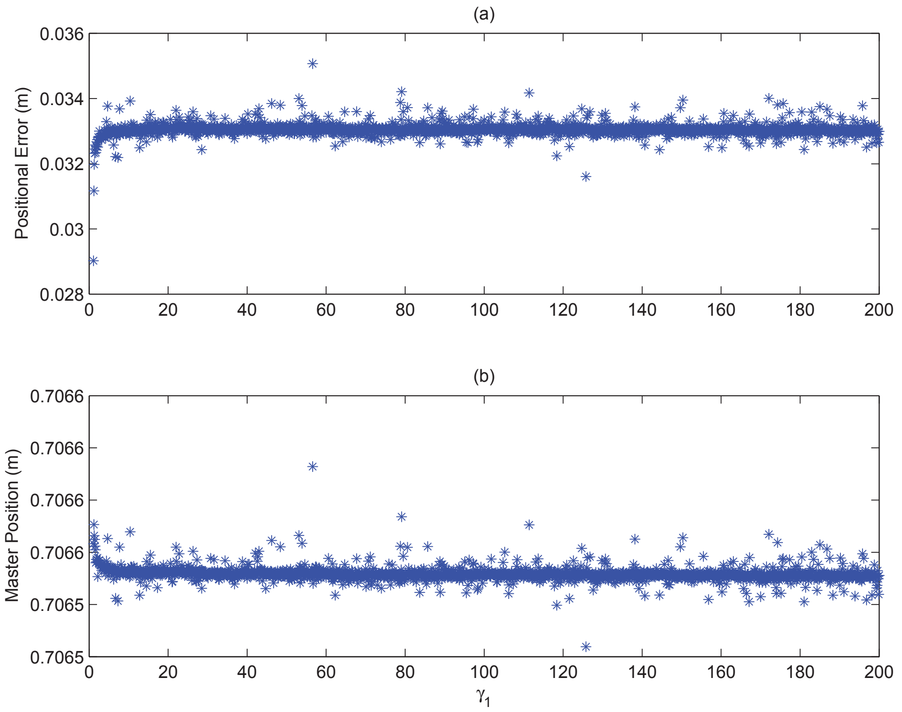


Figure 4.24: (a) Maximum positional error and (b) Maximum master position as γ_1 varies

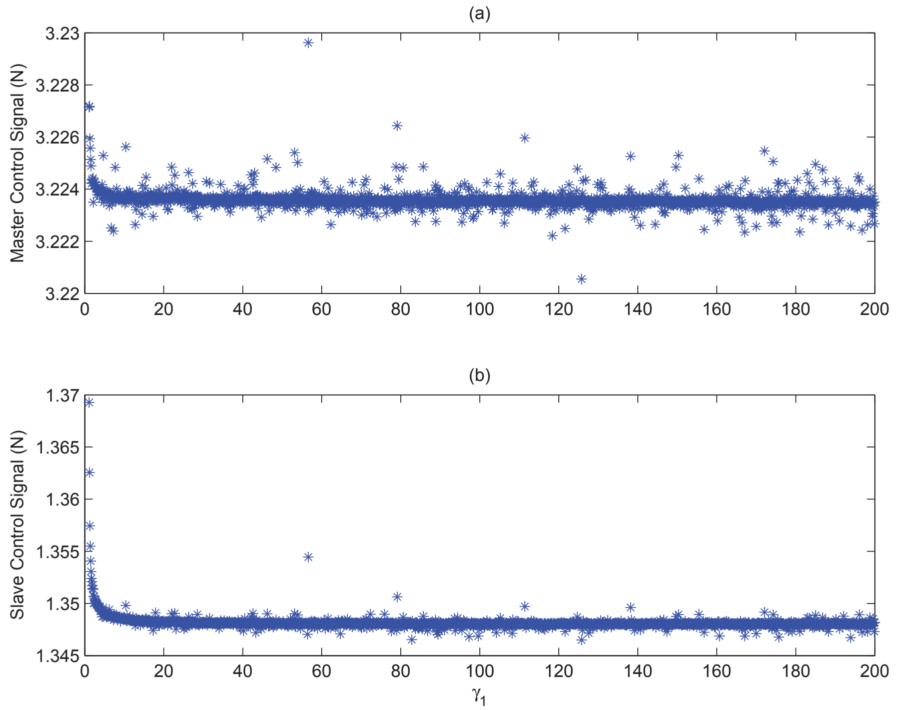


Figure 4.25: (a) Maximum master control signal and (b) Maximum slave control signal as γ_1 varies

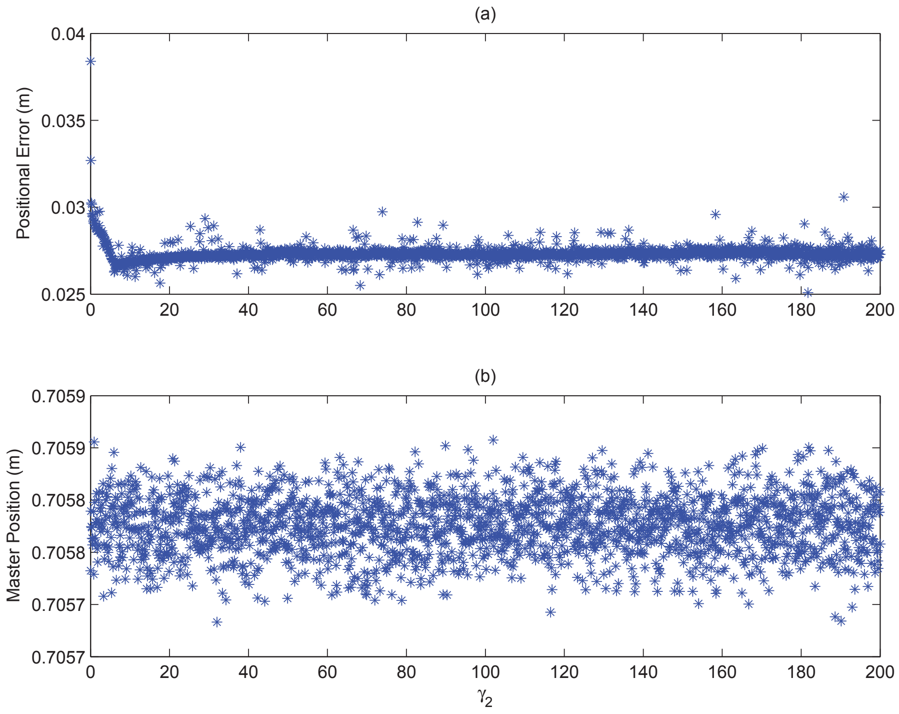


Figure 4.26: (a) Maximum positional error and (b) Maximum master position as γ_2 varies

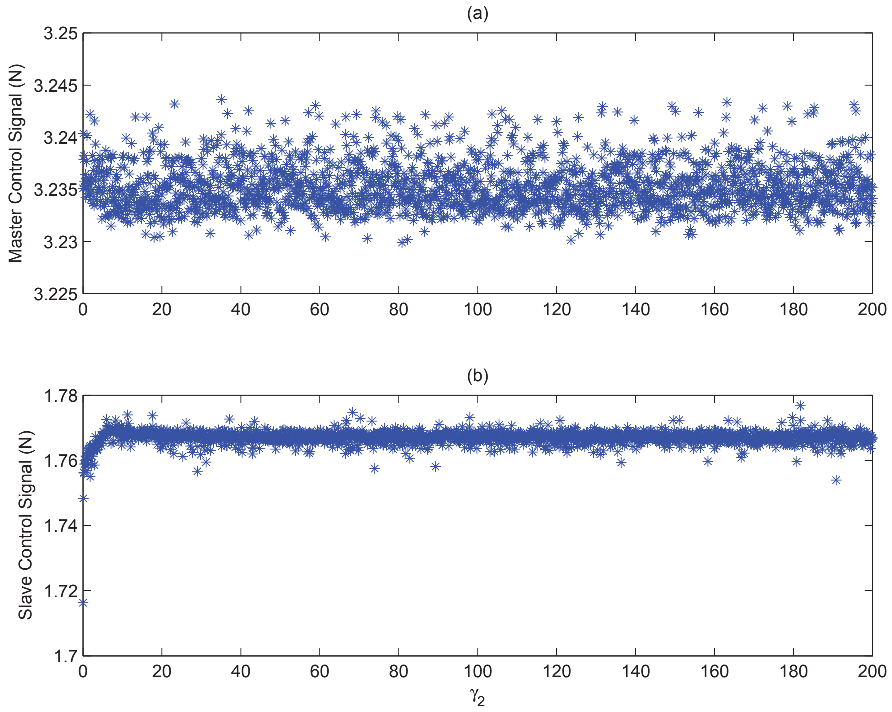


Figure 4.27: (a) Maximum master control signal and (b) Maximum slave control signal as γ_2 varies

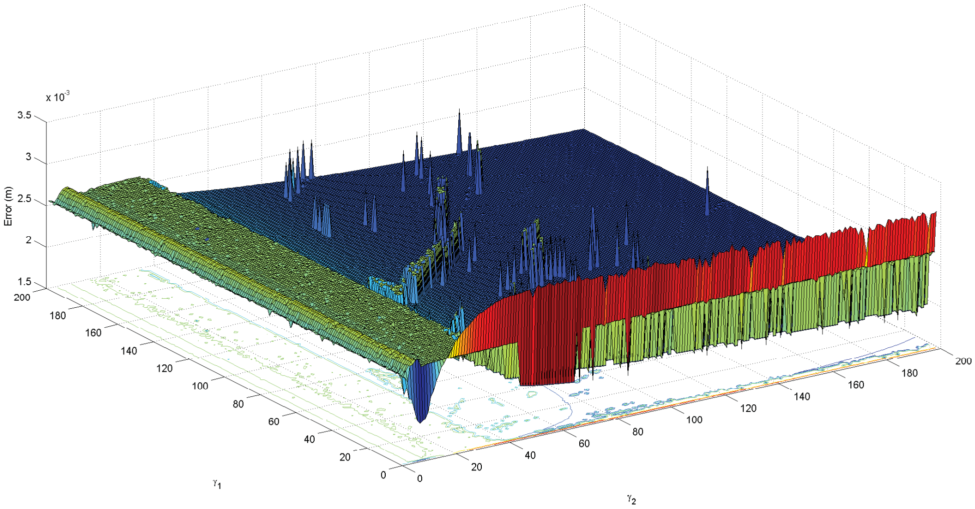


Figure 4.28: Error as γ_1 and γ_2 varies

Chapter 5

Experimental Work

For the experimental verification of the theoretical derivations, a bilateral teleoperation setup was required. For the test setup a Phantom Omni haptic device(Fig. 5.1(a)) made by Sensable was used as the master hardware, while a Novint Falcon haptic device(Fig. 5.1(b)) was used as the slave hardware. The Phantom Omni is marketed as a 6-degree of freedom manipulator with force feedback capabilities in three directions (x,y, and z). The Novint Falcon is a 3-degree of freedom manipulator with feedback for all directions of motion.

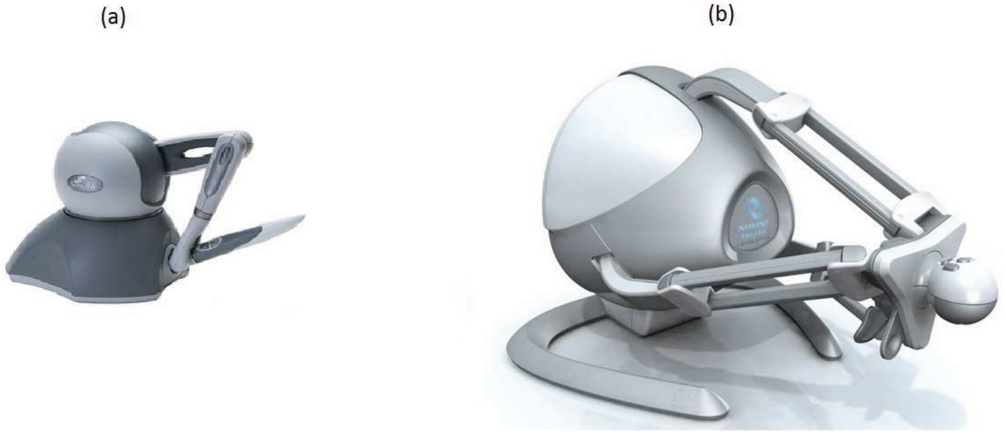
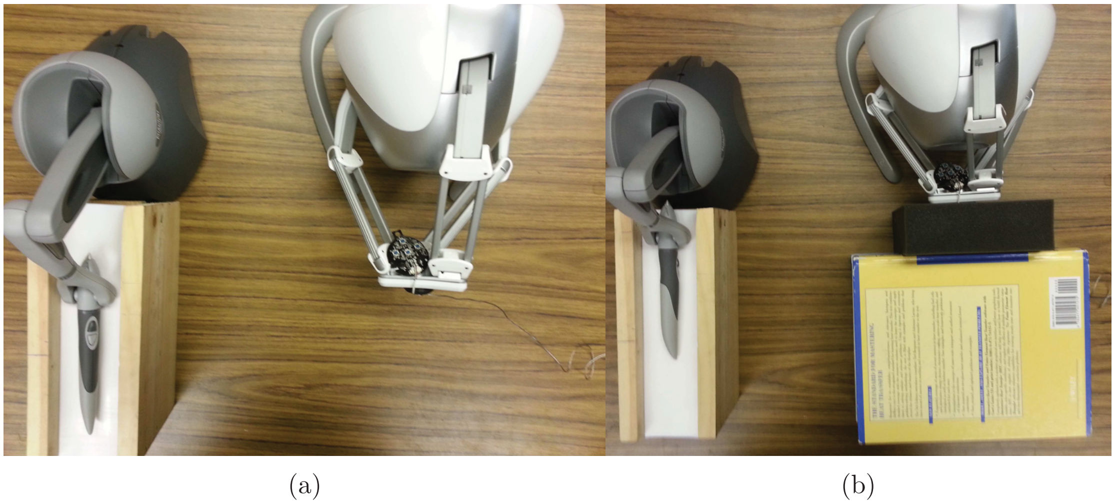


Figure 5.1: (a) Phantom Omni haptic device [62] and (b) Novint Falcon haptic device [61]

As the hardware is designed for multiple degrees of freedom and the theoretical derivations are for a single degree of freedom, modifications to the equipment were necessary. For the Phantom Omni, a brace was made to allow the system to operate with only a single degree of freedom. The Novint Falcon had a constant vertical force added to overcome gravitational forces as a simple brace could not be made for the device. The modifications are shown in Fig. 5.2, which displays the experimental setup for no contact (Fig. 5.2 (a)) and for when contact occurs (Fig. 5.2 (b)).



(a) (b)

Figure 5.2: Experimental hardware setup for (a) no contact and (b) contact

5.1 System Identification

Based on the derived control algorithm it was necessary to know the various system parameters for the controllers, and to also test them in the simulation environment to ensure safe operation. The method chosen to identify the manipulator models was the Observer Kalman Filter Identification (ERA/OKiD) algorithm [59]. ERA/OKiD is a system identification algorithm that uses a white noise signal with the system to be identified and measure the output from the system. Fig. 5.3 shows the general work flow of the algorithm. Step one is the computation of observer Markov parameters, where one should choose a number of parameters larger than the desired system order (at least 4 or 5 times). The next step is to compute the system Markov parameters, which simulate the system impulse response, and compute with the observer gain Markov parameters. The following step is to use the Eigenvalue Realization Algorithm to recover a state space model. The final step is to transform the realized model to modal coordinates to determine modal parameters if necessary or desired. For specific details on the processes see [59], as the focus of this work is on the control algorithm as opposed to system identification and modeling.

Test data was collected from the manipulators and is shown in Figs. 5.4 and 5.5. The magnitudes of the white noise input (3 N for the Omni and 5 N for the Falcon) were chosen such that the manipulators would move to collect useful data, but not reach the hardware limits. This was done to ensure that an accurate representation of the system dynamics would be captured.

Using the supplied data and the ERA/OKiD algorithm, a transfer function could

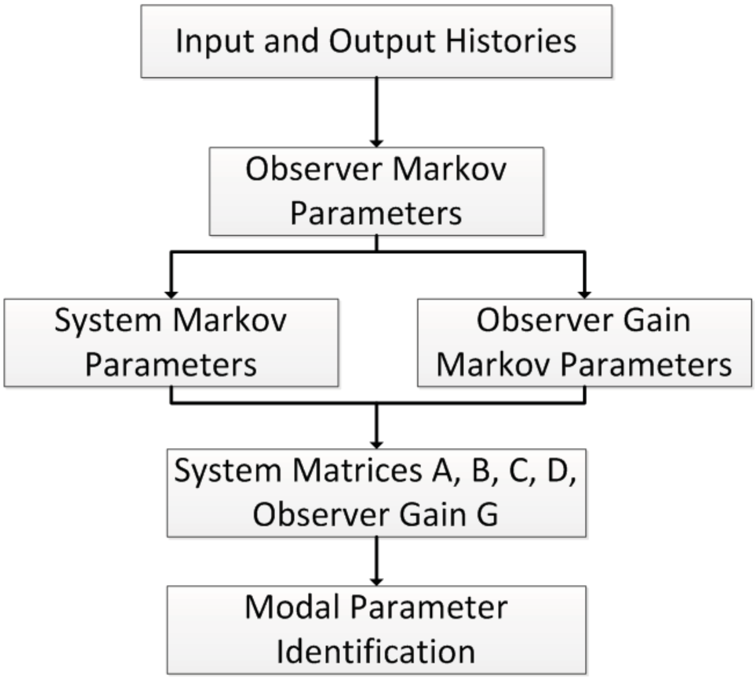

Input and Output Histories
Observer Markov Parameters
System Markov Parameters
Observer Gain Markov Parameters
System Matrices A, B, C, D, Observer Gain G
Modal Parameter Identification

Figure 5.3: ERA/OKiD flow chart [59]

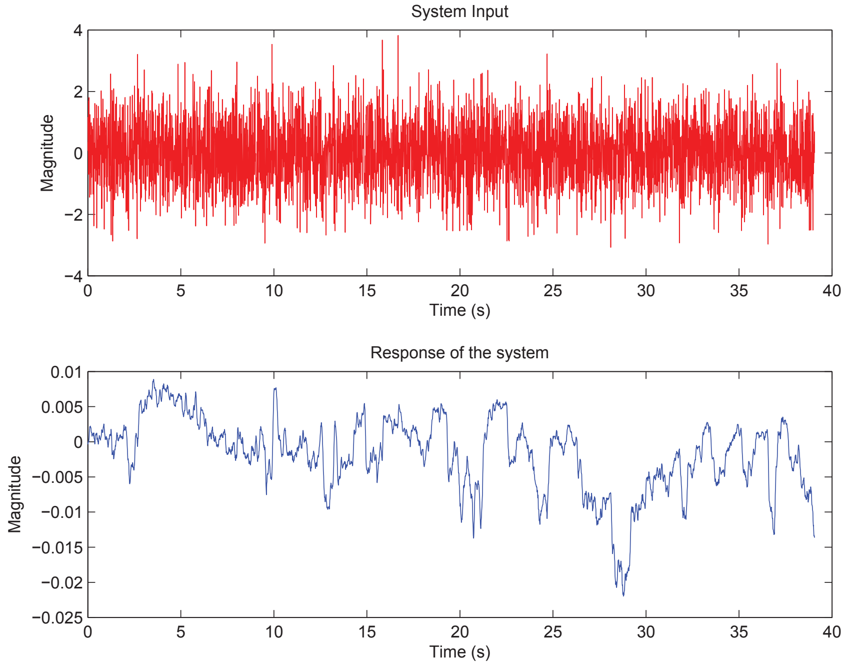


Figure 5.4: ERA/OKiD data for Phantom Omni device

be generated by the Matlab code (Appendix D). The input parameters for the code were 10 rows for the Hankel matrix, 10 columns for the Hankel matrix, 101 observer Markov parameters, and a system order of 2. From the transfer functions, the system

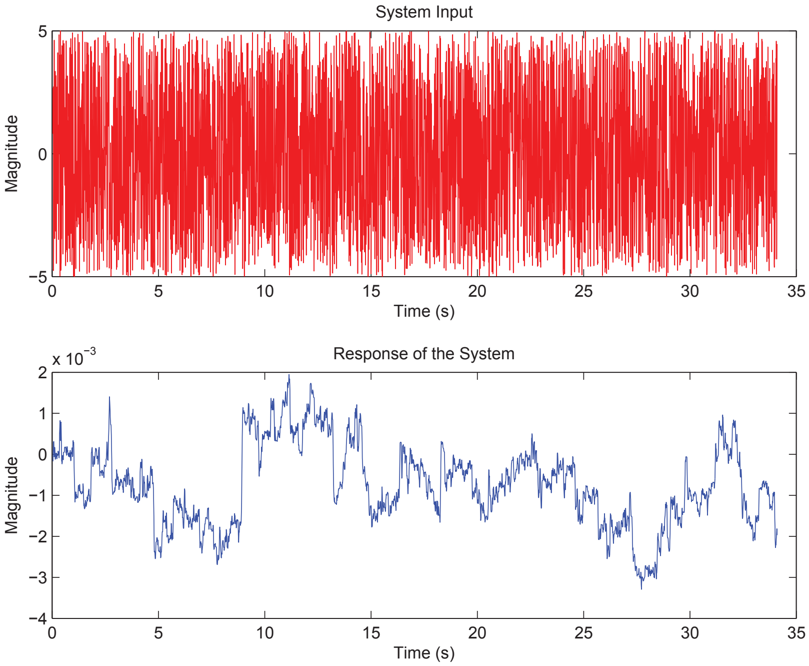


Figure 5.5: ERA/OKiD data for Novint Falcon device

properties could be determined by comparison to the standard transfer function for a single degree of freedom mechanical translational system (5.1).

$$G(s) = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} \tag{5.1}$$

From the ERA/OKiD algorithm with the data shown in Fig. 5.4 for the Phantom Omni, a corresponding transfer function (5.2) that can be compared to (5.1) was generated

$$G(s) = \frac{6.1227 \times 10^{-7}s^2 - 0.001268s + 4.4776}{s^2 + 77.252s + 28.192}. \tag{5.2}$$

The $6.1227 \times 10^{-7}s^2 - 0.001268s$ terms in the numerator can be neglected due to the orders of magnitude different from the rest of the terms. The system parameters can be approximated as

$$\begin{aligned} m &= 0.223\text{kg} \\ b &= 17.227\text{Ns/m} \\ k &= 6.286\text{N/m}. \end{aligned}$$

Similarly for the Novint Falcon a comparison of (5.3) to (5.1) using the data from Fig. 5.5 as the ERA/OKiD input was used to achieve the following transfer function

$$G(s) = \frac{4.6522 \times 10^{-7}s^2 - 0.003419s + 0.86376}{s^2 + 99.655s + 27.1689}. \tag{5.3}$$

As with the Phantom Omni, the first two terms of the numerator ($4.6522 \times 10^{-7}s^2 - 0.003419s$) are ignored due to the difference in the order of magnitude from the other numerator term. After the simplification, the following system parameters are obtained

$$\begin{aligned} m &= 1.158\text{kg} \\ b &= 115.40\text{Ns/m} \\ k &= 31.46\text{N/m}. \end{aligned}$$

As shown in Fig. 5.6, the model generated by the ERA/OKiD algorithm has some deviations from the actual output of the Phantom Omni when limited to a linear second order single degree of freedom model. While it is expected that variations between the model results and the actual results exist, the model does follow the same trends and lies within a similar magnitude bound. Initial attempts to identify the model were met with difficulty as the original system setup had a large amount of friction leading to greater deviations. Teflon was added to the sliding surface to reduce friction and to improve the results.

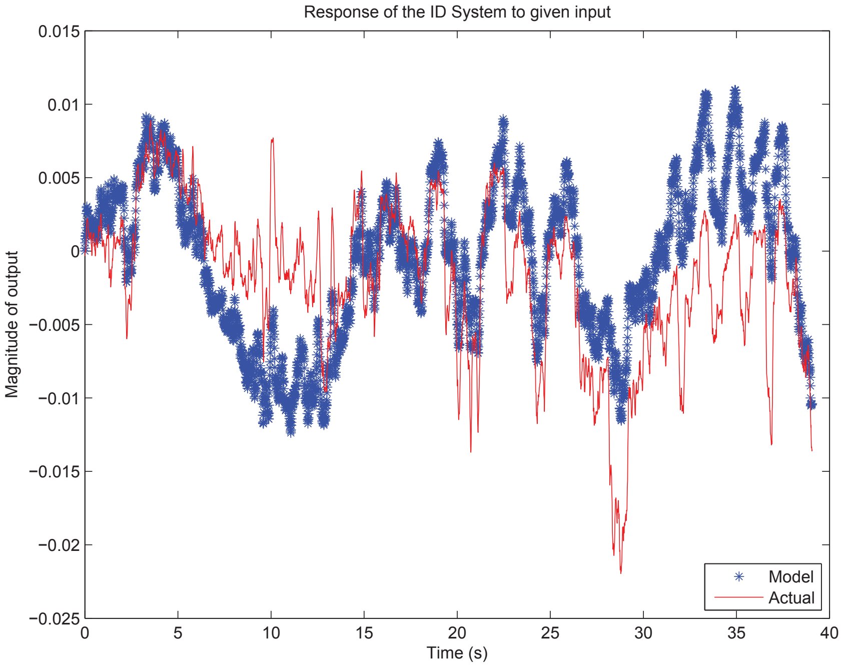


Figure 5.6: ERA/OKiD verification for Phantom Omni device

Displayed in Fig. 5.7 are the ERA/OKiD algorithm model results and actual test results for the Novint Falcon. Some deviations from the actual output of the Novint Falcon exist when limited to a linear second order single degree of freedom model. While it was expected that variations between the model results and actual results would exist, the level of deviation seems to be in an acceptable range considering the 3-degree of freedom to linear single degree of freedom, and non-linear to a linear model simplifications.

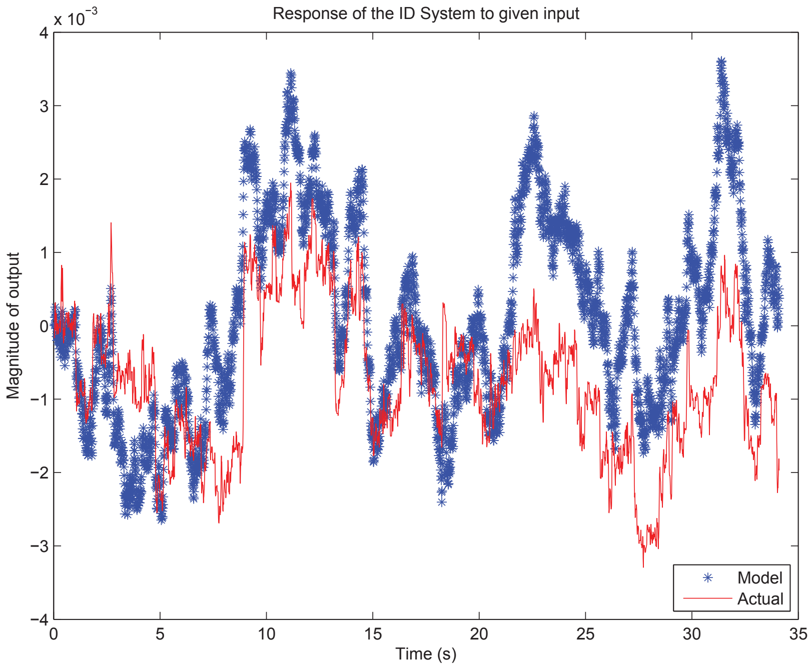


Figure 5.7: ERA/OKiD verification for Novint Falcon device

5.2 Model Verification

While the ERA/OKiD verification can help determine model appropriateness, a second method of verification is to apply the same input to both the generated model in Simulink and the real hardware and compare the results. Both a sinusoidal input and ramp inputs from various points in the operating range were used to explore if different spots behave differently (non-linearities) and to determine minimum force values required for motion.

Fig. 5.8 shows the results of applying the same sinusoidal input to both the hardware and the models generated by the ERA/OKiD system identification process. As shown, the hardware appears to have a minimum required force to cause motion (to be expected because of frictional effects), while the theoretical models do not have this phenomenon.

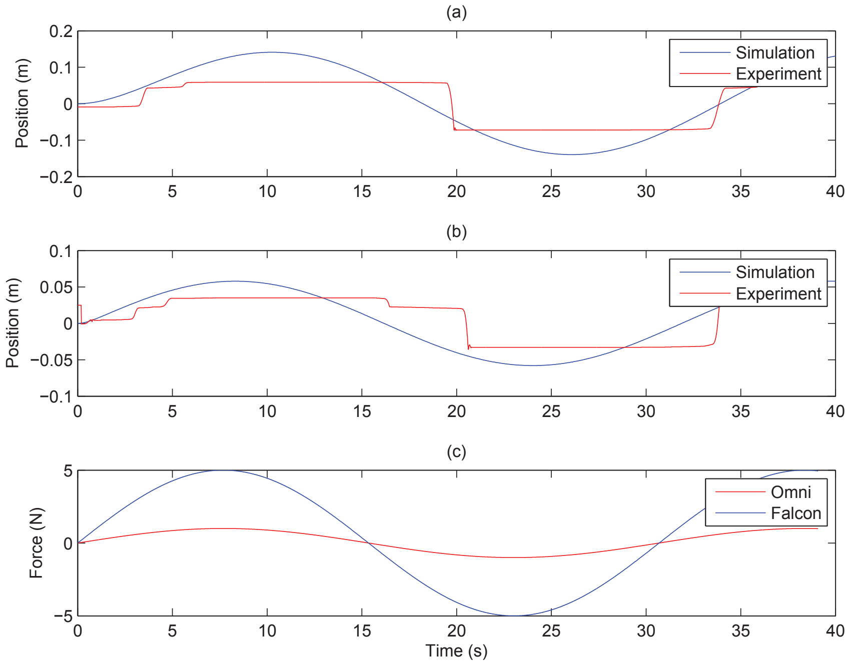


Figure 5.8: Sinusoidal verification tests (a) Omni positions, (b) Falcon positions, (c) Input force

Due to some of the jumps in motion in Fig. 5.8, it was decided to do some ramp testing at various positions in various directions to determine if the minimum force for movement varies with the location, and to better determine these values. Figs. 5.9 to 5.12 display the results of the ramp testing from various points in the system, with the model results shown for reference. The same ramp force of 0.2 N/s was applied in each case, with just the direction reversed for the negative case.

When considering the Phantom Omni, a wide range of motion start times were noted, ranging from approximately 3-5 s (0.6-1 N) for all but the negative ramp from the end, which took around 8 s (1.6 N) for motion to occur. The results were expected based on the linked nature of the manipulators, which suggested that more force would be needed for movement when the arms are fully extended (at the end), then when they are bent (all other positions).

When looking at the Novint Falcon, the range of motion start times begins at 12 s (2.4 N) for the negative ramp from the midpoint to a maximum of 27 s (5.4 N) for a negative ramp from the end. These results indicate that a non-linear type relationship, similar to the Phantom Omni, exists. However, it is not apparent why this relationship exists due to the absence of noticeably moving parts on the manipulator, and because of the parallel nature of the manipulator motion.

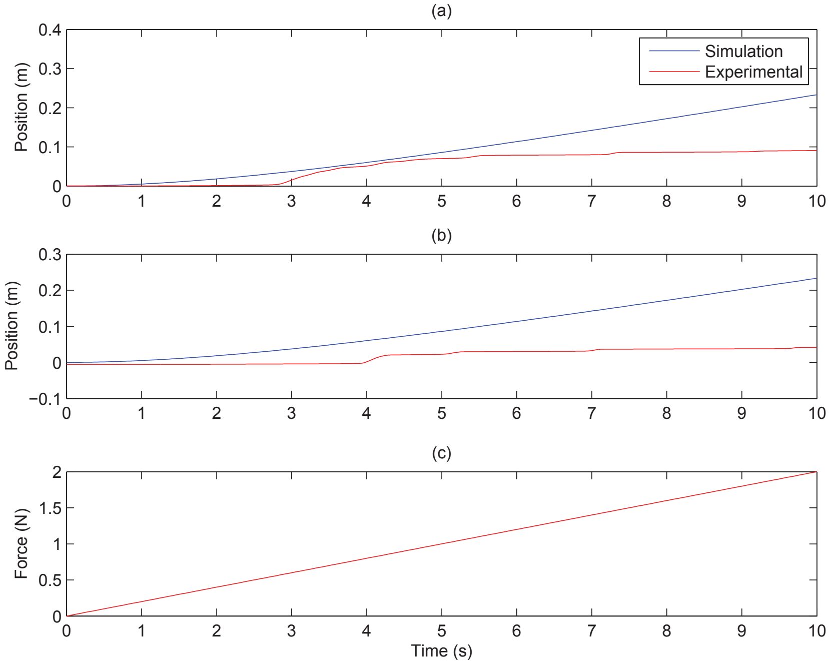


Figure 5.9: Positive ramp input for Phantom Omni device (a) From base, (b) From midpoint (c) Applied force

5.3 Data Acquisition

The derived control algorithm not only requires several pieces of constant information, such as system parameters, to develop the control gain, but it also requires a collection of data in real time to execute the desired control strategy on the hardware. The data necessary for the control algorithm are the manipulator states, human user input force, and environmental contact force.

5.3.1 States

State information is collected using the Quarc software add-on for Simulink made by Quanser. Quarc allows the user to build a control model in the Simulink environment. The model is then converted to a C model and runs on a real-time kernel within Windows. This environment allows for simpler coding, and quicker and more controlled changes for users who are more familiar with Matlab and Simulink than

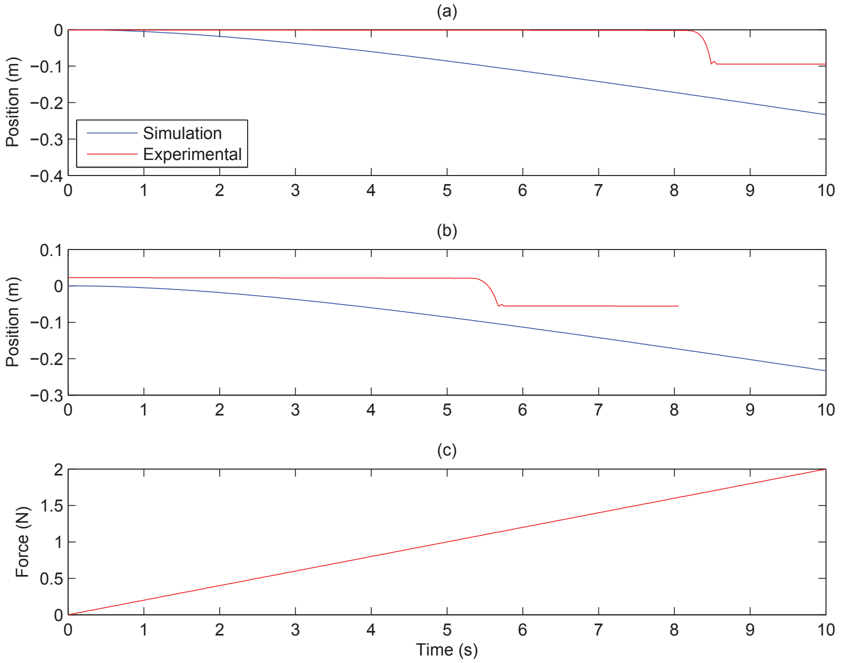


Figure 5.10: Negative ramp input for Phantom Omni device (a) From end, (b) From midpoint, (c) Applied force

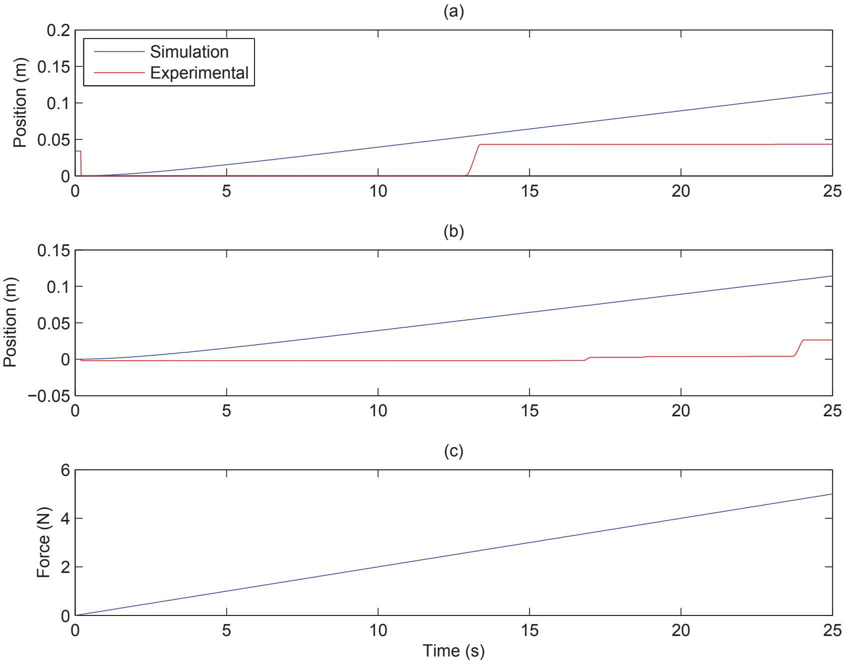


Figure 5.11: Positive ramp input for Novint Falcon device (a) From base, (b) From midpoint, (c) Applied force

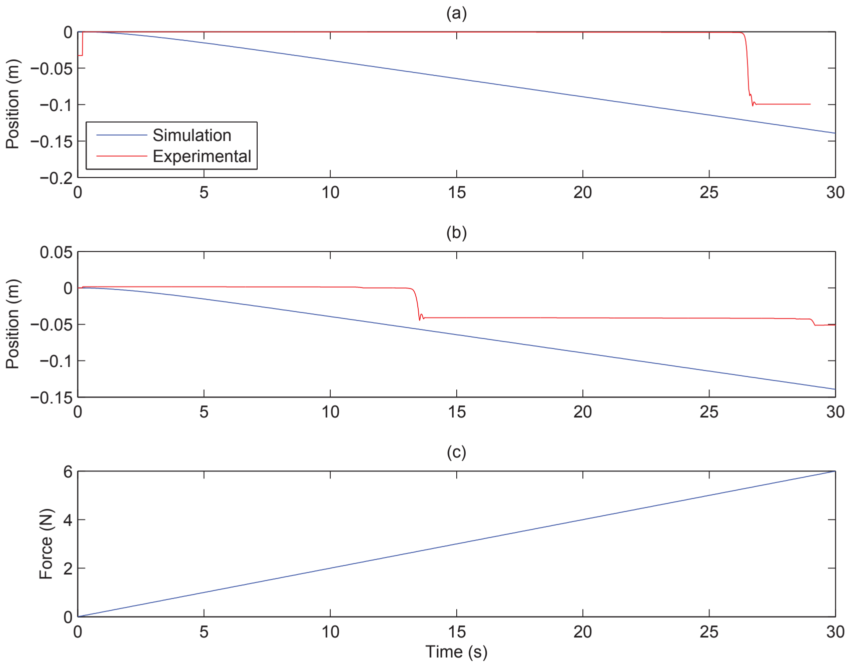


Figure 5.12: Negative ramp input for Novint Falcon device (a) From end, (b) From midpoint, (c) Applied force

with C based programming. Quarc has Simulink specific blocks for both the Phantom Omni and Novint Falcon devices. The Phantom Omni block allows the user to work in either joint spaces (by measuring angles from encoders and applying torques to motors) or in the cartesian workspace with positional output in millimeters and input of force in newtons. With only positional data available, it is then necessary to compute velocity data using a combination of a derivative block and a filter in order to prevent the discontinuities that occur with just a derivative block. For the Novint Falcon, the only output option is cartesian coordinates (in meters), and the input is in force in the cartesian plane (in newtons). Again it is necessary to compute the velocity profile from the position signal and the time step data with a filter to prevent discontinuities.

5.3.2 Forces

The measurement of forces require additional sensors as neither the Phantom Omni or Novint Falcon come equipped with the ability to measure applied loads. Instead of affixing a sensor to measure the input on the master hardware, an artificial input force was generated within Simulink similar to what was done for the simulations. For the environmental contact force, a Futek 10 lb rated load cell was affixed to the Novint Falcon as shown in Fig. 5.13. The load cell was then connected to a Quanser Q8 DAQ board, from which data can be pulled into Simulink through the Quarc software package. Unfortunately, the data was quite noisy (Fig. 5.14(a)) and required filtering to smooth out the data while not impeding the performance of the system. An example of filtered data is shown in Fig. 5.14(b).

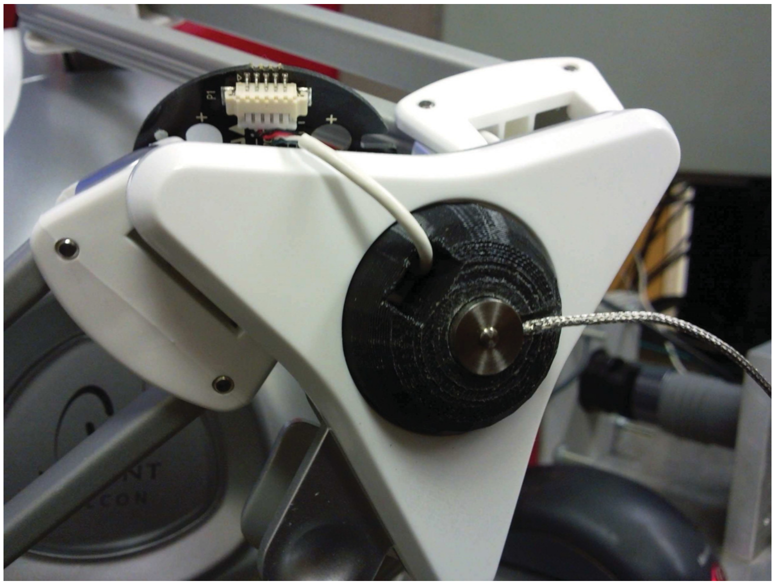


Figure 5.13: Rapid prototype attachment for load cell

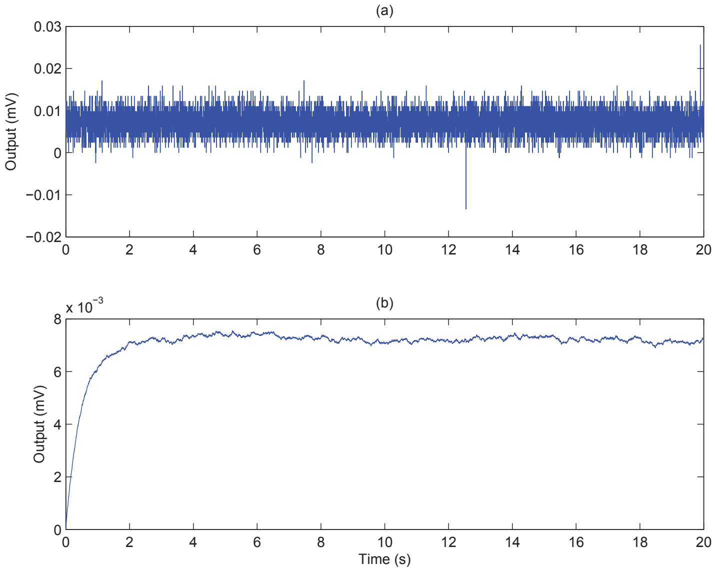


Figure 5.14: Initial load cell data (a) Unfiltered, (b) Filtered

Due to the higher than expected level of noise and odd starting behavior of the filtered signal, a second test was performed with no load applied to see the characteristics of DAQ system. The results in Fig. 5.15 indicated that the DAQ board was indeed influencing the noise and strange behavior of the load cell.

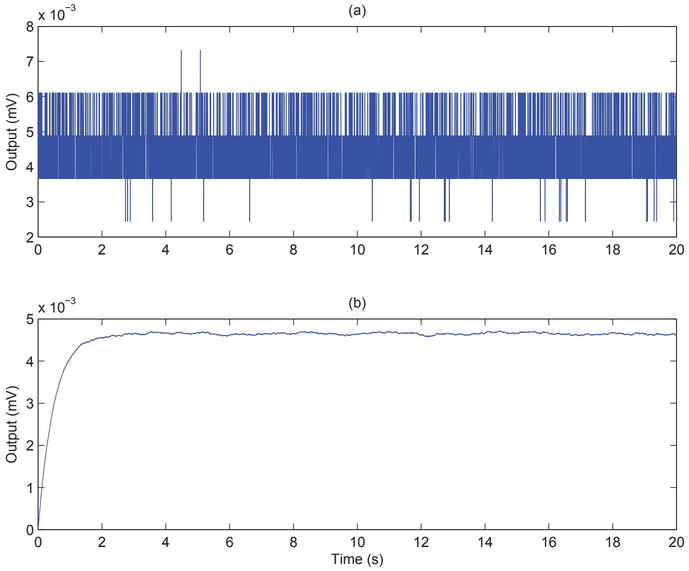


Figure 5.15: Analog input with no connections (a) Unfiltered, (b) Filtered

After considering the results the specifications for the DAQ board were re-examined and found to have a set input range of ±10 V. Since the DAQ board has a 14 bit A/D converter the resulting resolution was 0.122 mV. For the Futek sensor with a 5 V supply this would represent a change in load of 5 N, which is far too coarse resolution. To overcome the resolution issue, an amplification circuit was built to amplify the load cell signal by a factor of 1141.7. Using the circuit yielded a resolution of 0.0045N, which is a far finer resolution for preventing errors arising from A/D conversion. A sample output with a 2kg test load can be seen in Fig. 5.16, which also suggests filtering may not be necessary because of the minimal noise and the slower response caused by the filter.

5.4 System Tests

After determining suitable system models for the hardware involved, and tuning the load cell, it was necessary to run various experimental tests to compare to the theoretical results of Chapter 4. All the following tests were done using Case II (upper and lower bound on time delay). A brief summary is listed in Table 5.1. For reference, the positional error was calculated as $x_s - x_m$ while the net forces were calculated as $U_m + F_h$ for the master and $U_s - F_e$ for the slave.

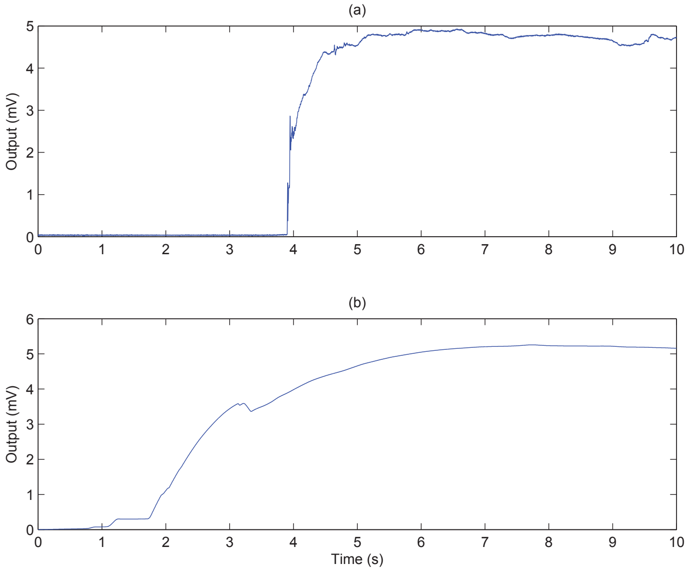


Figure 5.16: Load cell data after amplification (a) Unfiltered, (b) Filtered

Table 5.1: Experimental results summary

Case	Description	Results
1	No Contact Sinusoid	Excellent forward tracking, slight delay on master in slave tracking from non-linearity
2	Direct Foam Contact Sinusoid	Questionable load cell readings, but excellent tracking
3	Plastic Foam Contact Sinusoid	Proper load cell readings, elevated error readings as master non-linearity more noticeable on return motion
4	Plastic Foam Contact Step	Delay in load cell readings causes steady state error as correction force is too small to fix position

5.4.1 No Contact

The system parameters were set as found from the system identification in Section 5.1. Therefore the master controller parameters were set as $M = 0.223$, $B = 17.227$, and $K = 6.286$, which resulted in a control gain set of $K_c = [8.7759, 3.5279]$. A sinusoidal input with a magnitude of 1 N and a frequency of 0.2 Hz was used for the human input F_h. The corresponding system forces are represented in Fig. 5.17, while Fig. 5.18 shows positions (a), velocities (b), and positional error (c). As shown, several peaks occur in the error profile when motion begins in the manipulators.These peaks are due to modeling inaccuracies and various levels of stiction that the manipulators overcome at different rates. Potentially higher control gains may help reduce these peaks, but could add other instability issues to the system.

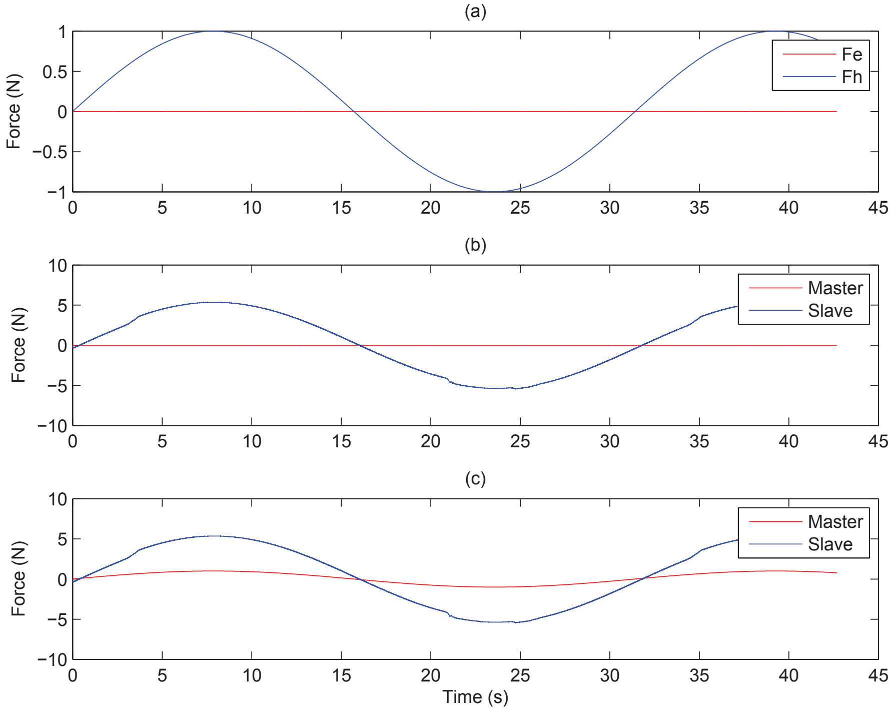


Figure 5.17: Experimental forces without contact (a) External Force, (b) Control Signal, (c) Net force

For direct comparison, the same inputs were used in the simulation model from Chapter 4 with forces in Fig. 5.19 and states in Fig. 5.20.

The results from the comparison of the theoretical linear models to the actual hardware are shown in Figs. 5.17-5.20, and can be seen to vary by a significant margin in terms of magnitudes and smoothness of response. The main contributing factors would be the modeling and linearization of the manipulators. Both manipulators are designed for multiple degrees of freedom, where the degrees are not necessarily independent, and the models are definitely not linear for the degree considered. A second major factor is that the simulation does not account for phenomena such as stiction, since the theoretical manipulators can move with any input force unlike real hardware. While differences are apparent, it is also clear that a level of tracking is achieved on the hardware. Major error spikes happen when one manipulator moves before enough force is applied to the other manipulator to cause motion, due to modeling inaccuracies and non-linearities.

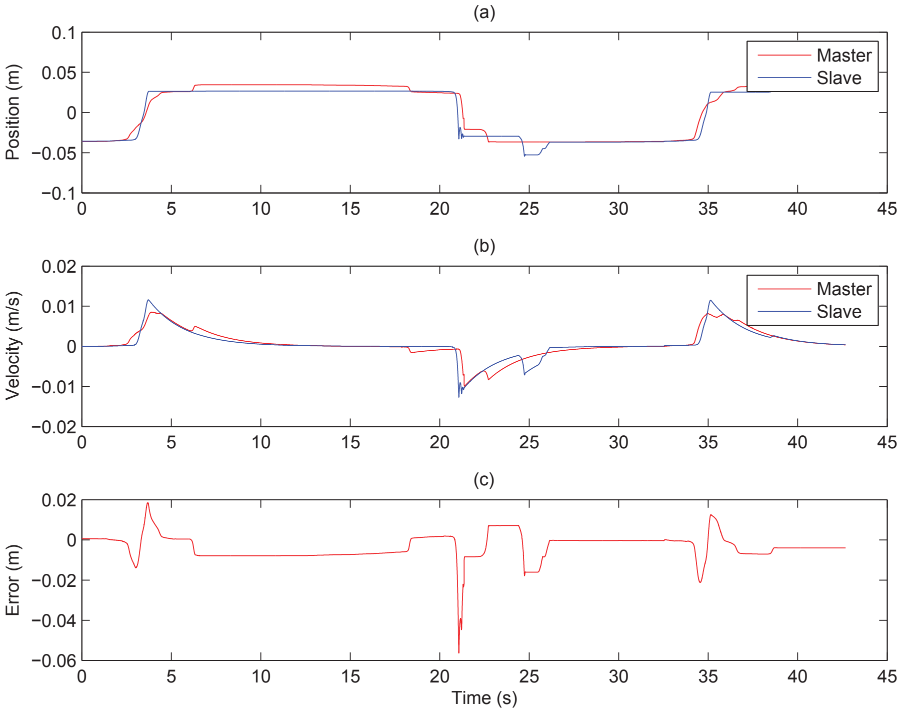


Figure 5.18: Experimental states without contact (a) Position (b) Velocity, (c) Positional error $x_s - x_m$

5.4.2 Contact

For the experimental situations with contact, three different cases were considered: a sinusoid input with direct foam contact, a sinusoid input with a hard surface attached to the foam, and the modified step input into the foam with the hard surface attached to the foam. The hard surface was added to the foam to ensure that the load cell would read the correct level of force, as without it the foam would compress around the other components and affect the load cell readings. In all of the following cases the impedance controller values on the master side were

$$\begin{aligned} M &= 0.223\text{kg} \\ B &= 17.227\text{Ns/m} \\ K &= 6.286\text{N/m} \end{aligned}$$

which led to a control gain of K_c =[8.7759, 3.5279].

Direct Foam Contact with Sinusoid Input

The first case considered was with a sinusoid input and direct contact with the foam surface (no hard wall between the slave and foam). The setup allowed for a softer

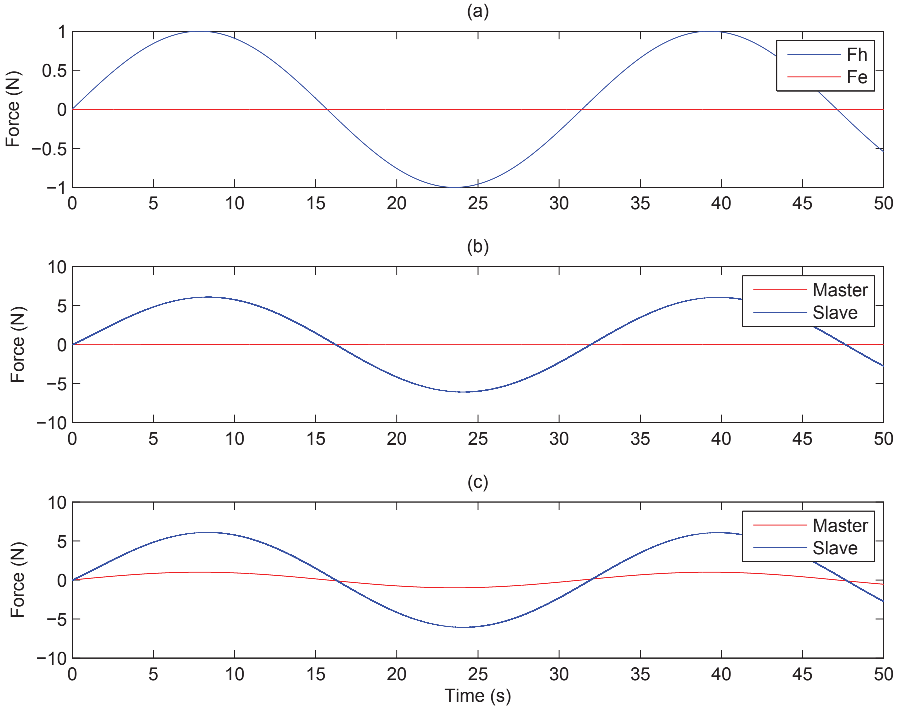


Figure 5.19: Simulation forces without contact (a) External Force, (b) Control Signal, (c) Net Force

wall as it was not necessary to compress more foam than the contact point, however an issue arose with the load cell reading. As shown in Fig. 5.22 (a), the external force that was read during contact shows a slight initial jump and then smoothes off, which varies from the sinusoidal tracking as predicted in Chapter 4. The force measurement behavior is is caused by parts of the manipulator taking part of the load when the foam compresses around the load cell. In such an instance the sensor reads only a fraction of the net force, as shown in Fig. 5.21. While a good level of tracking is achieved as displayed in Fig. 5.23 (c), the validity of these results are to be questioned due to the misleading readings from the load cell.

Plastic Covered Foam Contact with Sinusoid Input

To overcome the issue of components other than the load cell taking load, a clear piece of hard plastic was affixed to the foam as to provide a flat surface for the load cell to rest against (Fig. 5.24). The hard plastic surface allowed the load cell to take the entirety of the contact force, a factor required for accurate readings. Also, by adding the plastic, the total surface area of compression was increased, which caused an increase in the effective stiffness of the wall. The experimental results for the hard

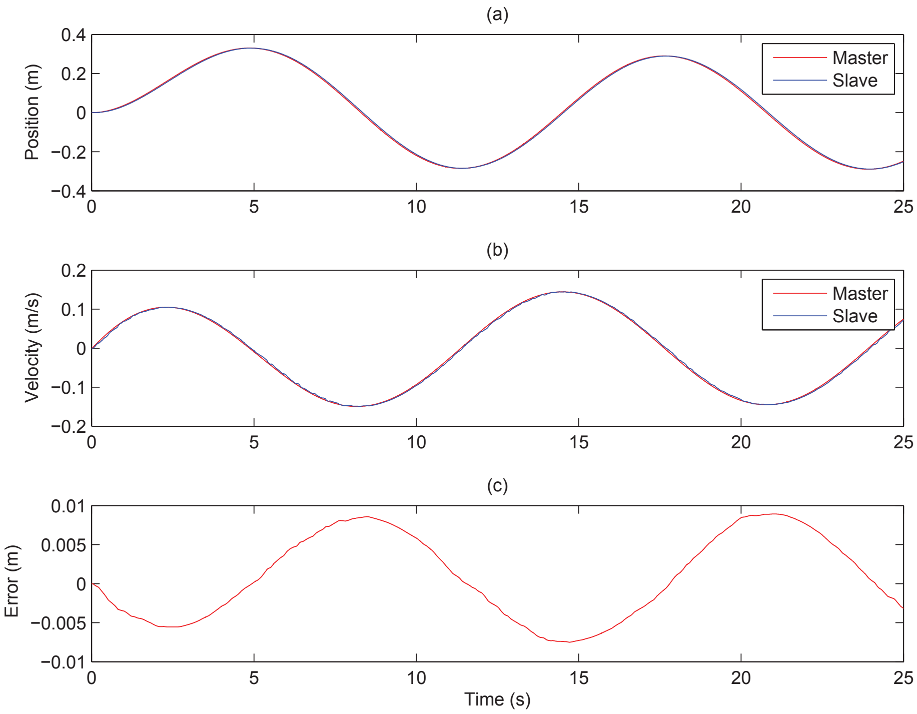


Figure 5.20: Simulation states without contact (a) Position (b) Velocity, (c) Positional Error

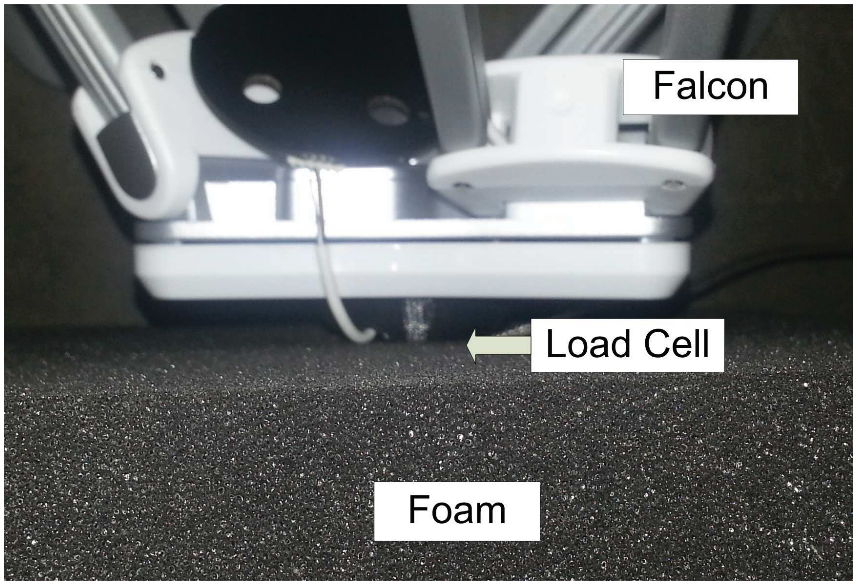


Figure 5.21: Direct foam contact

plastic with a sinusoid input are displayed in Figs. 5.25 and 5.26. As shown in Fig. 5.25 (a), the contact force now shows the sinusoidal shape that was missing in Fig. 5.22 (a). The plot of system error is displayed in Fig. 5.26 (c), which shows elevated

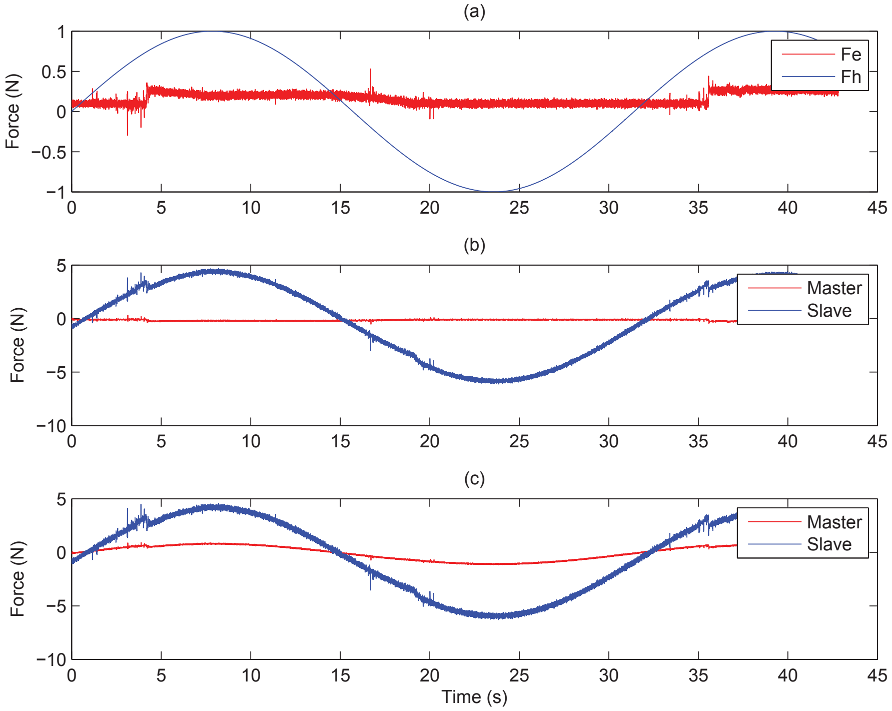


Figure 5.22: Experimental forces with direct foam contact under sinusoid input (a) External Force, (b) Control Signal, (c) Net Force

error compared to the direct foam contact case. In particular, the first cycle shows elevated steady state error, which can be in part explained by the lag in the contact force (Fig. 5.25 (a)), along with modeling inaccuracies and sensor reading variations. Similar to the no contact case, spikes in error exist when one manipulator moves before the other due to modeling and friction issues. Another oddity is when the manipulators move back to the base, the slave manipulator leads the master. The behavior can be traced back to the non-linearities in the manipulators and modeling inaccuracies.

Comparison simulations were conducted using the exact same parameters as set out in the experiment. In order to fully model the contact force F_e, an experimental approximation of the foam stiffness was required. Testing was done using the attached load cell to read the force. A difference in position from when contact occurred to the final position was used to determine the amount of compression. Based on the linear trends shown in Fig. 5.27, the linear relationship of $F = kx$ was used to approximate the foam stiffness at 4000 N/m.

The resulting simulation results are shown in Figs. 5.28 and 5.29. The most noticeable issue occurs during the second wall contact phase and the oscillatory motion as shown

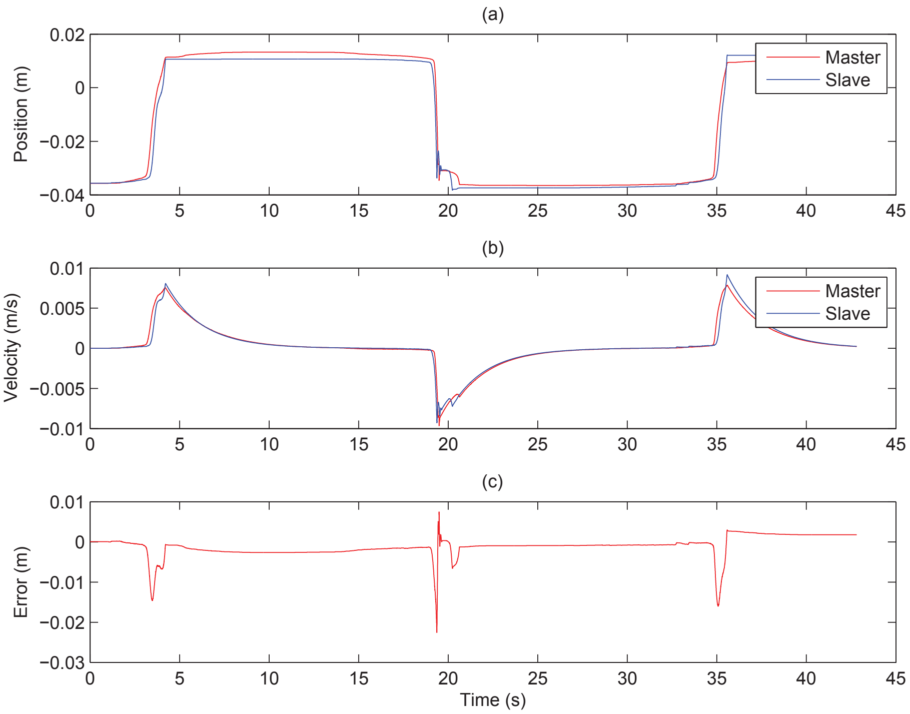


Figure 5.23: Experimental states with direct foam contact under sinusoid input (a) Position (b) Velocity, (c) Positional Error

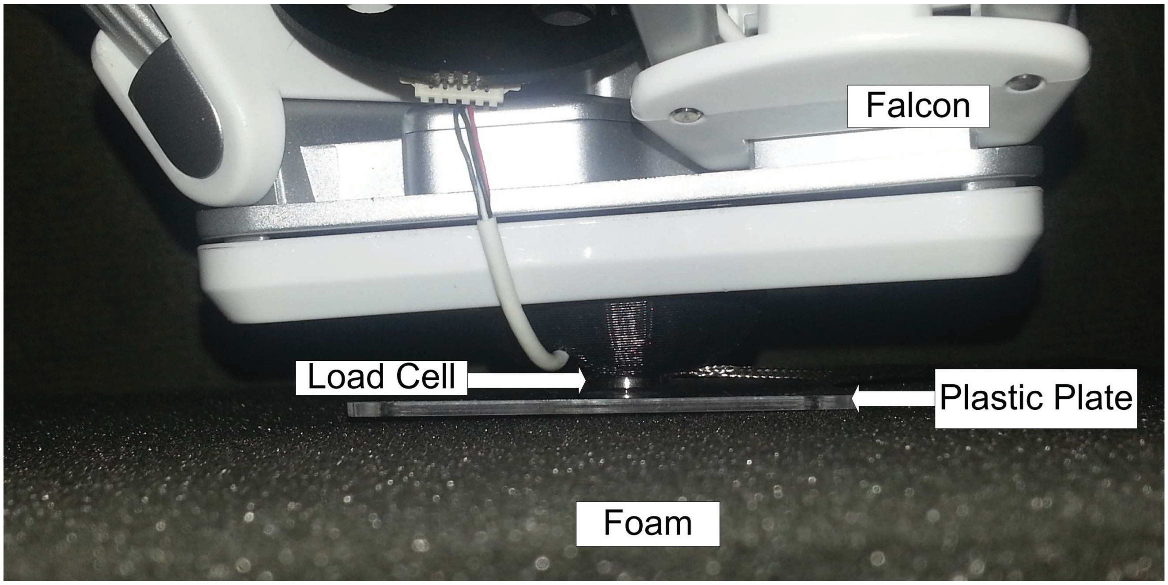


Figure 5.24: Contact with plastic affixed to the foam

in Fig. 5.28 (a). The other major discrepancy is the smooth nature of the motion in the simulation compared to the jerky motion of the actual hardware. The difference can be attributed to trying to use linear models for the equipment, to inaccuracies in

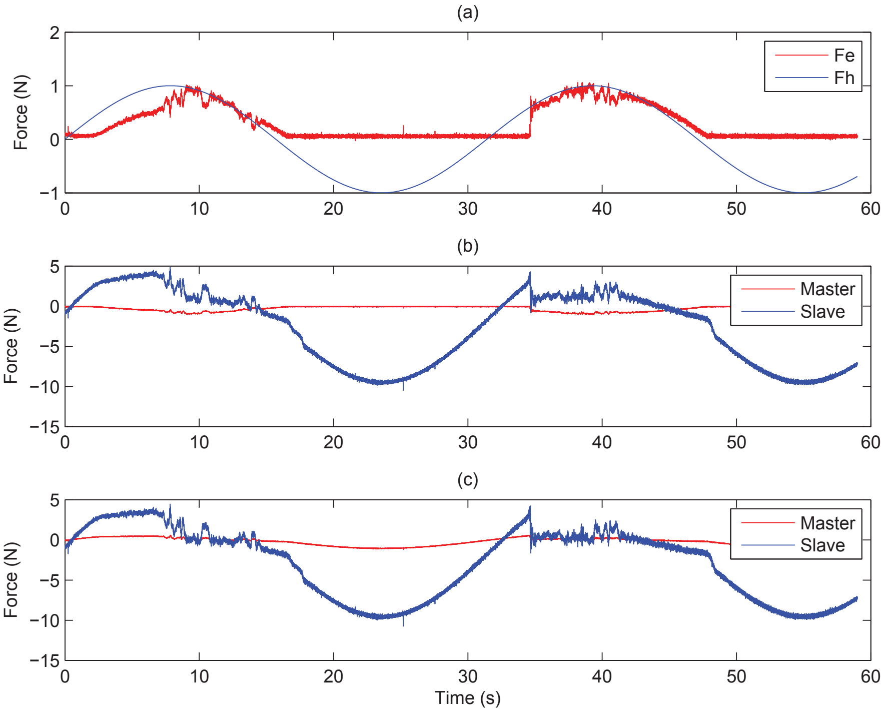


Figure 5.25: Experimental forces with contact under sinusoid input (a) External Force, (b) Control Signal, (c) Net Force

the model parameters, and to system friction.

Foam Contact with Modified Step Input

The final case considered was with the modified step input to evaluate the steady state performance. This case is a departure from the sinusoidal input which was used to evaluate constantly changing states and forces. Figs. 5.30 and 5.31 show the results for when a modified step input (ramp up to 1.5 N in 9 s) is applied to the system. Similar to the ramp test results, a minimum force was needed to move the manipulators as demonstrated by the lack of position change and constant environmental force up to 4 s. As shown in Fig. 5.31 (a), the master manipulator travels much further than the slave manipulator due to the delay in motion of the slave, and to the fact that the force read by the load cell is less than the input ramp. In the ideal case of Fig. 5.32 (a), the contact force should mimic that of the input when there is no change between the master and its impedance controller model. The variation between the simulation and actual results can be attributed to the use of linear models for the non-linear hardware, inaccurate parameters from the system identification, and the load not acting directly perpendicular to the load cell surface.

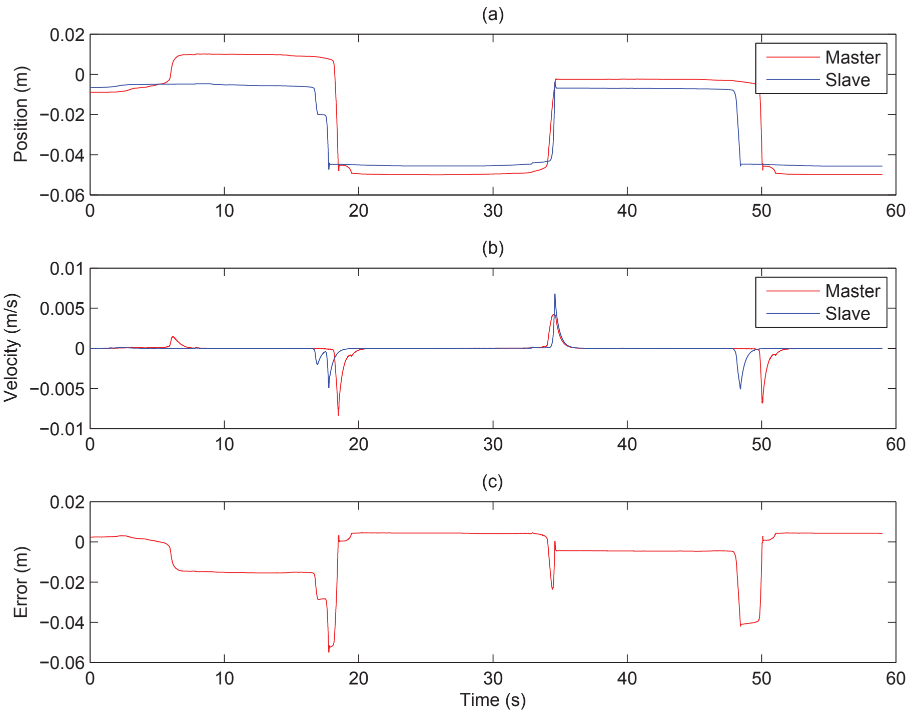


Figure 5.26: Experimental states with contact under sinusoid input (a) Position (b) Velocity, (c) Positional Error

The same set of parameters was used to run a simulation to compare the experimental results to the expected results. See Figs. 5.32 and 5.33. The major point of concern was the 'chattering' behavior of the slave control signal and net force during the ramp as shown in Figs. 5.32 (b) and (c). The 'chattering' velocity, as shown in Fig. 5.33 (b), indicated that the manipulators were vibrating at a high frequency with low displacement. The behavior was caused by the high stiffness of the contact wall. Fig. 5.33 (c) shows the positional error going to zero as the theory indicates. This behavior was not seen in the actual experiment as the correcting force is too small to overcome frictional effects, along with other issues previously discussed.

5.5 Summary

In summary, linear models for both the master hardware (Phantom Omni) and slave hardware (Novint Falcon) were found using the ERA/OKiD identification method using white noise as the force input. A Futek load cell was used to measure contact forces on the slave side. An amplification circuit was required to produce accurate results due to the input range of the DAQ card. Experiments were conducted for

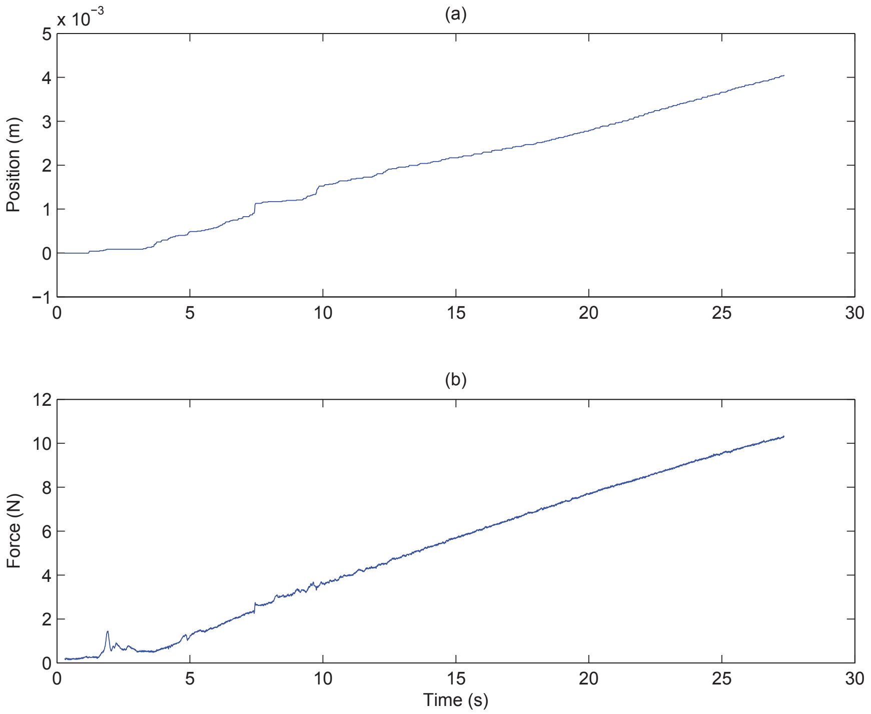


Figure 5.27: Wall stiffness testing (a) Position (b) Force

four cases: no contact, direct foam contact with a sinusoid input, hard plastic covering contact with a sinusoid input, and hard plastic contact with a modified step input. The experimental results vary from the simulation results due to modeling inaccuracies, non-linearities in the manipulators, and difficulties in ensuring that all the contact force is properly transmitted through the load cell.

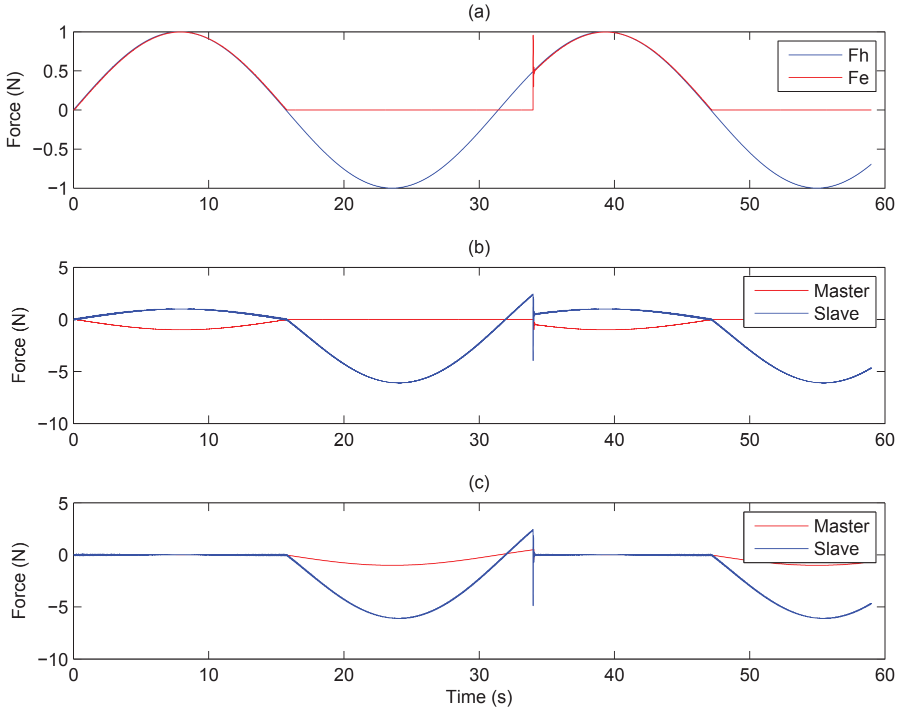


Figure 5.28: Simulation forces with contact under sinusoid input (a) External Force, (b) Control Signal, (c) Net Force

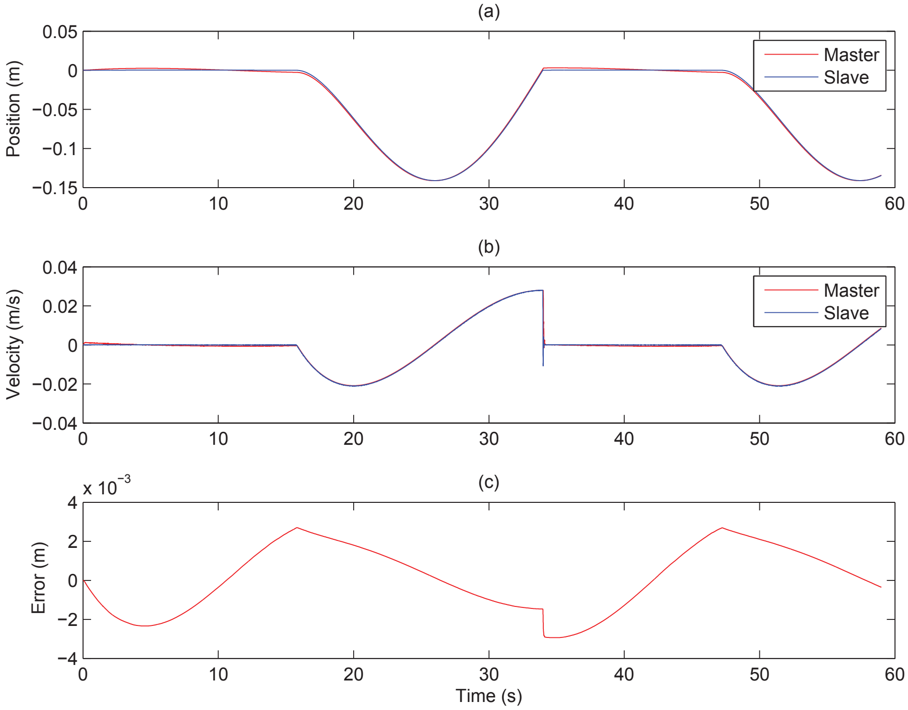


Figure 5.29: Simulation states with contact under sinusoid input (a) Position (b) Velocity, (c) Positional Error

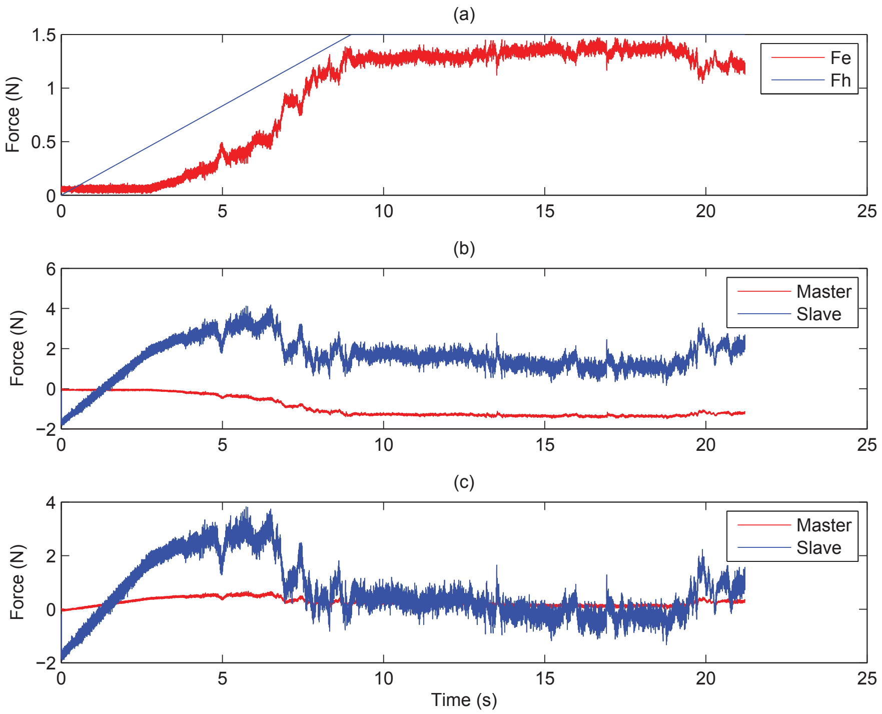


Figure 5.30: Experimental forces with contact under modified step input (a) External Force, (b) Control Signal, (c) Net Force

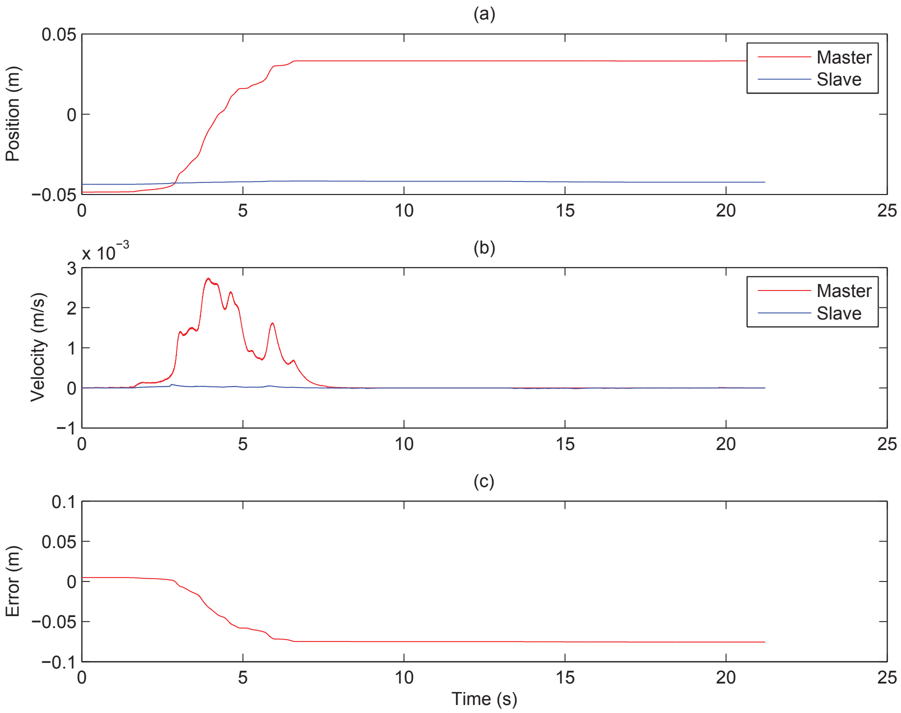


Figure 5.31: Experimental states with contact under modified step input (a) Position (b) Velocity, (c) Positional Error

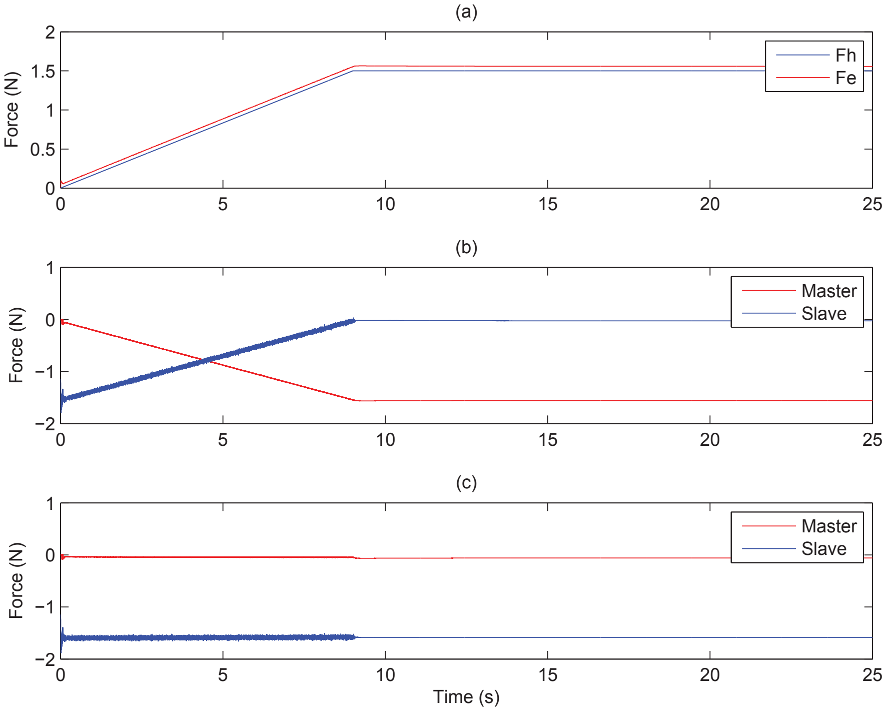


Figure 5.32: Simulation forces with contact under modified step input (a) External Force, (b) Control Signal, (c) Net Force

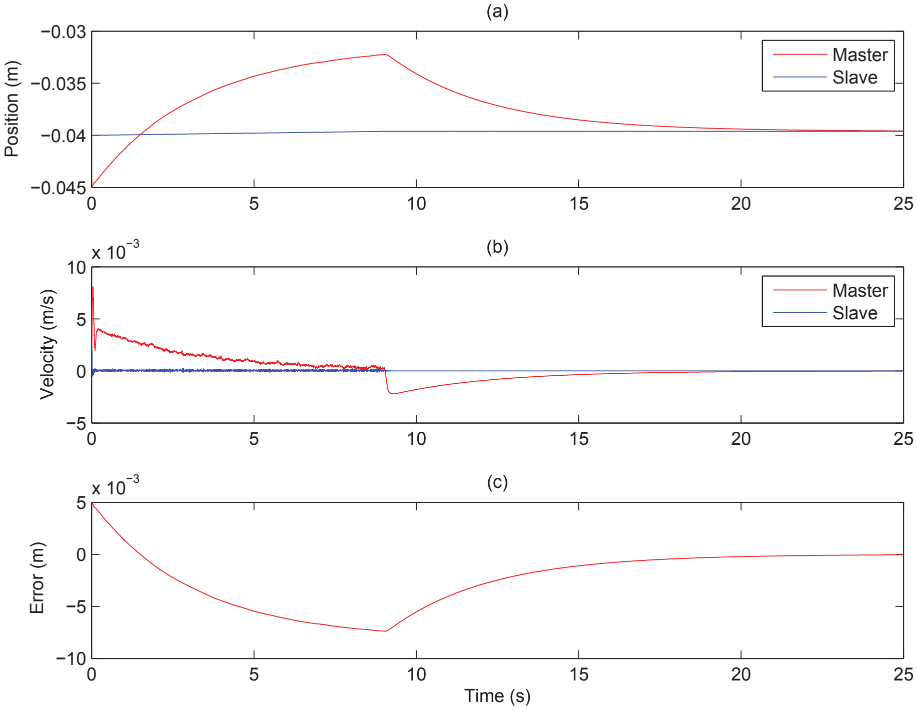


Figure 5.33: Simulation states with contact under modified step input (a) Position (b) Velocity, (c) Positional Error

Chapter 6

Conclusions and Future Work

This chapter summarizes the results of this thesis and suggests developments to be pursued in the future.

6.1 Conclusions

In conclusion, a novel stabilization scheme for asymmetric bilateral teleoperation systems was developed and tested. The work formulates the bilateral teleoperation problem into a NCS problem with external disturbances. The simulation work shows that the algorithm provides stable results by ensuring master stability with an impedance matching controller, and by proving there is a bound on the error between the master and slave states. The bound is based on the rate of change of the disturbances (human input and environmental contact), and the LMI variable results γ_1, γ_2, R_1, R_2, and S. Simulation cases with a constantly varying input (sinusoid), and a constant input after a fixed time (modified step), show that the error is bounded in both cases, and that when there are no disturbances, the error tends to zero.

Experimental tests were also conducted using the algorithm to verify performance on a real world system. Matlab and Simulink were used with the Quarc package to create the test environment for the Phantom Omni (master) and Novint Falcon (slave). A Futek load cell was attached to the slave hardware to measure the environmental force, while Simulink was used to generate the human input. The results showed that for the sinusoid cases a level of tracking was achieved, but with larger errors due to the modeling inaccuracies and linearization of non-linear hardware. The modified step results varied from the simulation results primarily due to the effects of friction. The correcting control force was not large enough to move the manipulator in practice, but would be large enough in a perfectly linear theory model.

6.2 Future Work

There are a couple directions one could go for future work based on the results of this thesis. The first is to do the experiments on more linear hardware to better verify the control algorithm results. An extension of this would be to also consider other methods for system identification, as identifying a higher order model and reducing the results may produce a more accurate model than methods proposed in this thesis. It should also be noted that other methods for system identification exist and may be further explored in an effort to better determine the system properties. The second would be to try and improve the modeling from linear single degree of freedom manipulators to higher degrees of freedom, and explore how non-linear effects or parametric uncertainty can be incorporated into the algorithm. A third area of future work would be modeling of the network behavior. Currently only time varying delays with an upper and lower bound are considered. Advanced modeling of phenomenon like packet loss, and packet ordering should be considered as these effects occur on lower quality networks such as the internet.

Bibliography

[1] Y. Pan, C. Canudas-de-Wit, O. Sename, "A New Predictive Approach for Bilateral Teleoperation With Applications to Drive-by-Wire Systems," Robotics, IEEE Transactions on , vol. 22, no. 6, pp.1146-1162, December 2006.

[2] P.G. de Barros, R.W. Lindeman, M.O. Ward, "Enhancing robot teleoperator situation awareness and performance using vibro-tactile and graphical feedback," 3D User Interfaces, IEEE Symposium on, 2011, pp. 47-54.

[3] J. R. Ware. "Power Based Time Domain Passivity Control and its Application to Bilateral Teleoperated Robotic Vehicles." M.A.Sc. thesis, Dalhousie University, Canada, 2010.

[4] J. Y. Lew and D. Repperger, "Wave Variables Based Teleoperation with Time Delay: Application to Space Based Laser Maintenance," IEEE Aerospace Conference Proceedings, 2004, pp. 2912-2919.

[5] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," Robotics and Autonomous Systems, vol. 38, no. 1, pp. 49-64, January 2002.

[6] T. Slama, N. De Rossi, A. Trevisani, D. Aubry, R. Oboe, "Stability experiments of a scaled bilateral teleoperation system over Internet using a model predictive controller," Industrial Electronics, IEEE International Symposium on, 2007, pp. 3150-3156.

[7] M. Alise, R. Roberts, D. Repperger, "Simulating a Bilateral Teleoperation System Using Matlab and SIMULINK," Air Force Research Lab, AFRL-HE-WP-JA-2007-0008, August 2007.

[8] H. Takahashi, T. Yonemura, N. Sugita, M. Mitsuishi, S. Sora, A. Morita, R. Mochizuki, "Master manipulator with higher operability designed for micro neuro surgical system," Robotics and Automation, IEEE International Conference on, 2008, pp. 3902-3907.

[9] J. Artigas; J.H. Ryu; C. Preusche, "Position drift compensation in time domain passivity based teleoperation," Intelligent Robots and Systems (IROS),IEEE/RSJ International Conference on, 2010, pp.4250-4256.

[10] N. Chopra, M. W. Spong, R. Lozano, "Synchronization of bilateral teleoperators with time delay," Automatica, vol. 44, no. 8, pp.2142-2148, August 2008.

[11] J. H. Ryu and J. H. Kim, "Stable and high performance teleoperation with time domain passivity control: reference energy following scheme," Advanced Robotics, 12th International Conference on , 2005, pp. 782-787.

[12] N. Chopra, M.W. Spong, R. Ortega, N.E. Barabanov, "On tracking performance in bilateral teleoperation," Robotics, IEEE Transactions on , vol. 22, no. 4, pp.861-866, August 2006.

[13] K. Ogata, "Modern Control Engineering," 4th ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[14] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," Automatica, vol.42, no.12, pp.2035-2057, December 2006.

[15] K. H. Ang, G. Chong, L. Yun, "PID control system analysis, design, and technology," Control Systems Technology, IEEE Transactions on , vol. 13, no. 4, pp.559-576, July 2005.

[16] E. Nuno, R. Ortega, N. Barabanov, L. Basanez, "A Globally Stable PD Controller for Bilateral Teleoperators," Robotics, IEEE Transactions on , vol. 24, no. 3, pp.753-758, June 2008.

[17] A. Forouzantabar, H.A. Talebi, A.K. Sedigh, "Bilateral control of master-slave manipulators with constant time delay," American Control Conference (ACC), 2011, pp. 1133-1138.

[18] L.G. Garcia-Valdovinos, V. Parra-Vega, J.-A. Mendez-Iglesias, M.A. Arteaga, "Cartesian sliding PID force/position control for transparent bilateral teleoperation," 31st Annual Conference of IEEE Industrial Electronics Society, 2005, pp. 1979-1985.

[19] H. Khalil; "Nonlinear Systems," 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[20] S. Khan, A. Sabanovic, A.O. Nergiz, "Scaled Bilateral Teleoperation Using Discrete-Time Sliding-Mode Controller," Industrial Electronics, IEEE Transactions on , vol. 56, no. 9, pp.3609-3618, September 2009.

[21] H. C. Cho, J. H. Park, K. Kim, J.-O. Park, "Sliding-mode-based impedance controller for bilateral teleoperation under varying time-delay," Robotics and Automation, IEEE International Conference on , 2001, pp. 1025-1030.

[22] A. Hace, M. Franc, K. Jezernik, "Sliding mode control for scaled bilateral teleoperation," 37th Annual Conference on IEEE Industrial Electronics Society , 2011, pp. 3430-3435.

[23] H. Gao, W. Sun, P. Shi, "Robust Sampled-Data H_∞ Control for Vehicle Active Suspension Systems," Control Systems Technology, IEEE Transactions on , vol. 18, no. 1, pp.238-245, Jan. 2010.

[24] J. Li, Q. Zhang, Y. Xie, "Robust H∞ control of uncertain networked control systems with dropout compensation and Markov jumping parameters," Intelligent Control and Automation, 7th World Congress on , 2008, pp. 7970-7975.

[25] P. Sun, S. Li, “Robust H-Infinity Filtering on Stochastic Uncertain Systems,” Wireless Communications, Networking and Mobile Computing, 4th International Conference on ,2008, pp. 1-4.

[26] Y. Cai, X. He, X. Xu, W. Zhang, “Robust H∞ filter design for uncertain linear systems with state delay,” Intelligent Control and Automation, Proceedings of the 4th World Congress on , 2002, pp. 1726-1730.

[27] K. B. Kim, “Design of feedback controls supporting TCP based on the state-space approach,” Automatic Control, IEEE Transactions on , vol.51, no.7, July 2006, pp. 1086-1099.

[28] J. Qiu, G. Feng, H. Gao, “Asynchronous Output-Feedback Control of Networked Nonlinear Systems With Multiple Packet Dropouts: TS Fuzzy Affine Model-Based Approach,” Fuzzy Systems, IEEE Transactions on , vol.19, no.6, December 2011, pp. 1014-1030.

[29] J. Lin and R.-J. Lian, “Intelligent Control of Active Suspension Systems,” Industrial Electronics, IEEE Transactions on , vol.58, no.2, February 2011, pp. 618-628.

[30] G. Goebel, U. Munz, F. Allgower, “Stabilization of linear systems with distributed input delay,” American Control Conference (ACC), 2010, pp. 5800-5805.

[31] Q. Wang, D.R. Broome, I.T.W. Daycock, “Adaptive force control of the PUMA 560 industrial robot manipulator ,” Control International Conference on , 1991, pp. 358-361.

[32] M. Nohmi, M. Ando, T. Bock, “Contact task by space teleoperation using force reflection of communication time delay,” Computational Intelligence in Robotics and Automation, IEEE International Symposium on ,2005, pp. 193-198.

[33] R. Anderson and M.W. Spong, “Bilateral control of teleoperators with time delay,” IEEE Transactions on Automatic Control, vol. 34, no. 5, pp.494-501, May 1989.

[34] L.-Y. Chen, H. Fujimoto, K. Miwa, T. Abe, A. Sumi, Y. Ito, “A dental training system using virtual reality,” Computational Intelligence in Robotics and Automation, IEEE International Symposium on , 2003, pp. 430-434.

[35] W.J. Book, L.J. Love, M. Farah, “A Teleoperation Testbed for Nuclear Waste Restoration,” Spectrum94: International Topical Meeting on Nuclear and Hazardous Waste Management, 1994, pp. 1017-1021.

[36] Intuitive Surgical Inc. Intuitive surgical - products.[Online] Available: http://www.intuitivesurgical.com/products/index.aspx [Accessed: 15 June 2011]

[37] K. Schilling, H. Roth, R. Lieb, “Teleoperations of rovers. From Mars to education,” Industrial Electronics, Proceedings of the IEEE International Symposium on , 1997, pp. SS257-SS262.

[38] Y.-J. Pan and J. Gu, “Stabilization of a class of remote control systems and its robust stability analysis,” Mechatronics and Automation, IEEE International Conference , 2005, pp. 584-591.

[39] K. Walker, Y.-J. Pan, J. Gu, “Control gain design for bilateral teleoperation systems using linear matrix inequalities,” Systems, Man and Cybernetics, IEEE International Conference on , 2007, pp. 2588-2593.

[40] L. El Ghaoui, F. Oustry, M.AitRami, “A cone complementarity linearization algorithm for static output-feedback and related problems,” Automatic Control, IEEE Transactions on , vol. 42, no. 8, pp.1171-1176, August 1997.

[41] C. Peng, Y.-C. Tian, M. O. Tad , “State feedback controller design of networked control systems with interval time-varying delay and nonlinearity,” Int. J. Robust Nonlinear Control, vol. 18, no. 12, pp. 12851301, August 2008.

[42] C. Peng, D. Yue, E. Tian, Z. Gu, “A delay distribution based stability analysis and synthesis approach for networked control systems,” Journal of the Franklin Institute, vol. 346, no. 4, pp. 349-365, May 2009.

[43] K. Gu, V. L. Kharitonov, J. Chen,“ Stability of Time-delay Systems,” Birkhauser, Basel, 2003.

[44] T. Namerikawa and H.Kawada, “Symmetric Impedance Matched Teleoperation with Position Tracking,” Decision and Control, 45th IEEE Conference on , 2006, pp. 4496-4501.

[45] B. Tang, “State Feedback Controller Design for Networked Control System with Random Network-Induced Delay and Packet Loss,” Networking, Sensing and Control, IEEE International Conference on 2007, pp. 798-802.

[46] C. Hua and P.X.Liu, “Delay-dependent stability analysis of teleoperation systems with unsymmetric time-varying delays,” Robotics and Automation, IEEE International Conference on , 2009, pp. 1146-1151.

[47] Y. Wang, Z. Sun, W. Chou, “Robust Controller Design for Teleoperation Systems with Time-Varying Delays,” Measuring Technology and Mechatronics Automation (ICMTMA), International Conference on , 2010, pp. 266-269.

[48] M.Azadegan, S. Ozgoli, H.R. Taghirad, “Delay-independent robust stability analysis of teleoperation,” Chinese Control and Decision Conference (CCDC), 2011, pp. 4129-4133.

[49] M. Sha Sadeghi, H.R. Momeni, R. Amirifar, “H_∞ and L1 control of a teleoperation system via LMIs,” Applied Mathematics and Computation, vol. 206, no. 2, pp.669-677, December 2008.

[50] Y.Kang, Z. Li, W. Shang, H. Xi, “Motion synchronisation of bilateral teleoperation systems with mode-dependent time-varying communication delays,” Control Theory and Applications, IET , vol. 4, no. 10, pp.2129-2140, October 2010.

[51] E. Nuo, R. Ortega, L. Basaez, D. Hill, “Synchronization of Networks of Nonidentical Euler-Lagrange Systems With Uncertain Parameters and Communication Delays,” Automatic Control, IEEE Transactions on , vol. 56, no. 4, pp.935-941, April 2011.

[52] E. Nuo, R. Ortega, L. Basaez, “An adaptive controller for nonlinear teleoperators,” Automatica, vol. 46, no. 1, pp.155-159, January 2010.

[53] K. Walker, “Stability and control of bilateral teleoperation over the internet with force feedback,” M.A.Sc. thesis, Dalhousie University, Canada, 2010.

[54] N.Tanner and G. Niemeyer, “High-frequency acceleration feedback in wave variable telerobotics,” IEEE/ASME Transactions on Mechatronics, vol. 11, no. 2, pp.119-127, April 2004.

[55] W. T. Ang, P. K.Pradeep, C. N. Riviere, “Active tremor compensation in microsurgery,” In Proceedings of the 26th Annual International Conference of the IEEE EMBS, 2004, pp. 2738-2741.

[56] D. Feth, A. Peer, M. Buss, “Incorporating human haptic interaction models into teleoperation systems,” Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on , 2010, pp. 4257-4262.

[57] Y. Gao, J. Li, H. Su, J. Li, “Development of a teleoperation system based on virtual environment,” Robotics and Biomimetics (ROBIO), IEEE International Conference on , 2011, pp. 766-771.

[58] E. Nuno, L. Basanez, R. Ortega, M. W. Spong, “Position tracking for non-linear teleoperators with variable time delay,” The International Journal of Robotics Research, vol. 28, no. 7, pp.895910, July 2009.

[59] J.-N. Juang, “Applied System Identification,” Upper Saddle River, N.J.,: Prentice Hall, 1994.

[60] J.S.-H. Tsai, Y.-C. Shiu, S.-M. Guo, L.-S. Shieh, “Fault-tolerant tracker for a class of unknown interconnected nonlinear systems with input constraint,” Systems, Signals and Devices (SSD), 8th International Multi-Conference on , 2011, pp. 1-6.

[61] Sensable Inc. Phantom Omni. [Online] Available: http://www.sensable.com/haptic-phantom-omni.htm, [Accessed: 14 June 2011.]

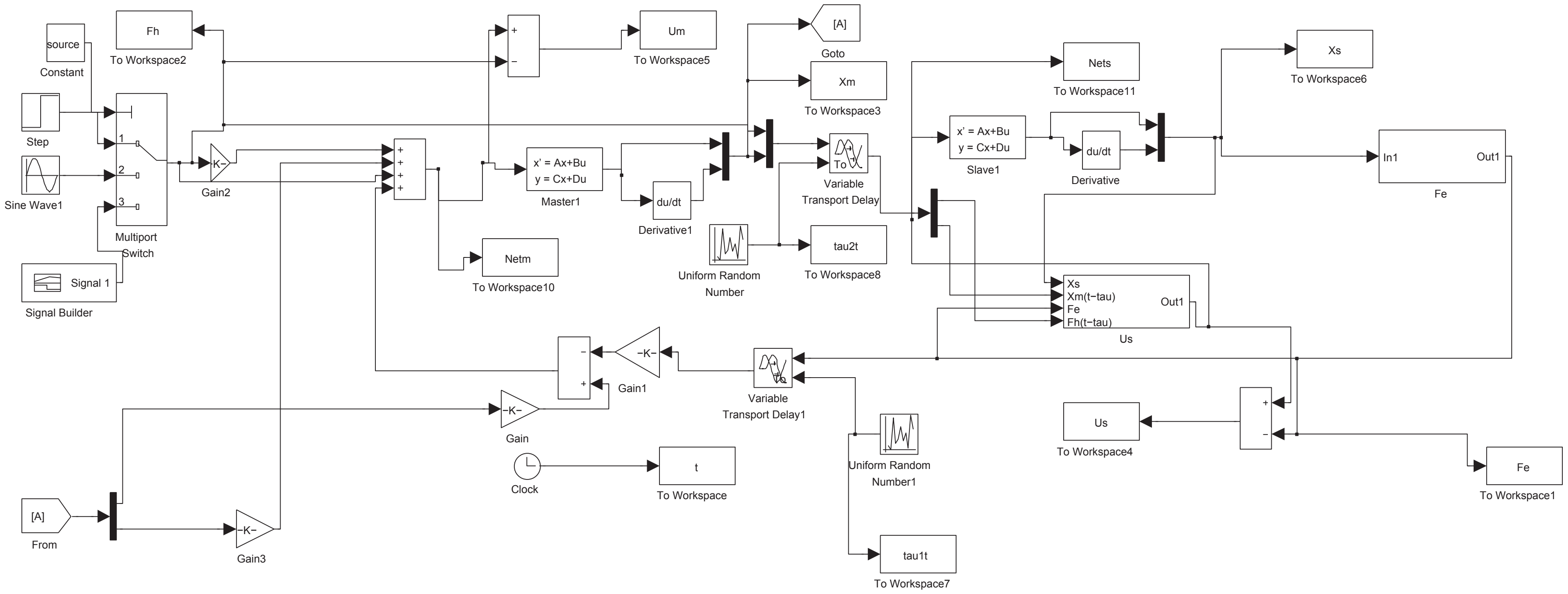
[62] Novint Technologies Inc. Novint Falcon. [Online] Available: http://www.novint.com/index.php/novintfalcon, [Accessed: 14 June 2011.]

[63] Y. Ye, Y. Pan, Y. Gupta, “Time domain passivity control of teleoperation systems with random asymmetric time delays,” IEEE Conference on Decision and Control, 28th Chinese Control Conference. CDC/CCC 2009. 2009, pp. 7533-7538.

[64] C. Melchiorri, “Robotic telemanipulation systems: an overview on control aspects,” in Proc. 7th IFAC Symp. Robot Control, 2003, pp. 707-716.

Appendix A

Matlab Simulink Block Diagram for Simulation


source
Constant
Fh
To Workspace2
Step
Sine Wave1
1
2
3
Multiport
Switch
Signal 1
Signal Builder
K-
Gain2
+
+
+
+
+
-
Um
To Workspace5
x' = Ax+Bu
y = Cx+Du
Master1
du/dt
Derivative1
Netm
To Workspace10
[A]
Goto
Xm
To Workspace3
Variable
Transport Delay
To
Uniform Random
Number
tau2t
To Workspace8
Nets
To Workspace11
x' = Ax+Bu
y = Cx+Du
Slave1
du/dt
Derivative
Xs
To Workspace6
In1
Out1
Fe
Xs
Xm(t-tau)
Fe
Fh(t-tau)
Out1
Us
-
+
-K-
Gain1
Variable
Transport Delay1
Uniform Random
Number1
tau1t
To Workspace7
-K-
Gain
Clock
t
To Workspace
Us
To Workspace4
+
-
Fe
To Workspace1
[A]
From
-K-
Gain3

Appendix B

Sample Matlab Code

Case I

```
clear; clc; close all;
global A_m A_s B_s B_m U_m U_s F_e F_h
%% Master Variables
m_m=3; %Mass in kg
b_m=3;% Damping
k_m=0;%Stiffness, N/m
%% Slave Variables
m_s=2;%Mass in kg
b_s=2;% Damping
k_s=0;%Stiffness, N/m

%% Master Controller Variables
M=1;%Mass in kg
B=4;% Damping
K=4;%Stiffness, N/m

%% System Matrices
A_m=[0 1;
    -K/M -B/M];
Am=[0 1;
    -k_m/m_m -b_m/m_m];
B_m=[0; 1/m_m];
C_m=[1 0;0 1];
A_s=[0 1;
    -k_s/m_s -b_s/m_s];
B_s=[0; 1/m_s];
C_s=[1 0;0 1];
eig(A_m);
I=eye(2);
```

```
i=1;

tau1=.000; %delay in s
tau2=0.08;%delay in s
gamma1=5;
gamma2=.9;
%% Slave Controller Parameters
alpha=m_s/M;
% LMI CODE for Initial Feasible Set
setlmis([]);
%LMI Variables
X=lmivar(1,[2 1]);
[Y,n,sY]=lmivar(2,[1 2]);
sY;
[YT,n,sYT]=lmivar(3,[sY(1);sY(2)]);
sYT;
Q=lmivar(1,[2 1]);
R=lmivar(1,[2 1]);
P=lmivar(1,[2 1]);
J=lmivar(1,[2 1]);
L=lmivar(1,[2 1]);
G=lmivar(1,[2 1]);

%Main LMI
lmiterm([1 1 1 Q],-1,1);
lmiterm([1 1 1 R],-1,1);

lmiterm([1 1 5 R],1,1);

lmiterm([1 2 2 Q],1,1);
lmiterm([1 2 2 X],A_m,1,'s');
lmiterm([1 2 2 R],-1,1);

lmiterm([1 2 3 0],I);
lmiterm([1 2 5 R],1,1)
lmiterm([1 2 5 Y],B_s,1)

lmiterm([1 2 6 X],1,A_m');

lmiterm([1 2 7 X],tau2,A_m');

lmiterm([1 3 3 0],-I*gamma1);

lmiterm([1 3 4 X],1,1)

lmiterm([1 4 4 0],-I);

lmiterm([1 5 5 R],-2,1);

lmiterm([1 5 6 YT],1,B_s');

lmiterm([1 5 7 YT],tau2,B_s');

lmiterm([1 6 6 0],-I*gamma2);

lmiterm([1 7 7 G],-1,1);
%X>0
lmiterm([-2 1 1 X],1,1);
%Q>0
lmiterm([-3 1 1 Q],1,1);
%R>0
lmiterm([-4 1 1 R],1,1);
%G>0
lmiterm([-5 1 1 G],1,1);
%P>0
lmiterm([-6 1 1 P],1,1);
%J>0
lmiterm([-7 1 1 J],1,1);
%L>0
lmiterm([-8 1 1 L],1,1);
%Condition for P=X^-1
```

```
lmiterm([-9 1 1 X],1,1);
lmiterm([-9 1 2 0],I);
lmiterm([-9 2 2 P],1,1);
%Condition for J=G^-1
lmiterm([-10 1 1 G],1,1);
lmiterm([-10 1 2 0],I);
lmiterm([-10 2 2 J],1,1);
%Condition for L=R^-1
lmiterm([-11 1 1 R],1,1);
lmiterm([-11 1 2 0],I);
lmiterm([-11 2 2 L],1,1);
%G<X^TR^-1X
lmiterm([-12 1 1 J],1,1);
lmiterm([-12 1 2 P],1,1);
lmiterm([-12 2 2 L],I,1);

lmisys = getlmis;
[tmin,xfeas]=feasp(lmisys);

X0 = dec2mat(lmisys, xfeas, X);
Y0 = dec2mat(lmisys, xfeas, Y);
R0 = dec2mat(lmisys, xfeas, R);
Q0 = dec2mat(lmisys, xfeas, Q);
L0 = dec2mat(lmisys, xfeas, L);
P0 = dec2mat(lmisys, xfeas, P);
J0 = dec2mat(lmisys, xfeas, J);
G0 = dec2mat(lmisys, xfeas, G);
K_1=Y0*inv(X0);

tmin=1;
while tmin>0.00000001
%Trace Minimization

% LMI CODE
 setlmis([]);
%LMI Variables
X=lmivar(1,[2 1]);
```

```
[Y,n,sY]=lmivar(2,[1 2]);
sY;
[YT,n,sYT]=lmivar(3,[sY(1);sY(2)]);
sYT;
Q=lmivar(1,[2 1]);
R=lmivar(1,[2 1]);
P=lmivar(1,[2 1]);
J=lmivar(1,[2 1]);
L=lmivar(1,[2 1]);
G=lmivar(1,[2 1]);


%Main LMI
lmiterm([1 1 1 Q],-1,1);
lmiterm([1 1 1 R],-1,1);

lmiterm([1 1 5 R],1,1);

lmiterm([1 2 2 Q],1,1);
lmiterm([1 2 2 X],A_m,1,'s');
lmiterm([1 2 2 R],-1,1);

lmiterm([1 2 3 0],I);

lmiterm([1 2 5 R],1,1)
lmiterm([1 2 5 Y],B_s,1)

lmiterm([1 2 6 X],1,A_m');

lmiterm([1 2 7 X],tau2,A_m');

lmiterm([1 3 3 0],-I*gamma1);

lmiterm([1 3 4 X],1,1)

lmiterm([1 4 4 0],-I);
lmiterm([1 5 5 R],-2,1);

lmiterm([1 6 5 Y],B_s,1);

lmiterm([1 7 5 Y],B_s,tau2);

lmiterm([1 6 6 0],-I*gamma2);

lmiterm([1 7 7 G],-1,1);
%X>0
lmiterm([-2 1 1 X],1,1);
%Q>0
lmiterm([-3 1 1 Q],1,1);
%R>0
lmiterm([-4 1 1 R],1,1);
%G>0
lmiterm([-6 1 1 G],1,1);
%Condition for P=X^-1
lmiterm([-7 1 1 X],1,1);
lmiterm([-7 1 2 0],I);
lmiterm([-7 2 2 P],1,1);
%Condition for J=G^-1
lmiterm([-8 1 1 G],1,1);
lmiterm([-8 1 2 0],I);
lmiterm([-8 2 2 J],1,1);
%G<X^TR^-1X
lmiterm([-9 1 1 J],1,1);
lmiterm([-9 1 2 P],1,1);
lmiterm([-9 2 2 L],I,1);
%P>0
lmiterm([-11 1 1 P],1,1);
%J>0
lmiterm([-12 1 1 J],1,1);
%Condition for L=R^-1
lmiterm([-13 1 1 R],1,1);
lmiterm([-13 1 2 0],I);
lmiterm([-13 2 2 L],1,1);
```

```
%L>0
lmiterm([-14 1 1 L],1,1);

 lmisys = getlmis;

n = decnbr(lmisys) ;
c = zeros(n,1);
for j=1:n,
[Xj,Pj,Jj,Gj,Rj,Lj] = defcx(lmisys,j,X,P,J,G,R,L) ;
c(j) = trace(Xj*P0+Pj*X0+Gj*J0+Jj*G0+Rj*L0+Lj*R0);
end
c
options = [1e-5,0,0,0,0] ;
[copt,xopt] = mincx(lmisys,c,options);
%
X = dec2mat(lmisys, xopt, X);
Y = dec2mat(lmisys, xopt, Y);
R = dec2mat(lmisys, xopt, R);
Q = dec2mat(lmisys, xopt, Q);
L = dec2mat(lmisys, xopt, L);
P = dec2mat(lmisys, xopt, P);
J = dec2mat(lmisys, xopt, J);
G = dec2mat(lmisys, xopt, G);
q=gamma2;


%Check if LMI valid
 setlmis([]);

   P=lmivar(1,[2 1]);
   J=lmivar(1,[2 1]);
   L=lmivar(1,[2 1]);
%  gamma1=lmivar(1,[1 1]);
 gamma5=lmivar(1,[1 1]);

%Main LMI
```

```
lmiterm([1 1 1 0],-Q);
lmiterm([1 1 1 0],-R);

lmiterm([1 1 5 0],R);

lmiterm([1 2 2 0],Q);
lmiterm([1 2 2 0],A_m*X+X*A_m’);
lmiterm([1 2 2 0],-R);

lmiterm([1 2 3 0],I);

lmiterm([1 2 5 0],R)
lmiterm([1 2 5 0],B_s*Y)

lmiterm([1 2 6 0],X*A_m’);

lmiterm([1 2 7 0],tau2*X*A_m’);

lmiterm([1 3 3 0],-I*gamma1);

lmiterm([1 3 4 0],X);

lmiterm([1 4 4 0],-I);

lmiterm([1 5 5 0],-2*R);

lmiterm([1 6 5 0],B_s*Y);

lmiterm([1 7 5 0],B_s*Y*tau2);

lmiterm([1 6 6 0],-I*gamma2);

lmiterm([1 7 7 0],-X*R^(-1)*X);

%Condition for P=X^-1
lmiterm([-7 1 1 0],X);
lmiterm([-7 1 2 0],I);
lmiterm([-7 2 2 P],1,1);
%Condition for J=G^-1
lmiterm([-8 1 1 0],G);
lmiterm([-8 1 2 0],I);
lmiterm([-8 2 2 J],1,1);
%G<X^TR^-1X
lmiterm([-9 1 1 J],1,1);
lmiterm([-9 1 2 P],1,1);
lmiterm([-9 2 2 L],I,1);
%P>0
lmiterm([-11 1 1 P],1,1);
%J>0
lmiterm([-12 1 1 J],1,1);
%Condition for L=R^-1
lmiterm([-8 1 1 0],R);
lmiterm([-8 1 2 0],I);
lmiterm([-8 2 2 L],1,1);

lmisys = getlmis;

[tmin,xfeas]=feasp(lmisys);
% Reset variable for looping
 X0 = X;
 Y0 = Y;
 R0 = R;
 Q0 = Q;
 P0 = dec2mat(lmisys, xfeas, P);
 J0 = dec2mat(lmisys, xfeas, J);
 L0 = dec2mat(lmisys, xfeas, L);
 G0 = G;
 gamma10 = gamma1
 gamma20 = gamma2

end

K_2=Y0/X0;
```

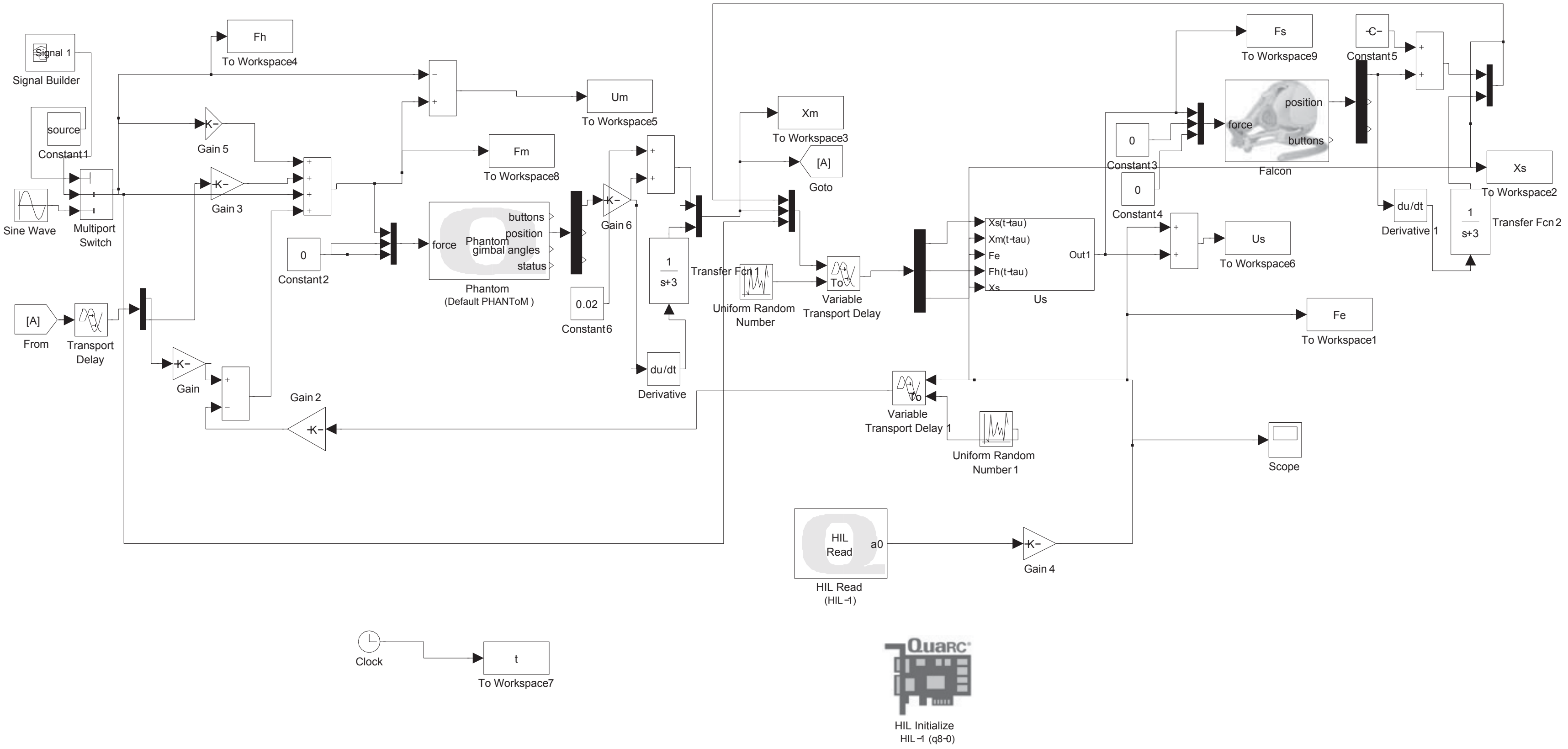
```
K_c=K_2

%% Simulation parameters
tmax=15; %seconds
tstep=0.001; %step size in seconds
t=0:tstep:tmax; %time vector

%Fh=sin(t);%Human input
Be=0; %Slave Environment Damping
Ke=4;%Slave Environment Stiffness
FeG=[Ke Be];
mag=5;
freq=0.5;
delta1=K/M-k_s/m_s;
delta2=B/M-b_s/m_s;
noise=0.1;
%Input Source
source=3;

%% Initial Conditions
x_m(1)=0;
x_s(1)=0;
v_m(1)=0;
v_s(1)=0;
Xm(1,1)=x_m(1);
Xm(2,1)=v_m(1);
Xs(1,1)=x_s(1);
Xs(2,1)=v_s(1);
%% Simulations
sim('finalv1posonly')
e=Xs-Xm;

figure (1)
subplot(3,1,1)
plot(t,Fe,'r',t,Fh,'b')
xlabel('Time (s)')
ylabel('Force (N)')
```

```
legend('Fe','Fh')
title('(a)')
grid on


subplot(3,1,2)
plot(t,Um,'r',t,Us,'b')
xlabel('Time (s)')
ylabel('Force (N)')
legend('Master','Slave')
title('(b)')

subplot(3,1,3)
plot(t,Um+Fh,'r',t,Us-Fe,'b')
xlabel('Time (s)')
ylabel('Force (N)')
legend('Master','Slave')
title('(c)')

figure (2)
subplot(3,1,1)
plot(t,Xm(:,1),'r',t,Xs(:,1),'b')
xlabel('Time (s)')
ylabel('Position (m)')
legend('Master','Slave')
title('(a)')
grid on

subplot(3,1,2)
plot(t,Xm(:,2),'r',t,Xs(:,2),'b')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
legend('Master','Slave')
title('(b)')
grid on

subplot(3,1,3)
```

```
plot(t,e(:,1),'r')
xlabel('Time (s)')
ylabel('Error (m)')
legend('Error')
title('(c)')
grid on

saveas(1,'Images\upperstepforces.eps','psc2')
saveas(2,'Images\upperstepstates.eps','psc2')
```

Appendix C

Matlab Simulink Block Diagram for Experiment


Signal 1
Signal Builder
source
Constant1
Sine Wave
Multiport Switch
Fh
To Workspace4
Gain 5
Gain 3
Um
To Workspace5
Fm
To Workspace8
0
Constant2
force
Phantom
buttons
position
gimbal angles
status
Phantom
(Default PHANToM)
Gain 6
0.02
Constant6
1
s+3
Transfer Fcn1
du/dt
Derivative
Xm
To Workspace3
[A]
Goto
Uniform Random Number
Variable Transport Delay
[A]
From
Transport Delay
Gain
Gain 2
Xs(t-tau)
Xm(t-tau)
Fe
Fh(t-tau)
Xs
Out1
Us
Fs
To Workspace9
-C-
Constant5
0
Constant3
0
Constant4
force
position
buttons
Falcon
Xs
To Workspace2
du/dt
Derivative 1
1
s+3
Transfer Fcn 2
Us
To Workspace6
Fe
To Workspace1
Variable Transport Delay 1
Uniform Random Number 1
Scope
HIL
Read
a0
HIL Read
(HIL-1)
Gain 4
Clock
t
To Workspace7
QUARC
HIL Initialize
HIL-1 (q8-0)

Appendix D

ERA/OKiD Code for System Identification

```
close all
clear all
clc

scrsz = get(0,'ScreenSize') + [10 100 -100 -200];
%% Input parameters
N_rows = input('\nHow many rows would you like in the Hankel Matrix? ')
N_cols = input('\nHow many columns would you like in the Hankel Matrix? ')
nmp_input = input('\nHow many Observer Markov Parameters would you
like to calculate [nmp]? ')
p_input = input('\nWhat would you like p to be? ')

fprintf('\n############### INPUT PARAMETERS ##################\n')
fprintf('The number of rows in the Hankel Martrix is: %3.0f',N_rows)
fprintf('The number of columns in the Hakel Martix is:%3.0f',N_cols)
fprintf('The number of Markov Parameters calculated is:%3.0f',nmp_input)
fprintf('The observer number, p is set to be:       %2.0f',p_input)

%% Test Data
load('falcOKID_2');

%load inpt_outpt
ACC = simout(250:4000)+0.021;
u_rand = force(250:4000);
dt = 0.01;    % Specified sample rate
u_inputR = force(250:4000);
y_outputR =simout(250:4000)+0.021;
%p_noise=.025; %Changes the magnitude of the noise

%t_max = length(ACC) * dt - dt;
```

```
t = time(250:4000);

u_max = length(t);

u_impulse = zeros(u_max,1); % The input matrix
u_impulse(1) = 1;           % The unit impluse

%% Plotting
figure('Name','Original excitation of system','Position',[scrsz])
subplot(2,1,1)
plot(t,u_rand,'r')
title('System Input')
ylabel('Magnitude')
xlabel('Time (s)')

subplot(2,1,2)
plot(t,ACC,'b')
title('Response of the System')
ylabel('Magnitude')
xlabel('Time (s)')

%% Setting up OKID
[l,m,r]=size(ACC);
% l = number of data points
% m = number of outputs
% r = number of inputs

p = p_input;
k = 4;
nmp = nmp_input;

%% Looping for Input V matrix
V = [];                  % Null to begin
for i = 1:p+1
    if i==1
```

```
        V_row = [];
    else
        V_row = zeros(r+m,i-1);
    end

    for j = 1:l-i+1
        if i==1
            V_row = [V_row, u_rand(j,:)’];
        else
            V_row = [V_row, [u_rand(j,:)’; ACC(j,:)’] ];
        end
    end
    V = [V; V_row];
end


%% Computing Observer Markov Parameters
obsMP = ACC’*pinv(V);     % Eqn 6.15

%% Moving around Observer Markov Parameters (Y_bar)

Y_bar_1 = obsMP(:,1);

for i = 2:2:length(obsMP)
    temp = obsMP(i);
    Y_bar_1 = cat(2,Y_bar_1, temp);
end

Y_bar_2 = 0;

for i = 3:2:length(obsMP)
    temp = -obsMP(i);
    Y_bar_2 = cat(2,Y_bar_2, temp);
end
```

```
%% creating the system Markov parameters
Y_sys = obsMP(1);

sum = 0;                % initializing the variable

i = 1;                  % setting the index for the series
j = 1;                  % setting the index for the summation

% Builds the Parameters from Y_o_MP(2) until Y_o_MP(p)
while i <= p            % index of the series
    while j <= (i)    % index of the summation
       sum = sum + Y_bar_2(j+1)*Y_sys(i+1-j);
        j = j + 1;
    end
    Y_sys(i+1) = Y_bar_1(i+1) - sum;
    i = i + 1;
    sum = 0;
    j = 1;
end

% Builds the Parameters from Y_o_MP(p+1) until Y_o_MP(nmp)
sum = 0;                % resetting the summation
i = p+1;                % resetting the index for the series
j = 1;                  % resetting the index for the summation
while i <= nmp
    while j < (p+1)
        sum = sum + (Y_bar_2(j+1) * Y_sys(i-j+1));
        j = j + 1;
    end
    Y_sys(i+1) = -sum;
    i = i + 1;
    sum = 0;
    j = 1;
end
%% Computing Observer Gain Markov Parameters (Y_o)
Y_o_0 = obsMP(:,1);   % makes the first observer Markov Parameter
Y_o_1 = - obsMP(:,3);
```

```
Y_o_MP = cat(2,Y_o_0,Y_o_1); % Builds the first two terms array

sum = 0;                % resetting

i = 2;                  % setting the index for the series
j = 1;                  % setting the index for the summation

%% Builds the Parameters from Y_o_MP(2) until Y_o_MP(p)
while i <= p            % index of the series
    while j <= (i-1)    % index of the summation
        sum = sum + Y_bar_2(i)*Y_o_MP(i+1-j);
        j = j + 1;
    end
    Y_o_MP(i+1) = Y_bar_2(i+1) - sum;
    i = i + 1;
    sum = 0;

end

%% Builds the Parameters from Y_o_MP(p+1) until Y_o_MP(nmp)
sum = 0;
i = p+1;                % resetting the index for the series
j = 1;                  % resetting the index for the summation
while i <= nmp
    while j < (p+1)
        sum = sum + (Y_bar_2(j+1) * Y_o_MP(i-j+1));
        j = j + 1;
    end
    Y_o_MP(i) = -sum;
    sum = 0;
    i = i+1;
end

% Note y = Y_sys';
%compare OKID system Markov parameters to actual impulse response
%[y_act_impulse,X] = dlsim(A,B,C,D,u_impulse);
```

```
figure(’Name’,’OKID system Markov Parameters’,’Position’,[scrsz])
subplot(2,2,1)
plot(Y_sys,’bx’)
hold on
title(’System Markov parameters’)
ylabel(’Magnitude of output’)
legend (’System Markov Parameters’)

%% ERA
y = Y_sys’;   % for ease of coding

% Constructing the Henkel Matrix

[l,m,r]=size(y);
% l = number of data points
% m = number of outputs
% r = number of inputs

%d=q*m/r;    % q and d are related to the size of the Hankel matrix

[H_0, H_1] = make_hankel(y, N_cols, N_rows);

[R,E,S] = svd(H_0);         % Note: R*E*S’ = H_0
%% Plotting

%figure(’Name’,’Singular Values’)
subplot(2,2,2)
plot(diag(E),’*’)
title(’Singular Values’)
ylabel(’Singular Value’)
xlabel(’Index’)

n=input(’How many non-zero singular values are there for this system?’);

%%
local = axis/6; % [XMIN XMAX YMIN YMAX]
```

```
fprintf('\n\n\n\n##################### iD SYSTEM ####################')
fprintf('\n######################## RESULTS ########################')
fprintf('\n################## ERA and OKID METHOD ###############\n')

Pq = R(:,1:n)*E(1:n,1:n)^.5;  % Pq = R * E^.5   Observability Matrix
Qd = E(1:n,1:n)^.5*S(:,1:n)'; % Qd = E^.5 * S'  Controlability Matrix

Ad = pinv(Pq)*H_1*pinv(Qd)% PINV is Pseudoinverse denoted by (*)
Bd = Qd(:,1:r)            % Bd = Qd * Er  Bd is takes r columns from Qd
Cd = Pq(1:m,:)            % Cd = Em' * Pq Bd is takes m rows from Pq
Dd = y(1)                 % Dd = Yo

%% Calculating a Continuous system from the iD Discrete
sysd = ss(Ad,Bd,Cd,Dd,dt);
sysc = d2c(sysd);
[Acc,Bcc,Ccc,Dcc] = ssdata(sysc);

[num,den] = ss2tf(Acc, Bcc, Ccc, Dcc);

fprintf('\nThe full identified transfer function:')
printsys(num,den,'s');

%% Simulating the iD system
[y_pred_impulse,X] = dlsim(Ad, Bd, Cd, Dd, u_impulse);
[y_pred_actual,X] = dlsim(Ad, Bd, Cd, Dd, u_inputR);

%% Plotting

subplot(2,1,2)
plot(t, y_pred_impulse, 'b')
hold on
title('Response of the iD System to a Unit Impluse')
ylabel('Magnitude of output')
xlabel('Time (s)')

figure('Name','Compare iD system response','Position',[scrsz])
plot(t, y_pred_actual, '*b',t,ACC,'r')
```

```
hold on
title('Response of the ID System to given input')
ylabel('Magnitude of output')
xlabel('Time (s)')
legend('Model','Actual','Location','SouthEast');

saveas(1,'Images\falconOKIDdata.eps','psc2')
saveas(3,'Images\falconOKIDcomparison.eps','psc2')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n\n\n\n**********************************************')
fprintf('\n****************************************************')
fprintf('\n\n\n\n')
```

Appendix E

Operations Manual

Simulation
1. Ensure both the Simulation .m file and the model file (.mdl) are in the same folder and have different names
2. Set the system parameters at the top of the .m file, and the simulink model parameters after the LMI code section
3. Run the .m file. This will set the variables as have been set in the file, run the .mdl file, and then plot the desired variables as listed in the bottom of the file.

Experiment
1. Ensure that both the Phantom Omni and Novint Falcon are plugged in and connected to the computer. The Omni uses a firewire cable while the Falcon uses a USB cable.
2. Go to the directory "C: Program Files Novint Test Utilities" and run the test application. This is necessary to clear any communication errors and to home the device encoders.
3. Open the Phantom test utility from the start menu in the Sensable folder to home the device and ensure it is connected properly.
4. Open Matlab R2008b, and open the model "buildv1.mdl", and the .m file "Final_ver_v2".
5. Update the parameters of the .m file in the same manner as the simulation case.
6. Run the .m file to load the parameters into the Matlab workspace.
7. Go to the model file and make sure the Simulation Mode is set to External. Under the 'Quarc' menu select build. This may take a few minutes, and you should see the phrase "model name downloaded to target" on the main Matlab screen.
8. Go back to the Simulink block diagram and press the connect to target button to load the model to the real time kernel.
9. Check that the load cell and the amplification circuit are connected, with the amplification output going to Analog Input Channel 0. Plug in the power supply and turn it on to power the load cell.
10. Press the run button on the Simulink screen to run the experiment.
11. Open the file "expplotting.m" and set the plots you wish to see and run the code.

12. Save the data to a .mat file using the command ”save FILENAME” in the command prompt.