Output Feedback Bilateral Teleoperation with Force Estimation in the Presence of Time Delays

by

John M. Daly

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

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

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

Abstract

This thesis presents a novel bilateral teleoperation algorithm for n degree of freedom nonlinear manipulators connected through time delays. Teleoperation has many practical uses, as there are many benefits that come from being able to operate machines from a distance. For instance, the ability to send a remote controlled robotic vehicle into a hazardous environment can be a great asset in many industrial applications. As well, the field of remote medicine can benefit from these technologies. A highly skilled surgeon could perform surgery on a patient who is located in another city, or even country. Earth to space operations and deep sea exploration are other areas where teleoperation is quite useful.

Central to the approach presented in this work is the use of second order sliding mode unknown input observers for estimating the external forces acting on the manipulators. The use of these observers removes the need for both velocity and force sensors, leading to a lower cost hardware setup that provides all of the advantages of a position-force teleoperation algorithm. Stability results for this new algorithm are presented for several cases. Stability of each of the master and slave sides of the teleoperation system is demonstrated, showing that the master and slave are both stabilized by their respective controllers when the unknown input observers are used for state and force estimation. Additionally, closed loop stability results for the teleoperation system connected to a variety of slave side environments are presented. Delay-independent stability results for a linear spring-damper environment as well as a general finite-gain stable nonlinear environment are given. Delay-dependent stability results for the case where the slave environment is a liner spring-damper and the delays are commensurate are also presented. As well, stability results for the closed loop under the assumption that the human operator is modeled as a finite-gain stable nonlinear environment are given. Following the theoretical presentation, numerical simulations illustrating the algorithm are presented, and experimental results verifying the practical application of the approach are given.

Acknowledgements

I would first like to thank my supervisor, Dr. David Wang, for all of his help, both personally and professionally, over the last number of years. He has been a great source of encouragement, and the result of that encouragement is the completion of this research. He has always made himself available when it was needed and provided good direction and sound advice. I really enjoyed spending the time doing this research, and Dave is a major reason that it's been such a positive experience.

I would like to thank the professors on my committee for their help and involvement in my work. There have been many times when I have been able to go to them for direction and guidance, and they have been very willing and helpful.

I would also like to thank the members of our research group for their help and friendship over the past few years. In particular, Mike Tribou has been a great help in terms of helping me implement components of the numerical simulations, and for discussing research related ideas. Kevin Walker was instrumental in helping me get the robots working for the experimental results presented in the thesis. As well, Kevin did a great job performing the system identification of the robots. I'm very grateful for this help from my peers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Teleoperation is an area that holds a significant amount of interest to researchers. There can be many benefits to being able to operate machines from a distance. One example is the ability to send a remote controlled robotic vehicle into a hazardous environment [1]. Another useful application of teleoperation is in the field of remote medicine [2]. A highly skilled surgeon could perform surgery on a patient who is located in another city, or even country. Another example of a teleoperated system is in Earth to space operations, where someone on the surface of the Earth controls a robot in space [3]. Deep sea exploration is another application that benefits from teleoperation [4].

In such teleoperated systems, a major issue is that of delay in the communications channel. As the physical distance between two sites grows, the time required to transmit data across the network connecting the two sites can increase. As well, communications across the Internet suffer from unpredictability in the amount of time required to transmit. Since packets of data may take different routes to the remote end, congestion on the network due to other sources of traffic can also affect the time delay. In Earth to space operations, communication delays can also be significant – on the order of at least 0.4 seconds round-trip for vehicles in low earth orbit, and at least 3 seconds for vehicles near the moon [3].

In force reflecting teleoperation systems, where both velocity and force information are transmitted across the communication channel in a closed loop, even a small time delay can lead to instability of the closed loop [5].

The layout of a typical bilateral teleoperator system is given in Figure 1.1. An operator interacts with a master manipulator in order to control a slave manipulator. The slave manipulator could either be in free motion or in contact with the environment. When the operator moves the master manipulator, the position and/or velocity signals are transmitted across the communications network and used as the desired trajectory for the slave manipulator to follow. The slave then follows these delayed master signals. As the slave interacts with its environment, the force applied to it is sent back through the communications to the master side. The master manipulator then experiences a delayed version of the force acting on the slave manipulator. The *bilateral* teleoperation system is one in which signals are transmitted in both directions – from the master to the slave, and from the slave back to the master. This transmission of signals in two directions results in a closed loop between the master and slave. The work proposed here will focus exclusively on bilateral teleoperation.

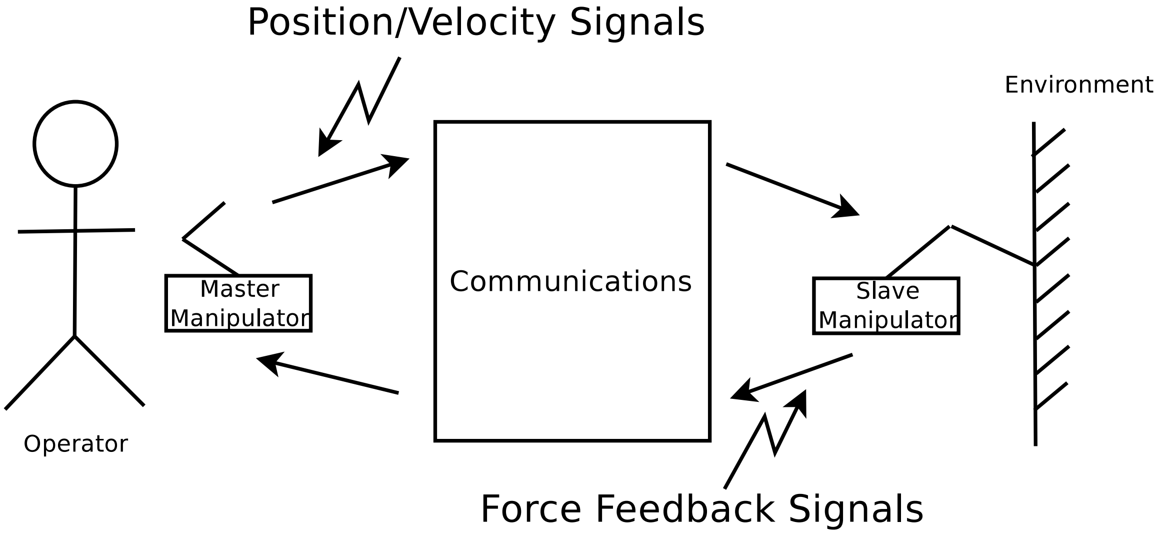


Figure 1.1: Typical layout of a bilateral teleoperation system.

There are several approaches to dealing with the time delay problem. Early results presented by Anderson and Spong [5] and Niemeyer and Slotine [6, 7, 8, 9] use passivity-based approaches. These approaches ensure that the communication link remains passive regardless of time delay, which is a condition for stability. However, as time delay increases, the performance of the systems suffers. A non-passive communication link, such as one with time delays, may add energy to the teleoperation system, destabilizing it.

While it is possible to stabilize a bilateral teleoperator with time delay, one desires more than stability. Ideally, a system that presents the operator with the sensation that he or she is in fact working with the remote environment in real time is desired. This desire leads to the examination of the performance of bilateral teleoperation systems. Typically *transparency* is used as the measure of performance. This concept will be discussed in detail in Chapter 2. The closer a teleoperator is to the ideal transparency condition, the better the sense of presence in the remote environment.

1.1 Original Contributions

While the field of teleoperation is one that has been under active research for several decades now, there are a number of unsolved problems. This thesis aims to address and resolve two of these problems.

Two popular teleoperation architectures, which will be discussed in detail in Chapter 2, are the position-position architecture and the position-force architecture. The position-position architecture allows for a more straightforward hardware setup. It does not rely on force sensors to measure human and environment forces. Instead, it transmits other signals such as positions, velocities, or position errors, to give some sense of the forces acting in the remote environment. However, the cost associated with this decision comes out in terms of performance. The position-position systems do not possess as high a degree of transparency as the position-force approaches. In this thesis, an approach based on sliding mode observers is used to generate force estimates of the human and environment forces. These estimates may be used in place of measurements. This leads to the first original contribution:

1. A novel bilateral teleoperation algorithm is developed that yields the performance, from a transparency point of view, of a position-force algorithm without requiring

the use of any force sensors in the system. Additionally, a thorough proof of stability of the novel algorithm including the sliding mode observers is developed.

While performance is an important issue to consider in control design, the issue of stability of the closed loop is at least as important. This work presents new closed loop stability results for a teleoperator system where the slave is in contact with a variety of environments. Many approaches to closed loop stability in the literature rely on the assumption that the slave environment is passive, and prove stability based on that. In this work, the assumption of a passive environment is not required. In particular, the second major contribution presented is:

2. A sufficient proof of stability is given for any teleoperator system where both the master and slave can be made into n linear mass spring damper systems in the closed loop and the environment is modeled as a set of linear spring dampers. Provided that the L_2 norm of the slave+environment system is known, a constructive method to select the master closed loop impedance parameters to ensure closed loop stability is presented. A second sufficient proof of stability is given for the same master and slave, but where an unknown nonlinear environment that is finite gain stable with a known upper bound on the gain is interacting with the slave. Finally, a necessary and sufficient stability result for the same master and slave with the set of n linear spring-dampers as the environment is introduced, provided that the time delays in the closed loop are commensurate. This assumption on commensurate time delay may be fulfilled in any teleoperation system in practice by making use of signal buffers at teach side of the teleoperation system.

1.2 Thesis Overview

The remainder of this document is organized as follows:

Chapter 2 provides the necessary background information and knowledge required to understand the rest of the thesis. The target audience of this work is the control research community. As such, elementary aspects of control are not covered here. However, an introduction to bilateral teleoperation is presented, along with an introduction to sliding mode observers and higher order sliding modes. Additionally, a review of the state of the art is given here.

Chapter 3 presents the novel bilateral teleoperation algorithm developed in this work. This chapter contains the major theoretical contribution. The algorithm is developed and stability is demonstrated for each of the master and slave sides individually. Following the proofs of stability for each of the master and slave sides, closed loop stability results for the entire teleoperator under various environment cases are presented. As well, a numerical example showing the computation of closed loop parameters in order to ensure stability is given.

Chapter 4 presents the numerical simulations and the experimental results from the novel algorithm. The numerical simulations confirm the effectiveness of the proposed approach in an environment that is close to ideal. The experimental results, performed on the University of Waterloo Teleoperation Platform, verify that the algorithm is useful in practice, in the presence of non-idealities such as unmodeled dynamics and sensor noise.

Chapter 5 concludes the thesis, summarizing the work and the results that have been presented. As well, suggestions for future research are given.

Chapter 2

Background

Bilateral teleoperation requires knowledge from several fields. In particular, one must be familiar with control, robotics, electrical systems, haptic systems, and in some cases, estimation. This chapter will begin by introducing some of the background knowledge required to understand bilateral teleoperation, including different teleoperation architectures. Following that, a review of sliding mode observers will be presented. These observers are fundamental to both the state and force estimation in the algorithm presented in this work. After introducing these concepts, a review of the state of the art of bilateral teleoperation will be given.

2.1 Bilateral Teleoperation

In bilateral teleoperation systems, as opposed to unilateral teleoperation systems, some sense of force feedback from the slave side is available to the operator [10]. Closing the loop in this way allows the operator to have some sensation of the slave side remote environment. While it is advantageous to have this sense of feedback, doing so complicates the stability of the teleoperator system, particularly in the presence of time delays. As well, decisions must be made in terms of which signals are transmitted between the master and slave sides. Different signals give rise to different teleoperation architectures. Two architectures of importance that will be discussed here are the position-position and position-force architectures.

In order to evaluate the performance of a teleoperation system, the notion of transparency is often used [11]. This concept refers to how "close" the experience is to what the operator would experience if interacting with the remote environment directly. Transparency is effectively a measure of the performance of the closed loop teleoperator. Intuitively, the system should feel like the master is connected to the remote environment by a virtual rod that has an infinitesimal mass and an infinitely large stiffness [12]. More formally, Lawrence [11] suggests that, given a task impedance of Z_s at the slave environment and an impedance Z_m which is experienced by the operator, the two must be equal. Perfectly transparent teleoperation is not possible in practice [11] but is a good measure of the performance of a teleoperation system.

Typically, one makes use of the hybrid matrix when analyzing the transparency of a teleoperation system. This matrix relates the operator force $F_h(t)$ and slave velocity $V_s(t)$ to the slave environment force $F_e(t)$ and master velocity $V_m(t)$. Expressing these signals in the Laplace domain, the hybrid matrix is given as [11],

$$\begin{bmatrix} F_h(s) \\ V_s(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} V_m(s) \\ -F_e(s) \end{bmatrix} \tag{2.1}$$

When the slave is in contact with the environment, the slave force is related to the slave velocity through the slave environment impedance Z_e. That is,

$$F_e = Z_e(V_e)$$

Similarly, the relationship between the force exerted by the operator on the master and the velocity of the master is given as,

$$F_h = Z_t(V_h)$$

where Z_h is the impedance experienced by the operator. In the case of ideal transparency, one desires that the operator not sense the teleoperation dynamics. That is, for a force $F_h(t)$ exerted by the operator, and a corresponding velocity $V_m(t)$ of the master, the slave responds such that $F_e(t) = F_h(t)$ and $V_s(t) = V_m(t)$. This leads to the ideal transparency matrix,

$$\begin{bmatrix} F_h(s) \\ V_s(s) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V_m(s) \\ -F_e(s) \end{bmatrix} \tag{2.2}$$

In the case of a time delayed bilateral teleoperator, with time delays of T_1 seconds from master to slave of T_2 seconds from slave to master, the ideal transparency matrix has the form [13],

$$\begin{bmatrix} F_h(s) \\ V_s(s) \end{bmatrix} = \begin{bmatrix} 0 & -e^{-T_2 s} \\ e^{-T_1 s} & 0 \end{bmatrix} \begin{bmatrix} V_m(s) \\ -F_e(s) \end{bmatrix} \tag{2.3}$$

All elements of the hybrid matrix have some physical meaning. The term $H_{11}(s) = F_h(s)/V_m(s)|_{F_e(s)=0}$ represents the force applied to the operator by the motion of the master when the slave is in free motion. In an ideal transparent setting, no force would be exerted on the operator by the master when the slave is out of contact and this term would be zero. The term $H_{12}(s) = F_h(s)/F_e(s)|_{V_m(s)=0}$ represents the force tracking characteristics of the system when the motion of the master is fixed. In the ideal case, the force acting on the master would be exactly the delayed slave environment force. The term $H_{21}(s) = V_s(s)/V_m(s)|_{F_e(s)=0}$ represents the velocity tracking of the master by the slave when the slave is in free motion. Ideally, the slave exactly tracks the delayed master velocity trajectory. Finally, the term $H_{22}(s) = V_s(s)/F_e(s)|_{V_m(s)=0}$ represents the admittance of the slave when the master is not in motion. The ideal value for this term is zero, indicating that when the master is fixed, the slave does not experience any motion due to environmental forces acting on it.

By examining the properties of the transparency matrix for a particular bilateral teleoperation algorithm, one can glean information about the transparency properties of the system, providing a measure of closeness to the ideal teleoperator.

2.1.1 Position-Position Architecture

The position-position teleoperation architecture and its variants, including the position-error-based architecture [14], work by transmitting functions of the position states of both the master and slave across the communication channel. In a strict position-position architecture, both the master and slave would have local position tracking controllers, whose goals are to ensure that each manipulator tracks the other [11].

Anderson and Spong [5] as well as Niemeyer and Slotine [9, 8] have both produced well known stable bilateral teleoperation algorithms based on a position-error-based architecture. A principle advantage of these approaches is in being able to have a sense of force feedback without requiring any force sensors in the system. Instead of force measurements, a measure such as the slave tracking error is transmitted from the slave side back to the master side. In an ideal case, if the slave were perfectly tracking the master while being in free motion, there would be no tracking error signal transmitted to the operator, and hence no perception of any environment force from the slave side. If the slave were in contact with an environment, the slave tracking error signal would grow. This tracking error would then be transmitted back to the master, giving a perception of the force acting on the slave manipulator to the operator. In general, this is not exactly the force acting on the slave, but instead is some combination of states that are related to the magnitude of this force.

Both Lawrence [11] and Tavakoli *et al.* [13] have performed transparency analyses of the position-position architecture. These analyses point out the difficulties of this approach. It turns out that the position-position approach does not provide a high degree of transparency. When the slave is in free motion, the operator perceives the dynamics of the actual teleoperation system [11]. This can lead to a sensation of sluggishness in the system. Any tracking error between the slave and master, even in free motion, is felt by the operator. This gives a sense to the operator of an additional inertia in the system that is not there. Tavakoli *et al.* [13] have also shown that this architecture suffers from non-ideal force tracking even when the slave is in contact with the environment. By performing a transparency analysis, they have shown that $H_{12}(s)$ is never equal to -1. This will always result in non-ideal force tracking [13]. Additionally, the entry $H_{11}(s)$ is not equal to zero. This leads to the sensation of reflected force to the master even when the slave is in free motion.

2.1.2 Position-Force Architecture

The position-force architecture typically involves an impedance controller at the master side, allowing the designer to specify a closed loop desired impedance for the master manipulator [11, 15]. This impedance controller is also responsible for transmitting the reflected slave forces to the master manipulator. The master states (position, velocity, and perhaps acceleration) are transmitted to the slave side for tracking of the master trajectory by the slave. The environmental force acting on the slave is transmitted back to the master side. In this scenario, only one manipulator (the slave) is concerned with tracking. At the master side, the control objective is to provide a desired closed loop impedance and to apply the reflected force. When these goals are accomplished, in the ideal case with no time delays, the master operator would experience the same interaction with the environment as the slave manipulator [11].

In this architecture, when the slave is in free motion, no force is reflected back from the slave to the master [11]. As a result, the master feels no additional inertial effects from the slave tracking error. Tavakoli *et al.* [13] have performed a transparency analysis, with the conclusion that perfect force tracking is obtained with this architecture. However, the perception of free motion is not ideal, as the term $H_{11}(s)$ in the transparency matrix is nonzero. This term is smaller than it is in the case of the position-position architecture [13], providing a better perception than the former architecture.

The position-force architecture is more prone to issues with stability than the position-position architecture [11]. In particular, Lawrence [11] performs an analysis of this architecture and notes that the loop gain is proportional to the environmental stiffness. As a result, a very stiff environment can lead to oscillations when the slave is in contact. By reducing the force feedback gain, however, this effect can be diminished. As well, this setup requires additional hardware in the way of force sensors. The use of these sensors can complicate implementation, as they are costly and tend to provide noisy measurements [14]. Despite these issues, it is possible to obtain a more transparent teleoperator with this architecture. As a result, one of the aims of the work presented in this thesis is to develop a teleoperation architecture with the hardware requirements of a position-position architecture that provides the benefits of a position-force architecture.

2.2 Sliding Mode Systems

In any real control or estimation problem, the system model is never perfectly known. As a result, unless model uncertainty is taken into account in the design, there will be errors. In the case of a control problem, this could result in a steady state tracking error or even instability. For the estimation problem, the result can be errors in the state estimates. Many techniques exist for dealing with model uncertainty for both linear and nonlinear systems. One such approach is the use of sliding mode techniques. Sliding mode systems may be implemented for both controllers and observers. This section will present the background for both the sliding mode control and estimation problems.

2.2.1 Sliding Mode Control

Modeling error will, to some degree, always be an issue when controlling real systems. If, for instance, the unmodeled dynamics in a system represented a term that lead to instability, and the control method used did not account for this, it could be possible that the control law designed for the system model would not stabilize the system itself. Consider feedback linearization, which relies on exact mathematical cancellation of the system nonlinearities. With such a control approach, it is not possible to cancel terms that are unknown.

As a result of these issues, robust control approaches are desirable. Sliding mode control is one such approach. This section will provide an introduction to the concept of sliding mode control. This development follows [16].

Consider a second-order nonlinear system expressed as,

$$\dot{x}_1 = x_2 \tag{2.4}$$

$$\dot{x}_2 = h(x) + g(x)u \tag{2.5}$$

where $h : R^2 \to R$ and $g : R^2 \to R$ are unknown nonlinear functions. The control objective for this system will be to drive the trajectories to a certain manifold in the state space of the system. This manifold is chosen by the designer. All that is known about g is that it is bounded from below by some constant $g_0 > 0$. The manifold, or sliding surface, that the system trajectories are driven to is defined as,

$$s = ax_1 + x_2 = 0 \tag{2.6}$$

where $a > 0$ is some constant. Assume also that the following inequality holds,

$$\left| \frac{ax_2 + h(x)}{g(x)} \right| \leq \rho(x), \qquad \forall x \in R^2 \tag{2.7}$$

Note that the derivative of the sliding surface (2.6) is expressed as,

$$\dot{s} = ax_2 + h(x) + g(x)u \tag{2.8}$$

One desires to drive the system to the manifold $s = 0$ in finite time. On this manifold, the equation $x_2 = -ax_1$ holds. Substituting this into (2.4), one arrives at,

$$\dot{x}_1 = -ax_1 \tag{2.9}$$

Therefore, if the manifold $s = 0$ can be reached with the sliding mode controller, the origin of the system can be stabilized. This particular choice of sliding surface leads to a reduction in model order in the closed loop. The system is governed by the first order dynamics (2.9) on the sliding surface. The task remains to design a control law u that will drive the system to the sliding surface. Before developing the control law, note that the sliding dynamics are given as,

$$\dot{s} = ax_2 + h(x) + g(x)u \tag{2.10}$$

Taking the first time derivative of the sliding surface $s(t)$, one arrives at (2.10). Note that the control input u appears in the first derivative of the sliding surface. Systems of this nature are said to have a *relative degree* of one. Classical sliding mode control is limited to systems of relative degree one. That is, the control input must appear in the *first* derivative of the sliding surface in order for classical sliding mode control to be applicable.

In order to develop the control law, one seeks to stabilize the sliding dynamics (2.10). Consider the Lyapunov function candidate,

$$V = \frac{1}{2}s^2 \tag{2.11}$$

Taking the derivative of (2.11), evaluating it along the trajectories of the system (2.10), and rearranging, one arrives at,

$$\dot{V} = g(x)s\underbrace{\left(\frac{ax_2 + h(x)}{g(x)}\right)}_{\leq \rho(x)} + g(x)su \tag{2.12}$$

Then,

$$\begin{aligned} \dot{V} &\leq g(x)s\rho(x) + g(x)su \qquad (2.13) \\ &\leq g(x)|s|\rho(x) + g(x)su \qquad (2.14) \end{aligned}$$

Now choose the following control law,

$$u = -\beta(x)\text{sgn}(s) \tag{2.15}$$

where $\beta(x) = \rho(x) + \beta_0$, and $\beta_0 > 0$ is a parameter chosen by the designer. Substituting $\beta(x)$ into (2.14), one arrives at,

$$\dot{V} \leq -g(x)|s|\beta_0 \tag{2.16}$$

$$\leq -g_0\beta_0|s| \tag{2.17}$$

In order to show finite time convergence of the system states to the sliding surface dynamics, the Comparison Lemma [16, p. 102] is introduced. It is used to compute an upper bound on the convergence time to the sliding surface. Prior to introducing the lemma, the upper right-hand derivative $D^+v(t)$ is defined as [16, p. 659],

$$D^+v(t) = \lim \sup_{h \to 0^+} \frac{v(t+h) - v(t)}{h}$$

The lemma is expressed in terms of the upper right-hand derivative so that it applies even when $v(t)$ is not differentiable. However, if the function $v(t)$ is differentiable at time t, then $D^+v(t) = \dot{v}(t)$ [16]. This will be the case for functions in this work.

Lemma 2.2.1. ***(Comparison Lemma)*** *Consider the scalar differential equation,*

$$\dot{u} = f(t,u), \qquad u(t_0) = u_0$$

where $f(t,u)$ is continuous in t and locally Lipschitz in u, for all $t \geq 0$ and all $u \in J \subset R$. Let $[t_0, T)$ (T could be infinity) be the maximal interval of existence of the solution $u(t)$, and suppose $u(t) \in J$ for all $t \in [t_0, T)$. Let $v(t)$ be a continuous function whose upper right-hand derivative $D^+v(t)$ satisfies the differential inequality

$$D^+v(t) \leq f(t, v(t)), \qquad v(t_0) \leq u_0$$

with $v(t) \in J$ for all $t \in [t_0, T)$. Then, $v(t) \leq u(t)$ for all $t \in [t_0, T)$.

In order to make use of this lemma to show finite time convergence of the system trajectories to $s = 0$, define the function $W = \sqrt{2V} = |s|$. Then, the upper-right hand derivative of W may be expressed as,

$$D^{+}W = \dot{W} = \frac{1}{\sqrt{2V}}\dot{V} = \frac{1}{|s|}\dot{V} \leq \frac{-|s|}{|s|}g_0\beta_0 = -g_0\beta_0 \tag{2.18}$$

So $D^{+}W \leq -g_0\beta_0$. Define $f(t, W) = -g_0\beta_0$. Then $D^{+}W \leq f(t, W)$. Now consider $\dot{u} = f(t, u) = -g_0\beta_0$, $u(t_0) = W(s(t_0))$. Then,

$$\begin{aligned} \int_{t_0}^{t} \frac{du}{d\tau} d\tau &= -g_0\beta_0 \int_{t_0}^{t} d\tau & (2.19) \\ u(t) - u(t_0) &= -g_0\beta_0(t - t_0) & (2.20) \\ u(t) &= u(t_0) - g_0\beta_0(t - t_0) & (2.21) \end{aligned}$$

Making use of Lemma 2.2.1, the result $W(t) \leq u(t)$ is obtained. Then,

$$W(t) \leq W(s(t_0)) - g_0\beta_0(t - t_0) \tag{2.22}$$

Note that this inequality would be valid for all time for any $W(t)$ satisfying $D^{+}W \leq -g_0\beta_0$ where the range on $W(t)$ is not restricted. From (2.22), it is apparent that at some time the function will reach zero. In this case $W(t) = |s(t)|$ so $W(t)$ is restricted to positive values. Therefore, from (2.22), it is clear that after some finite time the value of $|s(t)|$ will reach zero, and the inequality (2.22) will hold up until that time. At this point the system trajectories have reached the sliding surface $s = 0$. The Comparison Lemma guarantees that the sliding surface will be reached in finite time. Due to the inequality (2.17), once the trajectories have reached the sliding surface, they will not leave. At this point, the closed loop system will be governed by the reduced order dynamics $\dot{x}_1 = -ax_1$. Of note in this example is the fact that, despite the uncertainty associated with h and g, the sliding mode controller is able to drive the system to the manifold $s = 0$ in finite time and keep it there. This fact highlights the robustness properties of sliding mode control. In practice, one must be able to exert enough actuator effort to ensure that the sliding motion can be induced. However, as long as that is possible, a very robust control strategy is available.

The application of a discontinuous control law, as in (2.15), results in a system with a discontinuity in the right hand side. As a result, standard existence and uniqueness results from the theory of ordinary differential equations do not apply. Filippov [17] developed much of the theory on fundamental issues such as existence and uniqueness of solutions to differential equations driven by discontinuous functions. The interested reader is encouraged to see [17] for an in depth treatment of the topic.

When a system's trajectories are on the sliding manifold, the sliding mode controller switches infinitely fast in any given finite time interval [18]. Utkin [19] has developed the method of *equivalent control* in order to analyze the system's behaviour on the sliding surface. The fact that the actual control signal exhibits fast switching makes the use of conventional analysis techniques difficult. As a result, Utkin proposes finding the so-called equivalent control signal on the surface. Under the assumption that $s = 0$, and as a result that $\dot{s} = 0$, one may compute the continuous control signal that would also lead to this behaviour. Since systems for which classical sliding mode control is applied have relative degree one in the sliding surface, the control signal appears in the term $\dot{s}$. Solving for this control signal in $\dot{s} = 0$ yields the equivalent control signal u_{eq}. For the system presented here, the time derivative of the sliding surface is,

$$\dot{s} = ax_2 + h(x) + g(x)u = 0 \tag{2.23}$$

when the system is in the sliding mode. Solving for the equivalent control signal one arrives at,

$$u_{eq} = \frac{-ax_2 - h(x)}{g(x)} \tag{2.24}$$

This signal effectively represents the control signal that is applied to the plant under the ideal case of infinitely fast switching. In practice, a signal cannot switch infinitely fast in a finite period of time. The real control signal applied switches at a finite frequency, and contains both high frequency and low frequency components [19]. The high frequency terms are filtered out by the plant, while the low frequency terms determine the plant's behaviour on the sliding surface. These low frequency terms represent, practically, the value of the equivalent control signal (2.24). If one is interested in obtaining the equivalent control signal, a low pass filtering operation on $u(t)$ may be performed. The bandwidth of the low pass filter should be such that the high frequency component is well filtered out, and the lower frequency equivalent control is not distorted [19]. This technique will be used with the sliding mode observers presented in this work in order to obtain external force input estimates from the switching components of the observers.

In practice, the issue of generating a fast, but finite frequency, switching signal and applying it to a plant can be troublesome. As well, in real cases there are imperfections in actuators and delays in the system. These factors lead to the phenomenon known as *chattering* [16]. Instead of the ideal situation, where the state trajectory stays exactly on the manifold once reaching it, the trajectory is affected by delays in the control action and will cross the sliding surface. Once on the other side of the sliding surface, the sign of the control signal will change, driving the system back toward the surface. Again, delays in the control action will allow the trajectory to cross over to the other side of the sliding surface. This pattern repeats, effectively causing an oscillating behaviour. This effect is known as chattering [16]. Chattering causes increased wear on mechanical components, results in a loss of accuracy in the control, and may excite unmodeled high frequency dynamics in the plant. This behaviour can be quite obvious in mechanical systems experiencing it, as the plant can appear to shake at a very high frequency.

There are several approaches that may be used to eliminate chattering. Utkin *et al.* present a number of them in [19]. One common approach is to replace the signum, which is a discontinuous function, with a high-slope saturation function [16]. Instead of the control law (2.15), consider the following control law,

$$u = -\beta(x)\text{sat}\left(\frac{s}{\epsilon}\right) \tag{2.25}$$

where the saturation function, sat$(\cdot)$, is defined as,

$$\mathrm{sat}(x) = \begin{cases} x, & \text{if } |x| \leq 1 \\ \mathrm{sgn}(x), & \text{if } |x| > 1 \end{cases}$$

and ϵ is a positive parameter which determines the slope of the saturation function. The slope is given by $1/\epsilon$. The high-slope portion of the saturation function gives rise to what is termed a boundary layer. The system trajectories are said to lie withing the boundary layer when $|s| \leq \epsilon$. An analysis of a sliding mode controller that makes use of a boundary layer is carried out in the same fashion as the case where a pure switching component is used, as long as the system is *outside* of the boundary layer. That is, in the same way as was done for the pure switching sliding mode controller, finite time convergence of the system to the *boundary layer* made be shown. Once the trajectory has reached the set $\{|s| \leq \epsilon\}$, it will remain inside for all time thereafter [16]. In general, the behaviour of the system inside the boundary layer is problem-dependent and must be analyzed on a case by case basis. However, it is possible to demonstrate ultimate boundedness of the system inside the boundary layer in general. The magnitude of this bound can be decreased by decreasing epsilon. If the boundary layer is made to be too narrow, chattering can still occur. A trade-off must be made between control accuracy and the presence of chattering in the system.

This section has presented a brief introduction to sliding mode control as a robust control strategy. The interested reader is referred to [16] and [19] for a more in depth discussion of sliding mode control. With an understanding of the basics of sliding mode control, the next section will present sliding mode observers.

2.2.2 Sliding Mode Observers

Typically one is interested in the use of state observers to reconstruct the state of a system from partial state measurements. For instance, in a mechanical system only positions may be measured, but one may need access to both positions and velocities. An observer is used to provide an estimate of the system state from only the position measurements. As in the case of control, one encounters modeling error in the system models used in observers. As a result, to ensure accurate state estimates, a technique for robust estimation is desirable. Consider a standard observable SISO linear system,

$$\dot{x}(t) = Ax(t) + B\left(u(t) + \xi(t)\right) \tag{2.26}$$

$$y(t) = Cx(t) \tag{2.27}$$

where $x(t) \in R^n$ is the state, $u(t) \in R$ is the controlled input, $y(t) \in R$ is the output, and $\xi(t) \in R$ is some unknown input, disturbance, or unmodeled dynamics. A standard Luenberger observer for this system is expressed as,

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(y(t) - C\hat{x}(t)) \tag{2.28}$$

where $\hat{x}(t)$ is the estimated state and K is the observer gain matrix. Note the absence of the unknown term $\xi(t)$, since it is unavailable to the designer. Defining the estimation error $e(t)$ as $e(t) = x(t) - \hat{x}(t)$ yields the following estimation error dynamics,

$$\dot{e}(t) = (A - KC)e(t) + B\xi(t) \tag{2.29}$$

Note here that the observer error dynamics may be made asymptotically stable by proper choice of K, but are driven by the term $\xi(t)$. As a result, for non-zero $\xi(t)$, the estimation error will not converge to zero. This example serves to show that classical Luenberger observers are not robust to disturbances or modeling uncertainties. If $\xi(t)$ were zero for all time, the estimation error would converge to zero asymptotically.

Owing to the robustness properties of sliding mode systems, sliding mode observers may be designed in order to deal with modeling uncertainties and disturbances. Early work in sliding mode observers focused on a structure that is very similar to a standard full order observer but with the replacement of the linear innovative term (the term of the form $K(y-\hat{y})$) with a discontinuous term of the form $K\text{sgn}(y-\hat{y})$. Under a suitable choice of the sliding gain, the observer trajectory converges to the manifold $y - \hat{y} = 0$ in finite time despite the possible presence of unknown disturbances. However, in [20] Drakunov and Utkin propose a new approach to the design of sliding mode observers that makes use of the equivalent control concept. In such an observer setup, the observer states converge one by one to a sliding manifold in finite time, until at some point the entire state estimate is available. Haskara *et al.* propose a design procedure for both continuous and discrete versions of this type of sliding mode observer in [21]. The work presented in [21] improves over the work in [20] in that the final observer equations are expressed in the original system coordinates, as opposed to an observer design requiring a transformation of the state variables.

An equivalent output injection first order sliding mode unknown input observer will now be developed for a nonlinear system in triangular form. Stability of the observer and finite time convergence of the estimated states to the true states despite unknown disturbances will be demonstrated. Consider a system of the form,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \\ h(x) + g(x)(u + \xi) \end{bmatrix} \tag{2.30}$$

with output $y = x_1$. The state vector is defined as $x = [x_1, x_2, \ldots, x_n]^T$ where each $x_i \in R$. As well the functions $h(x)$ and $g(x)$ are specified as $h : R \mapsto R$ and $g : R \mapsto R$. Here, $u \in R$ is the control input and $\xi \in R$ is an unknown input or disturbance. Though the development is limited to systems in this triangular form, many real systems are expressed this way. Mechanical systems with friction uncertainties fit in to this model structure. Modeling of mechanical systems typically results in systems with position and velocity states, where the control and disturbances enter through the derivative of the velocity state. The uncertain friction could be viewed as an unknown disturbance to the model, which would be rejected by this observer. As well, many nonlinear systems for which a diffeomorphism exists to transform the system into a form suitable for feedback linearization [16] can be expressed in the triangular form given above. In this case, the observer would be designed to estimate the transformed states, and not the original system states.

The following assumptions are made on the plant and the inputs,

Assumption 2.2.1. *The plant, given by (2.30), may be unstable but it does not have a finite escape time.*

Assumption 2.2.2. *The term $g(x)$ is bounded away from zero $\forall\, x \in R^n$.*

Assumption 2.2.3. *The control input $u(t)$ belongs to the extended L_p space*[1] *denoted as L_{pe}.*[2] *That is, any truncation of $u(t)$ to a finite time interval is essentially bounded.*

Assumption 2.2.4. *The unknown disturbance input $\xi(t)$ is bounded for all time with some known upper bound ξ_0.*

In order to estimate the plant states, the following equivalent output injection first order sliding mode observer can be used,

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \vdots \\ \dot{\hat{x}}_{n-1} \\ \dot{\hat{x}}_n \end{bmatrix} = \begin{bmatrix} \hat{x}_2 + \lambda_1 \mathrm{sgn}(y - \hat{x}_1) \\ \hat{x}_3 + E_1(\lambda_2 \mathrm{sgn}(\tilde{x}_2 - \hat{x}_2)) \\ \vdots \\ \hat{x}_n + E_{n-2}(\lambda_{n-1} \mathrm{sgn}(\tilde{x}_{n-1} - \hat{x}_{n-1})) \\ h(\hat{x}) + g(\hat{x})u + E_{n-1}(\lambda_n \mathrm{sgn}(\tilde{x}_n - \hat{x}_n)) \end{bmatrix} \tag{2.31}$$

where $\tilde{x}_i = \hat{x}_i + (\lambda_{i-1}\mathrm{sgn}(x_{i-1} - \hat{x}_{i-1}))_{eq}$. The term $E_i = 0$ if $x_i - \hat{x}_i \neq 0$ and $E_i = 1$ otherwise. While these terms are not necessary for stability, they are used as an anti-peaking structure [23]. Before an observer state has converged, its information is not used

[1] From [22], for each real $p \in [1, \infty)$, the set $L_p[0, \infty)$ consists of all measurable functions $f(\cdot) : R_+ \mapsto R$ such that $\int_0^\infty |f(t)|^p dt < \infty$. The set L_p^n consists of all n-tuples $f = [f_1 \dots f_n]^T$ where $f_i \in L_p$ for each i.

[2] From [22], the set L_{pe} consists of all measurable functions $f : R_+ \mapsto R$ with the property that the truncation of f to the interval $[0, T]$ is in L_p for all finite T, and is called the *extension of* L_p or the *extended* L_p-space.

in the next observer state down. This approach prevents peaks in the observer estimates as it is converging. The equivalent output injection terms $(\lambda_{i-1}\text{sgn}(\tilde{x}_{i-1} - \hat{x}_{i-1}))_{eq}$ are obtained by passing the terms $\lambda_{i-1}\text{sgn}(\tilde{x}_{i-1} - \hat{x}_{i-1})$ through a low pass filter. In practice, an ideal sliding mode cannot exist and so trajectories chatter around the sliding manifold [21]. According to Haskara *et al.* [21] the discontinuous term may be considered as a combination of the low frequency equivalent control and the high frequency switching signal. Passing this through a low pass filter with a bandwidth greater than the system bandwidth but smaller than the switching frequency, one may obtain the equivalent output injection term.

This observer is designed so that the state estimates converge one at a time in finite time, until eventually the entire state estimate is available. Once a particular state $\hat{x}_i$ has converged to the true state, the corresponding term E_{i+1} is set to one. This allows the next state to have the equivalent output injection information so that it may converge. A similar design is presented in [24] but does not take into account disturbances in the plant. Note that, in general, convergence of the first observer state $\hat{x}_1$ to the actual plant state x_1 does not imply that immediately $\hat{x}_2 = x_2$ despite the fact that $\dot{x}_1 = x_2$ in the plant structure. The reason for this is due to the selection of the initial conditions in the observer. If the observer initial state happened to be the same as the plant initial state then the state estimates would be equal to the plant states immediately. However, if one considered the case where every observer state were initialized to a value different than the actual plant state, each observer state would require time to converge to the actual plant state. An example showing this step by step convergence behaviour is given in our previous work [25].

In order to show convergence of the observer error dynamics to zero in finite time, an expression for the observer error dynamics must first be developed.

Defining the observer error as $e = x - \hat{x}$, the observer error dynamics are obtained from (2.30) and (2.31) and are expressed as,

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_{n-1} \\ \dot{e}_n \end{bmatrix} = \begin{bmatrix} e_2 - \lambda_1 \mathrm{sgn}(e_1) \\ e_3 - E_1(\lambda_2 \mathrm{sgn}(\tilde{x}_2 - \hat{x}_2)) \\ \vdots \\ e_n - E_{n-2}(\lambda_{n-1} \mathrm{sgn}(\tilde{x}_{n-1} - \hat{x}_{n-1})) \\ \Delta h(x, \hat{x}) + \Delta g(x, \hat{x})u + g(x)d - E_{n-1}(\lambda_n \mathrm{sgn}(\tilde{x}_n - \hat{x}_n)) \end{bmatrix} \tag{2.32}$$

where $\Delta h(x, \hat{x}) = h(x) - h(\hat{x})$, $\Delta g(x, \hat{x}) = g(x) - g(\hat{x})$.

The next theorem shows finite time convergence of the sliding mode observer, despite the presence of a disturbance term or unmodeled dynamics in the plant.

Theorem 2.2.1. *Consider the plant (2.30), observer (2.31), and Assumptions 2.2.1 to 2.2.4. Then, there exists observer gains λ_1 to λ_n such that the estimated state $\hat{x}$ converges to the actual state x in finite time. Further, after some finite time T_n seconds, an estimate of the unknown disturbance ξ is available as $\xi = g(x)^{-1}(\lambda_n sgn(e_n))_{eq}$.*

Proof. See Appendix A.1. □

This result has shown that a sliding mode observer, even in the presence of unknown but bounded disturbances, yields finite time convergence of the state estimates to the true states. As well, an estimate of the unknown disturbance/unmodeled dynamics is obtained. In terms of robustness, then, the sliding mode observer is superior to the standard Luenberger observer.

2.2.3 Higher Order Sliding Modes

While the previous subsection has demonstrated that it is possible to construct sliding mode observers that provide both state and unknown input estimates in finite time, some issues remain with this approach. Standard first order sliding mode algorithms, such as the one already presented, suffer from chattering. The presence of the discontinuous switching term in the dynamics is what leads to this phenomenon [16]. When a sliding mode controller is used with a physical system, this discontinuity in the control can lead to chattering, due to the finite bandwidth of the controller. This chattering can cause excitation of unmodeled high frequency dynamics, which can be hard on the system itself.

Higher order sliding mode control [26, 27] is an approach to deal with these practical issues that arise in classical, first order, sliding mode systems. Instead of having a discontinuity appear in directly in the control signal, higher order sliding mode control has the discontinuity appear in higher derivatives of the control signal. As a result, the discontinuity is integrated before reaching the plant, yielding a continuous control signal. This appearance of the discontinuity in higher derivatives of the control signal is why it is termed higher order sliding mode control. As well, first order sliding mode control is limited to systems that are relative degree one in the sliding dynamics, but an r-th order sliding mode controller may be designed for systems that are relative degree r in the sliding dynamics [28]. This makes it possible to extend the concept of sliding mode control to a broader class of systems.

Another advantage of higher order sliding mode controllers and observers is their accuracy in a discrete time implementation. When a first order sliding mode controller is implemented in a discrete way, such as using a digital computer, the precision of the sliding mode is proportional to the sample period [29]. However, in implementing an r-th order sliding mode controller, one may be able to achieve precision up to the r-th power of the sample period used in the system [29].

A number of second order sliding mode algorithms have been developed for systems that are both relative degree one and relative degree two in the sliding dynamics [26]. The twisting algorithm is a well-known second order sliding mode controller for uncertain relative degree two systems. Following [26], consider the single input system,

$$\dot{x}(t) = A(t)x(t) + b(t)u(t) \tag{2.33}$$

$$s(t) = \langle c(t), x(t) \rangle + \xi(t) \tag{2.34}$$

where $x(t) \in R^n$ is the system state and $s : [0, \infty) \to R$ is the sliding surface. The function

$c : [0, \infty) \to R^n$ is specified by the designer to design the sliding surface, as is the function $\xi(t) : [0, \infty) \to R$. The inner product here is taken to be the dot product. The structure of $A(t)$ and $b(t)$ need not be known, but upper bounds on those functions and their first derivatives must be known. As well, the sliding surface $s(t)$ must have relative degree two. That is, the control signal $u(t)$ must first appear in the second derivative of $s(t)$. Then, the twisting controller for this system is given as,

$$\dot{u}(t) = \begin{cases} -u(t) & \text{for } |u(t)| > 1 \\ -\alpha_m \text{sign}(s(t)) & \text{for } s(t)\dot{s}(t) \leq 0,\ |u(t)| \leq 1 \\ -\alpha_M \text{sign}(s(t)) & \text{for } s(t)\dot{s}(t) > 0,\ |u(t)| \leq 1 \end{cases} \tag{2.35}$$

where $\alpha_M > \alpha_m > 0$. For suitable choices of the parameters α_m and α_M [26], it can be shown that this controller yields a second order sliding mode and converges to the manifold in a finite time. Note that a system of this setup cannot be controlled using a first order sliding mode controller due to the relative degree two assumption of the sliding surface. One practical implementation issue for the twisting algorithm is the necessity of having access to the term $\dot{s}(t)$. Levant [26] proposes the use of the first difference of $s(t)$ since one only actually requires the signum of $\dot{s}(t)$ in (2.35).

The super-twisting algorithm is another type of second order sliding mode controller. However, this algorithm is designed for relative degree one systems [30]. The advantage of using this second order algorithm with a relative degree one system is the elimination of chattering [26]. As well, this algorithm does not require measurement of the first time derivative of the sliding surface, making implementation no more demanding than a first order sliding mode algorithm. It also features finite time convergence of the system trajectories to the sliding surface and has the advantage of giving more precision around the sliding surface in a discrete time implementation as compared to a first order sliding mode algorithm. The observers used in this work are based on the super-twisting algorithm so its details will be presented in the next section.

2.2.4 Second Order Sliding Mode Observer for Mechanical Systems

The observers used in this work are based on the super-twisting SISO observers presented in [31]. Here, the work in [31] is extended and MIMO observers are designed and a corollary is developed to prove that the SISO observers may be extended to MIMO observers while still ensuring finite time convergence of the observer error dynamics to zero.

First, the SISO version of the observers is presented in order to show stability in this case. The proof for the SISO observer is not a new result, but is found in [31]. Similar proofs for second order sliding mode controllers are also found in [27] and [26]. In this work, a proof was re-developed based on these references in order to include more detail.

Consider the following plant,

$$\begin{aligned} \dot{x}_1 &= x_2 && (2.36) \\ \dot{x}_2 &= f(t, x_1, x_2, u) + \xi(t, x_1, x_2, u) && (2.37) \\ y &= x_1 && (2.38) \end{aligned}$$

where $x_1 \in R$ is the plant position, $x_2 \in R$ is the plant velocity, $f(t, x_1, x_2, u)$ is the nominal known part of the dynamics and $\xi(t, x_1, x_2, u)$ is the unmodeled, unknown part of the dynamics. In order to estimate the states and unknown input, consider the following observer,

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + z_1 && (2.39) \\ \dot{\hat{x}}_2 &= f(t, x_1, \hat{x}_2, u) + z_2 && (2.40) \end{aligned}$$

where $\hat{x}_1 \in R$ and $\hat{x}_2 \in R$ are the position and velocity estimates. The variables $z_1 \in R$ and $z_2 \in R$ are the output injections given as,

$$z_1 = \lambda|x_1 - \hat{x}_1|^{1/2}\text{sign}(x_1 - \hat{x}_1) \tag{2.41}$$

$$z_2 = \alpha\text{sign}(x_1 - \hat{x}_1) \tag{2.42}$$

It is assumed that $\hat{x}_1(0) = x_1(0)$ and $\hat{x}_2(0) = 0$. This choice is always possible since x_1 is measured. Defining the estimation error as $\tilde{x}_1 = x_1 - \hat{x}_1$ and $\tilde{x}_2 = x_2 - \hat{x}_2$, the error dynamics are given as,

$$\dot{\tilde{x}}_1 = \tilde{x}_2 - \lambda|\tilde{x}_1|^{1/2}\text{sign}(\tilde{x}_1) \tag{2.43}$$

$$\dot{\tilde{x}}_2 = F(t, x_1, x_2, \hat{x}_2, u) - \alpha\text{sign}(\tilde{x}_1) \tag{2.44}$$

where $F(t, x_1, x_2, \hat{x}_2, u) = f(t, x_1, x_2, u) - f(t, x_1, \hat{x}_2, u) + \xi(t, x_1, x_2, u)$. Assume that the states can be bounded in the domain of interest, then one can assume the existence of a constant f^+ such that,

$$|F(t, x_1, x_2, \hat{x}_2, u)| < f^+ \tag{2.45}$$

holds for any possible t, x_1, and x_2 in the operational domain. As long as the controller used would stabilize the process in the case of full state measurements, one can choose the observer error dynamics to be fast enough so that the state estimates are recovered before the plant leaves some chosen area. This will ensure that the bound (2.45) remains satisfied in the operational domain [31]. Let α and λ satisfy the following inequalities,

$$\alpha > f^+ \tag{2.46}$$

$$\lambda > \sqrt{\frac{2}{\alpha - f^+}}\frac{(\alpha + f^+)(1 + p)}{(1 - p)} \tag{2.47}$$

where p is some chosen constant such that $0 < p < 1$.

Theorem 2.2.2. *Suppose that the parameters of the observer are selected according to the above conditions for α and λ and condition (2.45) holds for the plant. Then, the variables of the observer converge in finite time to the states of the system, i.e. $(\hat{x}_1, \hat{x}_2) \rightarrow (x_1, x_2)$. Further, the unknown term $\xi(t, x_1, x_2, u)$ may be recovered in finite time as $\alpha \text{sign}(\tilde{x}_1)_{eq}$.*

Proof. See Appendix A.2. □

Now consider an n-DOF plant,

$$\begin{aligned} \dot{x}_1 &= x_2 & (2.48) \\ \dot{x}_2 &= f(t, x_1, x_2, u) + \xi(t, x_1, x_2, u) & (2.49) \end{aligned}$$

where $x_1 \in R^n$, $x_2 \in R^n$, $u \in R^n$, and $y = x_1$. Here, $f(t, x_1, x_2, u)$ represents the known plant dynamics, while $\xi(t, x_1, x_2, u)$ represents the unknown inputs/dynamics. In this case, this term represents the unknown force input. Also, $f \in R^n$ may be expressed as,

$$f(t, x_1, x_2, u) = \begin{bmatrix} f_1(t, x_1, x_2, u) \\ f_2(t, x_1, x_2, u) \\ \vdots \\ f_n(t, x_1, x_2, u) \end{bmatrix}$$

and equivalently for $\xi(t, x_1, x_2, u) \in R^n$.

Now consider the following observer for the system (2.48) and (2.49),

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + z_1 & (2.50) \\ \dot{\hat{x}}_2 &= f(t, x_1, \hat{x}_2, u) + z_2 & (2.51) \end{aligned}$$

where $\hat{x}_1 \in R^n$ and $\hat{x}_2 \in R^n$ are the position and velocity estimates. As well, $z_1 \in R^n$ and $z_2 \in R^n$. The i-th element of vector z_1 is given as,

$$z_{1_i} = \lambda_i |x_{1_i} - \hat{x}_{1_i}|^{1/2}\text{sign}(x_{1_i} - \hat{x}_{1_i}) \tag{2.52}$$

and the i-th element of vector z_2 is given as,

$$z_{2_i} = \alpha_i \text{sign}(x_{1_i} - \hat{x}_{1_i}) \tag{2.53}$$

It is assumed that $\hat{x}_1(0) = x_1(0)$ and $\hat{x}_2(0) = 0$. This is a reasonable assumption, as $x_1(t)$ is measured so it is always possible to initialize the observer with $\hat{x}_1(0) = x_1(0)$. Now define the term $F(t, x_1, x_2, \hat{x}_2, u) \in R^n$ as,

$$\begin{aligned} F(t, x_1, x_2, \hat{x}_2, u) &= f(t, x_1, x_2, u) - f(t, x_1, \hat{x}_2, u) \\ &\quad + \xi(t, x_0, x_2, u) \end{aligned} \tag{2.54}$$

and assume that the inequality,

$$|F_i(t, x_1, x_2, \hat{x}_2, u)| < f_i^+ \tag{2.55}$$

holds over the operational domain. Provided that the controller used will stabilize the plant in the case of full state measurements, one could choose the observer error dynamics to be fast enough to ensure convergence of the state estimates before the plant leaves some chosen area [31]. This would ensure that the above bound remain satisfied. Let α_i and λ_i satisfy the following inequalities, for every element i,

$$\alpha_i > f_i^+ \tag{2.56}$$

$$\lambda_i > \sqrt{\frac{2}{\alpha_i - f_i^+}} \frac{(\alpha_i + f_i^+)(1 + p_i)}{(1 - p_i)} \tag{2.57}$$

where p_i is some chosen constant such that $0 < p_i < 1$. Then, one can state the following corollary,

Corollary 2.2.1. *Suppose that the parameters for the observer (2.50) and (2.51) are selected according to the above conditions (2.56) and (2.57) for α and λ, and that condition (2.55) holds over the operational domain of the plant. Then, the variables of the observer converge in finite time to the states of the system, i.e. $(\hat{x}_1, \hat{x}_2) \rightarrow (x_1, x_2)$. Further, the i-th component of the unknown vector $\xi(t, x_1, x_2, u)$ may be recovered in finite time as $\alpha_i \operatorname{sign}(x_{1_i} - \hat{x}_{1_i})_{eq}$.*

Proof. With the plant (2.48) and (2.49) the dynamics may be equivalently expressed as,

$$\begin{aligned}
\dot{x}_{1_1} &= x_{2_1} && (2.58)\\
\dot{x}_{2_1} &= f_1(t, x_1, x_2, u) + \xi_1(t, x_1, x_2, u) && (2.59)\\
\dot{x}_{1_2} &= x_{2_2} && (2.60)\\
\dot{x}_{2_2} &= f_2(t, x_1, x_2, u) + \xi_2(t, x_1, x_2, u) && (2.61)\\
&\vdots && (2.62)\\
\dot{x}_{1_n} &= x_{2_n} && (2.63)\\
\dot{x}_{2_n} &= f_n(t, x_1, x_2, u) + \xi_n(t, x_1, x_2, u) && (2.64)
\end{aligned}$$

For each measured position variable x_{1_i}, the i-th observer subsystem may be expressed. From (2.50) and (2.51), the observer for the subsystem made up of the i-th position and velocity states is defined as,

$$\begin{aligned}
\dot{\hat{x}}_{1_i} &= \hat{x}_{2_i} + z_{1_i} && (2.65)\\
\dot{\hat{x}}_{2_i} &= f_i(t, x_1, x_2, u) + z_{2_i} && (2.66)\\
z_{1_i} &= \lambda_i |x_{1_i} - \hat{x}_{1_i}|^{1/2} \text{sign}(x_{1_i} - \hat{x}_{1_i}) && (2.67)\\
z_{2_i} &= \alpha_i \text{sign}(x_{1_i} - \hat{x}_{1_i}) && (2.68)
\end{aligned}$$

Defining the i-th subsystem estimation error as,

$$\begin{aligned}\tilde{x}_{1_i} &= x_{1_i} - \hat{x}_{1_i} \\ \tilde{x}_{2_i} &= x_{2_i} - \hat{x}_{2_i}\end{aligned}$$

leads to the following observer error dynamics,

$$\dot{\tilde{x}}_{1_i} = \tilde{x}_{2_i} - \lambda_i |\tilde{x}_{1_i}|^{1/2} \text{sign}(\tilde{x}_{1_i}) \tag{2.69}$$

$$\dot{\tilde{x}}_{2_i} = F_i(t, x_1, x_2, \hat{x}_2, u) - \alpha_i \text{sign}(\tilde{x}_{1_i}) \tag{2.70}$$

Given that the bound (2.55) is satisfied, the i-th subsystem observer for the n-DOF system fits into the form required in Theorem 2.2.2. The i-th subsystem observer error dynamics will converge in T_i seconds. Since this is true of all observers for each subsystem, the total observer convergence time is upper bounded by T, where

$$T = \max_i (T_i) \tag{2.71}$$

Therefore, after T seconds the state estimates $(\hat{x}_1, \hat{x}_2)$ will have converged to the true state (x_1, x_2) and the unknown input vector $\xi(t, x_1, x_2, u)$ will be available. In this case,

$$\xi_i(t, x_1, x_2, u) = \alpha_i \text{sign}(\tilde{x}_{1_i})_{eq} \tag{2.72}$$

From these results it is clear that the mechanical observer proposed in [31] may be applied to n-DOF nonlinear mechanical systems for both finite time state and unknown input estimates. □

2.3 State of the Art of Bilateral Teleoperation

When studying teleoperation systems in the presence of network time delays there are a number of issues to deal with. Of principal importance are issues involving stability and transparency. Stability is of fundamental importance because, without it, a teleoperation system cannot be operated in a safe fashion. Teleoperation systems that have some delay in the communications are especially interesting to consider. It has been shown [5] that even small time delays in the closed loop of a bilateral teleoperation system can destabilize it. Transparency provides a measure of performance for a teleoperator system. As a result, it is an important factor to consider in evaluating teleoperators.

In the sections that follow, a review of the state of the art in dealing with these issues in teleoperation systems will be performed.

2.3.1 Passivity Approaches

Research in teleoperation systems dealing with time delay goes back as far as 1965 [32]. However this early work does not deal with force reflection. In 1966, research involving force-reflecting bilateral teleoperation systems was presented [33]. In that work, the destabilizing effects of time delay were apparent. A number of approaches for stable bilateral teleoperation in the presence of time delays have been developed based on ensuring the passivity of the system

In 1989, Anderson and Spong [5] presented work that developed a stable control law for force reflecting teleoperation. This approach makes use of the concepts of passivity and scattering theory. Scattering theory defines the *scattering operator* [5] as;

Definition 2.3.1. *The scattering operator* $S : L_2^n(R_+) \mapsto L_2^n(R_+)$ *is defined by*

$$u - y = S(u + y) \tag{2.73}$$

and maps effort plus flow into effort minus flow, where the flow y *is entering the system's ports, and the effort* u *is measured across the system's ports [5].*

In a general sense, effort can be regarded as the input to the system, and flow the resulting output. For a robot manipulator, the effort would be the torque exerted at each joint and the flow would be the angular velocity at each joint.

The following definition of passivity, taken from [22], is useful here.

Definition 2.3.2. *An operator $G : L_{2e} \mapsto L_{2e}$ is said to be* passive *if*

$$\langle x, Gx \rangle_T \geq 0, \forall T \geq 0, \forall x \in L_{2e}$$

and is strictly passive *if there exists a constant $\epsilon > 0$ such that,*

$$\langle x, Gx \rangle_T \geq \epsilon ||x||_{T2}^2, \forall T \geq 0, \forall x \in L_{2e}$$

When working with passivity, the following theorems are of importance [22],

Theorem 2.3.1. *Consider a feedback system in the standard configuration with blocks G_1 and G_2. Suppose there exist constants ϵ_i, δ_i, $i = 1, 2$, such that*

$$\langle x, G_i x \rangle_T \geq \epsilon_i ||x||_{T2}^2 + \delta_i ||G_i x||_{T2}^2, \forall T \geq 0, \forall x \in L_{2e}, i = 1, 2.$$

Then the system is L_2 stable without bias if $\delta_1 + \epsilon_2 > 0$, $\delta_2 + \epsilon_1 > 0$.

Corollary 2.3.1. *The standard configuration feedback system with blocks G_1 and G_2 is L_2 stable without bias if both G_1 and G_2 are strictly passive.*

From these theorems (which are proved in [22]) one observes that the passivity of operators in a closed loop configuration is linked to stability. In particular, the feedback of two strictly passive systems in the standard configuration is a *sufficient* condition for L_2 stability. It is important to note that, while passivity can be used to show stability, the sufficiency of the condition implies that in some cases it could be a conservative result.

The approach in [5] makes use of the analogy between electrical and mechanical systems to represent a teleoperation system as a network of electrical components. In this representation, each block is represented as an n-port system. Each port in an n-port will have some effort applied to it (torque in the case of a manipulator) and some flow (angular velocity for a manipulator) travelling through it. Anderson and Spong represent the teleoperation system in five blocks; a 1-port for the human operator, 2-ports for the master manipulator, the communication channel, and the slave manipulator, and finally a 1-port for the environment that interacts with the slave manipulator. This network representation is shown in Figure 2.1.

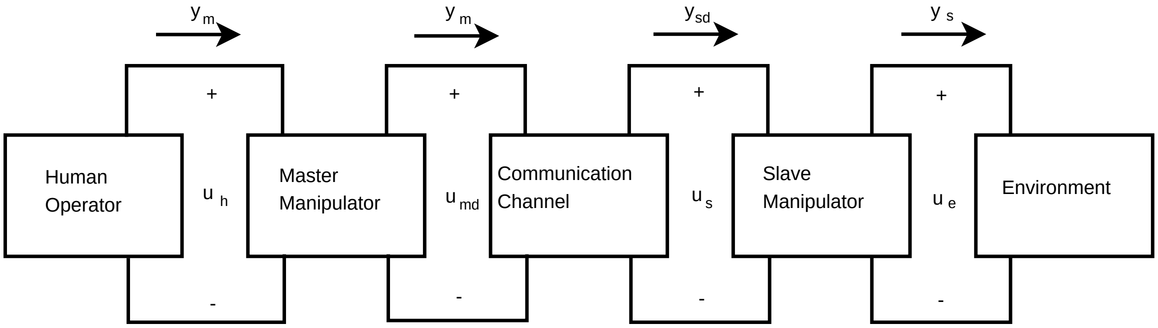


Figure 2.1: Representation of the teleoperator system using network ports.

Anderson and Spong evaluate the scattering matrix for the closed loop bilateral teleoperation system and show that the introduction of a time delay into the closed loop results in a scattering matrix norm of ∞. They also show that a system is passive if and only if the scattering matrix norm is less than or equal to one. Thus, the closed loop system is potentially unstable with a time delay. Control laws for both the master and slave that take time delay into account are then presented that result in a passive system, so the closed loop is stable. Simulation results are presented in [5] showing the performance of the proposed approach. While the system is stable, there remains significant position error in the tracking, especially during periods in which the slave manipulator is in contact with the environment.

Following the development of the passivity approach of Anderson and Spong [5], Niemeyer and Slotine developed the concept of wave variables in the analysis and design of teleoperation systems with time delays [6, 7, 8, 9]. Wave variables encode the force and velocity signals, denoted F and $\dot{x}$ respectively, using an algebraic transformation defined as follows;

$$u = \frac{b\dot{x} + F}{\sqrt{2b}}, \quad v = \frac{b\dot{x} - F}{\sqrt{2b}} \tag{2.74}$$

The variable u is termed the forward moving wave and carries signals from the master manipulator to the slave manipulator, while v is termed the backward moving wave and carries signals in the reverse direction. The scalar b is termed the characteristic wave impedance, which serves as a tuning parameter to trade off between speed of motion and level of forces [7]. As the wave impedance is increased, the velocity will become smaller while the force becomes larger. This causes the system to feel more damped. Conversely, decreasing the wave impedance makes movement easier and makes the system feel less damped.

The approaches just presented pass velocity information. Since no position information is explicitly passed, there can be a slow drift in the relative positions of the two manipulators. This could be due to errors in numerical integration, as well as data loss. As well, any initial offset in position between the two manipulators cannot be corrected [9]. In order to solve this problem, Niemeyer and Slotine [34] propose passing the integral of the wave variables as well as the wave variables themselves. These wave integrals encode position and momentum. Adding wave integrals to the communications does not effect passivity [34], but guarantees that position drift is no longer a problem.

Chopra *et al.* [35, 36] show, using the wave variables framework and assuming linear dynamics for the manipulators, that even in an ideal situation with both manipulators having the same initial position, the reference position for the slave to follow is a function of the time delay. Thus, as delay increases in the bilateral teleoperator, position drift between the master and the slave increases as well. They propose a new architecture that makes use of the wave variables but adds additional feedback paths where the master position is transferred directly to the slave, and vice versa. Additional proportional controllers are used on each side and it is shown that, for suitable choices of the controller gains, the position tracking error goes to zero at steady state for free motion. On contact with the environment, the force tracking error goes to zero at steady state [36]. Experimental results in the presence of a constant time delay and occasional packet loss are presented in [36].

Even in the presence of significant packet loss, the slave tracks the master at steady state. With the traditional wave variables architecture, position tracking degrades at steady state as packet losses increase [36].

In [37], Chopra *et al.* present a new bilateral teleoperation algorithm based on the wave variables approach that is designed to deal with varying time delays and data loss while still ensuring stability. The addition of Communication Management Modules (CMM) to the standard wave-based teleoperation system is used to preserve stability in cases where the network causes data loss and time varying transmission delays. The CMM is constructed such that, instead of transmitting the wave variables, some function of the wave variables that may be specified by the designer is transmitted. This portion of the CMM is termed the function generator (FG). A signal reconstructor (SR) is used at the remote side in order to reconstruct the signals transmitted by the FG. Chopra *et al.* prove [37] that satisfying the following conditions on the wave variables ensures stability of the master-slave system,

$$\int_0^t ||\hat{u}_s||^2 ds \leq \int_0^t ||u_m||^2 ds \tag{2.75}$$

$$\int_0^t ||\hat{v}_m||^2 ds \leq \int_0^t ||v_s||^2 ds \tag{2.76}$$

where u_m is the wave variable transmitted from master to slave, $\hat{u}_s$ is the output of the FG received by the slave, v_s is the wave variable transmitted from the slave to the master, and $\hat{v}_m$ is the output of the FG received by the master. Beyond satisfying these conditions, the choice of the FG and SR are up to the designer. One option that satisfies these choices is an identity function for the FG and time-varying gains for the SR. A discrete-time version of these conditions is then developed, since many communications networks are inherently discrete. Additionally, a technique for packet loss compensation is proposed. In order to achieve this, a packet formatter on the transmitting side is used. This packet formatter computes a running sum over time of the wave variable to be transmitted. As well, it encodes a unique index for each wave variable. At the receiving side, a packet reader extracts the summed wave variables and the packet index. Then, a subtractor block subtracts the last received values of the summed wave variable from the latest one.

Without packet loss, this difference will be equal to the transmitted wave variable. If this is not the case, an interpolator is used and the computed difference is split up into several samples using an interpolator in a way that preserves passivity [37]. This technique is verified experimentally and shown to be effective against packet loss and time varying delays.

Gu *et al.* [38] present an adaptive bilateral teleoperation algorithm that makes use of the wave variables transformation. This approach is developed for systems that experience time-varying delays through the communications. The authors note that the wave variables transformation as originally developed only preserves passivity for a constant time delay [38]. As a result, an adaptive approach based on the variation in time delay is proposed. By making use of the delay differentials, a modification to the wave variable equation is made such that it is integrated before being sent through the communications, and differentiated on the other side of the communications. This signal is then recombined with the original wave variable. It can be shown [38] that this operation preserves passivity even in the presence of varying time delays. While not an adaptive control algorithm in the classical sense, the final wave variable output is adjusted according to the variation in the time delay. It is in this sense that adaptation occurs. Simulation results using linear master and slave plants are presented to confirm the effectiveness of this algorithm in the presence of a varying time delay.

In [39], Aziminejad *et al.* perform passivity-based stability analyses for a variety of teleoperation architectures. Both admittance type position error based (a form of the position-position architecture) and hybrid position error based (which requires force sensing) are examined. As well, a new approach termed the admittance-type kinesthetic force-based (AKFB) algorithm is presented. This new approach is a two channel architecture that requires force sensors at both the master and slave sides. It was shown that, in the case of no time delays, the only non-ideal entry in the hybrid matrix is the entry $H_{22}(s)$. This is an improvement in terms of performance over the admittance position error based architecture. All of the approaches presented make use of the wave variables transformation in order to preserve passivity. An interesting conclusion of this paper is that, from the perspective of transparency, better performance is achieved when force sensors are used in the teleoperator system. The practical performance of each of the algorithms studied is examined through a series of experiments.

Haddadi and Hashtrudi-Zaad [40] present a new approach for stability analysis of wave-based bilateral teleoperation systems. This approach makes use of the concept of scattering parameters from microwave systems [40]. It is advantageous over the absolute stability approach based on Llewellyn's criteria in that it produces less conservative results. As well, this new approach yields bounds on the stability of a system that may be unstable when connected to a passive operator. This approach expands the opportunity to determine that a system is stable even if the system is not absolutely stable. A new concept termed bounded impedance absolute stability (BIAS) is introduced, allowing one to guarantee stability of a particular system provided that the operator and slave environment impedances lie within some bound. This approach is perhaps more rigorous than that of absolute stability in that it becomes possible to include the environment impedance in the stability analysis. However, the assumption that the environment is linear is retained when it is used in the analysis. Another advantage of this technique is that the stability results may be used in the design of the bilateral teleoperator in order to ensure that the system is robustly stable for the particular set of environment conditions that it will encounter [40]. The authors apply this analysis technique to both a four channel communications architecture and a two channel position-force architecture. The results of the analysis show that a wider range of possible environment impedances, over those allowed by Llewellyn's stability criteria, may be used while still ensuring stability of the system.

Delay Predictors

In [41] Niemeyer and Slotine develop a wave variable based method for dealing with systems that have variable time delays, such as the Internet. They show that, in the presence of time varying delays, the passivity of the system using wave variables is no longer conserved. To compensate for this, a reconstruction filter is used on the slave side of the system. This filter receives the delayed wave integral and wave energy signals from the master side, as well as local versions of the wave signal and wave energy, integrated on the slave side.

Defining the master wave variable as $u_m(t)$ and the received wave variable on the slave side as $\hat{u}_s(t)$, the integrals of these two signals are computed, and their difference is taken. This difference is a measure of the change in energy of the signal. By feeding that energy back into $\hat{u}_s(t)$, the lost energy is restored. So, the wave variable to be used on the slave side is computed as,

$$u_s(t) = \hat{u}_s(t) - \sigma\Delta(t) \tag{2.77}$$

In [42] the authors suggest that the choice of the gain σ must be made based on the trade-off between rapid convergence of the slave wave variable to the delayed master side variable and sensitivity to fluctuations in time delay. In order to balance this trade off, Mirfakhrai and Payandeh [43, 44] propose an approach to model the time varying delay in the communications. To model the delay, they propose the use of an autoregressive discrete-time model driven by white noise. The time delay data is then fit to this model and used to predict the future time delays. With the ability to predict future delay values, Mirfakhrai and Payandeh propose using the delay predictions to adjust, via a lookup table, the parameter σ in an optimal fashion to minimize the velocity and force errors.

Zhang and Li [45] propose a very similar delay prediction approach to tuning the feedback gain σ in (2.77). However, instead of using a discrete-time autoregressive model to estimate the time delay, they model the process dynamics of the delay as $y(t) = x(t-\tau(t))$, where $\tau(t)$ is the time-varying delay. The variable $\theta(t)$ is used as the estimate of the time delay, and the estimation error is determined as $e(t) = y(t) - x(t - \theta(t))$. The time delay is estimated online using a steepest descent algorithm. With an estimate of the time delay, heuristics are given to choose σ. If the time delay is large, σ should become more negative [45]. This gain is adjusted based on empirical results, and no formal method for choosing it is presented.

Predictive Approaches Using Wave Variables

Liu *et al.* [46] propose a pose reflecting teleoperation scheme that makes use of wave variables and incorporates wave prediction. This scheme differs from most bilateral teleoperation systems considered here in that, instead of reflecting force back from the slave to the master, the slave pose is reflected back. In this setup, the velocity and pose are encoded in the wave variables. This work assumes linear models for the dynamics. At the slave side, the dynamics of the slave as well as the wave transformations are treated as one plant, called $G(s)$. Effectively, this plant has wave variables as its input and output. A discrete-time state space representation of $G(s)$ is determined and it is used in a linear discrete-time implementation of the Kalman filter. The Kalman filter is placed on the master side. Using the Kalman filter update equations, the wave variable that would be received from the slave side is predicted n steps into the future, depending on the value of the time delay. The master side then uses this predicted estimate of the wave variable in obtaining an estimate of the slave pose. Simulation results are presented showing the effectiveness of this approach in the presence of a varying time delay, but no formal proof of stability is presented.

Ganjefar *et al.* [47] propose an extension to the wave variables approach that also involves predicting future values of the wave variables. A Smith predictor, based on the slave plant dynamics, is used on the master side to predict the value of the wave variable coming from the slave T seconds into the future, where T is the time delay in the system. Simulation results, using linear dynamics, are presented to show the performance of the approach. The addition of a Smith predictor has a positive effect on tracking performance. However, no formal analysis of the stability of this system is presented.

Munir and Book [48] propose an alternate approach for wave based prediction. In this setup, a modified Smith predictor, an energy regulator, and a Kalman filter are used to predict the future values of the wave variables. The prediction happens entirely in the wave domain, on the master side of the system. The entire slave side of the system, which includes the slave plant, PD controller, and the wave transformation, is given by the combined transfer function $G_R(s)$. Developing a state-space model of $G_R(s)$, a Kalman filter is then used to estimate the state of the entire slave side. This state estimate is delayed in time, since the returning wave that the Kalman filter uses as the measurement passes through the delay in the communications.

The delayed slave side state estimate is passed to a time forward observer. The time forward observer generates a predicted state $x_p(t)$, which is predicted T_T units of time ahead of the delayed state to compensate for the time delay.

This predicted state can then be used to generate a prediction of the slave wave variable, which is passed through an energy regulator. This regulator is designed to ensure that the passivity condition is met by the predicted wave. That is, the predictor must not increase the total energy in the system. Munir and Book then present a slight modification to this prediction scheme in order to ensure stability in the presence of time-varying delays.

Such prediction approaches are dependent on having an accurate model of the slave and environment [49]. However, when a slave manipulator moves from free space to environmental contact, or to a situation in which the manipulator is under load, the dynamics of the slave side change. To deal with this, Ching and Book [49] propose two approaches. The first makes use of an adaptive switching algorithm whereby, depending on the slave position error, different models of the slave are used to reflect the state of the slave and environment. A second approach is suggested that makes use of a recursive least squares (RLS) algorithm that is used to estimate the environmental force acting on the slave in an online fashion. The RLS algorithm is constantly updating the environmental force term in the predictor model on the master side.

Ching and Book state that the switching scheme is best in situations where the slave will be in rigid contact, since the RLS approach is unable to predict when the slave will leave the contact. The RLS algorithm is better suited to environments that vary slowly, due to the transmission delay and the learning curve of the RLS algorithm [49]. Stability of both of these approaches is guaranteed due to the use of the energy regulator, which is the same as that proposed in [48]. A compensator for position drift, which is an issue due to non-constant time delay and slave modeling error, is also presented in this work and shown to be effective in the simulations.

In situations where the dynamics may change suddenly, such as due to sudden changes in the environment, it may be beneficial to have a method to rapidly adapt to the change. Kamrani *et al.* [50] propose the use of multiple model adaptive controllers along with a wave variable approach and a Smith predictor to perform wave prediction. As well, their approach allows for possibly varying time delays, and models the delay with an ARX model. In a multiple model adaptive controller, a supervisor is constantly evaluating cost functions for each of the possible models. When some change in the system occurs, the supervisor determines the best model for the current situation and switches to the corresponding controller [50]. This allows for rapid changes to the control when the system changes rapidly. According to [50], the closed loop system with such a control strategy is BIBO stable. Simulation results show the effectiveness of the approach over a conventional wave prediction algorithm.

Non-Wave Based Passive Control

While it is common in the literature to see the use of wave variables when dealing with passivity approaches for bilateral teleoperation, other approaches exist as well.

In [11], Lawrence develops tools for the analysis of a general teleoperator architecture in terms of stability and performance. The teleoperator system used for the analysis is general enough to include the position-position architecture, the position-force architecture, and a transparency-optimized architecture, depending on how certain controllers and local feedback loops are chosen. The transparency-optimized architecture is a four channel setup, while the other two approaches fall into the two channel setup. Lawrence shows that a sufficient condition to achieve ideal transparency is the use of four channels. The analysis performed in this work shows that a trade-off must be made between transparency and stability. That is, an increased transparency setup provides a lower stability margin than a system that provides less transparency. The stability analysis of the teleoperator architectures is performed using passivity arguments that lead to a sufficient stability result. A detailed analysis of the transparency properties of each of the three specific architectures is performed as well, showing the relative performance of each approach.

Lee and Spong [51] propose a novel passivity-based control framework for nonlinear robot manipulators in the presence of a constant time delay. One advantage of this framework is the use of explicit position feedback. This feedback is useful in eliminating the potential for position drift found in other passivity-based algorithms [5, 8]. The authors propose to render passive the combination of the controller and communication blocks. The control objectives of this approach are master-slave position coordination when the manipulators are operating in free space, and static force reflection when the slave encounters the environment.

To accomplish this control objective, a very straightforward proportional-derivative controller with an additional dissipation term is used. The controller on the master side is given by,

$$T_m(t) = -K_v(\dot{q}_m(t) - \dot{q}_s(t-\tau_2)) - (K_d + P_\epsilon)\dot{q}_m(t) - K_p(q_m(t) - q_s(t-\tau_2)) \tag{2.78}$$

where τ_2 represents the delay from slave to master, $q_m(t)$ and $q_s(t)$ are the master and slave joint positions, K_v and K_p are the PD controller gains, K_d is a dissipation gain to ensure that the delayed P-control action is made passive, and P_ϵ is an additional damping term to ensure master-slave coordination. The control law used on the slave side is the same in form, but receives the delayed signals from the master side. Lee and Spong prove that with these controllers the closed loop is stable, in the absence of external forces the slave will exactly track the master, and when the slave is immobilized against an environment, the force applied at the master is equal to the force applied to the slave. Detailed proofs are given in [51]. This control framework is largely concerned with low-frequency performance, so high-frequency characteristics (i.e. detecting a sharp edge in the environment) have not yet been examined [51]. Both simulation and experimental results are given in [51] to show the effectiveness of the approach.

McIntyre *et al.* [52] propose a passivity based bilateral teleoperator control framework that also deals with nonlinear robot manipulators. In [52], two different control approaches are presented. The first approach allows for uncertainty in the master and slave dynamic models provided that the user and environmental input forces are measurable. The second approach requires that the master and slave dynamics be known, but allows for unmeasurable user and input forces [52]. The stability of each control approach is shown using a Lyapunov approach. However, this work does not deal with time delays and is less relevant to the problem being examined in this research proposal.

Ni and Wang [14] propose a gain switching control scheme for the bilateral teleoperation control problem. Their approach is based on the position-position architecture. In particular, the authors make use of position error feedback from the slave to the master. They mention that position error feedback can be used, instead of measuring the contact forces, to provide the master side with a sense of the force experienced at the slave. However, this approach provides poor transparency. In order to compensate for this, Ni and Wang show that in free motion situations, the master controller should have small gains while the slave controller should have large gains [14]. The opposite is the case when the slave is in contact with an environment. To deal with this, a gain switching strategy is employed. Another issue examined in [14] is that of contact transition stability. Despite the fact that a controller can be designed that ensures a stable closed loop in both free motion and contact, the transition from one mode to the other can lead to instability. The control approach presented in this work ensures asymptotic stability of the entire system while employing a gain switching controller for free motion, contact transitions, and constrained motion. Lyapunov arguments are used to prove stability. Note that time delay is not considered in this paper.

According to Ryu *et al.* [53], using a passivity approach in designing controllers for teleoperation systems leads to overly conservative controller designs. The reason for this is due to the analysis of system passivity in the frequency domain, which results in a fixed damping value regardless of operating conditions. In [53], the authors develop the concept of a passivity observer (PO) and a passivity controller (PC). The PO is always monitoring the energy through the controller/communications block to see whether it is storing/dissipating energy or generating energy. With an idea of the system energy at any given time instant, the PC can be designed to only dissipate the required amount of energy [53]. Such a setup with a PO and a PC can be implemented in the context of a standard position-force bilateral control setup. The paper [53] provides experimental results to show the effectiveness of the PO/PC when the slave comes into hard contact with its environment. This paper does not deal with teleoperation systems involving time delay, except to mention that future work must be put into that problem.

Naerum and Hannaford [54] present a global transparency analysis of an extended form of the general teleoperation architecture developed by Lawrence [11]. They examine the extension to this architecture that was proposed by Hashtrudi-Zaad and Salcudean. This extended architecture, referred to as the Extended Lawrence Architecture (ELA), differs from Lawrence's original architecture in that local force controllers are included at both the master and slave sides. Of particular interest in this work is a necessary and sufficient condition for transparency using the ELA. In particular, it is proved that a two channel architecture is a necessary and sufficient condition for achieving ideal transparency. However, not every two channel configuration yields transparency. For instance, both the position-position and position-force architectures will not yield ideal transparency. In order to achieve this in a two channel setup, one robot must be velocity controlled, while the other must be force controlled. Prior to this result, the common belief was that it is not possible to achieve ideal transparency with only two channels [54]. Naerum and Hannaford show that local feedback at each site with only two communication channels is equivalent to a setup using four communication channels with no local feedback loops. Note that this work deals only with teleoperation systems that do not contain delays in the communications.

In [55], Seo *et al.* present a stable bilateral teleoperation algorithm where passivity is ensured in the presence of time-varying delays using an energy bounding algorithm (EBA). This algorithm is designed in order to ensure that passivity is guaranteed despite the presence of time delays. The signals transmitted from the master to the slave, and vice-versa, are each passed through an EBA. These EBAs restrict the amount of energy that passes from one side to the other in such a way that passivity is always preserved. This approach was originally developed for haptic simulation of virtual environments [55], and as such the EBA on each side views the combination of the communications, remote plant and controller, sample and hold blocks, and environment/operator in an analogous way to a virtual environment. The EBA passifies this whole remote system in order to maintain stability. A PD controller is used at the slave side to ensure tracking of the delayed master trajectories. The output of this controller is used as the remote force that is transmitted back to the master side. This algorithm, then, is a position-error based type of position-position algorithm. Experimental results for this algorithm are presented for time-varying delays with nominal values up to 2.5 seconds in each direction and for cases involving both free and constrained motion at the slave side. The experimental results show stable behaviour in all cases with the EBA, but position error increases for large variable time delays.

2.3.2 Stable Teleoperation Without Relying on Passivity

While passivity is a popular tool to use in developing stable bilateral teleoperators, there is motivation for using approaches that do not depend on passivity. Lawn and Hannaford [56] performed a number of experiments on both passive and non-passive bilateral teleoperation schemes and concluded that the passive approaches resulted in a roughly 50% increase in task completion time. They conclude that the reason for this is that the guaranteed stability of the passivity approaches comes at the cost of reduced stiffness [56]. As a result, a number of approaches not based on passivity can be found in the literature.

In [57], Kim *et al.* present a case study of human-human interaction using haptic devices over the Internet. One side of the experiment took place in the U.K., while the other took place in the U.S.A. The high speed research network Internet2 was used to connect the sites. The study examines the effect of haptic feedback on the ability of two humans in geographically separate areas to complete a task together. In conducting the experiments, separate groups of people were tested. In one group, haptic information was available for feedback, while in the other it was not. Since human operators are placed at both the master and slave sides of the system, explicit controllers are not implemented. However, due to the network time delay and force feedback, additional damping is injected into the system at various points to maintain stability [57]. A collision prediction algorithm is employed, whereby the predicted motion of the remote hand is used to determine when the remote side will contact the shared environment. The results of the experiments showed that the addition of haptic feedback increased the sense of co-presence for the users. However, no proof of stability for the closed loop system is given.

Polushin *et al.* have recently proposed an approach to bilateral teleoperation over a network with unknown and varying time delays such as the Internet [58, 59, 60]. The proposed control scheme guarantees stability that is independent of the time delay. As well, if the assumption that the delay is "approximately smooth" holds then the control scheme will guarantee that the slave manipulator is able to track the master manipulator's *delayed* trajectory.

In order to deal with the unknown time delay and potential for data loss, a "dirty derivative" filter is introduced. This filter is effectively an observer on the slave side that has two uses. If data is lost in the communications from master to slave, it is possible that the slave will receive a discontinuous reference trajectory from the master. The filter will generate a smooth approximation of the reference trajectory for the slave to use. As well, this filter generates velocity and acceleration estimates of the delayed master state.

This approach does not make use of passivity methods to guarantee stability. Instead, a modified version of the input-to-state small-gain theorem for functional differential equations is used [58]. Using certain control laws, it can be shown [58] that, under certain assumptions about the time delays, input to state stability can be guaranteed as well as arbitrarily precise tracking of the delayed master manipulator signal by the slave manipulator. Simulation results [58] and experimental results [60] for the proposed approach show its effectiveness. Even in the presence of significant time delays (~ 2 seconds), the slave continues to track the delayed master position.

Alfi and Farrokhi [61] propose a stable bilateral control scheme that makes use of classical linear control techniques to stabilize the closed loop. They assume a standard bilateral teleoperator setup with a possibly time varying communication delay. Classical PD controllers are designed for both the master and the slave manipulators. Since the time delay can have a negative effect on system stability, a Smith predictor is used on the master side in order to compensate for this. Alfi and Farrokhi note in [61] that the use of a Smith predictor requires a constant time delay. In order to extend the use of the Smith predictor to time-varying delays, an FIR filter is used to estimate the time delay online using an LMS algorithm. The estimated time delay is then used in the Smith predictor in order to compensate for the delay. The stability of the closed loop is shown, assuming the magnitude of the closed loop frequency response is less than one for all frequencies. As well, an assumption on the time delay is made, in that the time delay must not be greater than 1 msec. Simulation results are given based on linear dynamics and small time delays.

While the results show the effectiveness of the approach, it is limited by its requirement for a very small time delay.

In [62], Pilieci reports on the use of haptic devices communicating across the Internet between Ottawa and Geneva. In this setup, humans are controlling each side of the teleoperator system. The setup allowed people to shake hands transatlantically, as well as the ability to feel a pulse in a human located at the other end of the teleoperator. No technical details are given in [62].

Nuno *et al.* [63] present new bilateral teleoperation algorithms for nonlinear manipulators in the presence of constant time delays. Two separate approaches are proposed. In both cases, the fairly common assumption that both the human operator and the slave environment are passive is made. The first approach is based on the position-position architecture. Both the master and slave track each other's position. In this setup, simple controllers are used at both the master and slave sites. Each controller has a proportional term that contains the tracking error and an additional damping term that is proportional to the plant's velocity. They then show [63] that, for suitable choices of the controller position and damping gains, the position error and velocity error of the teleoperator are bounded. Under the additional assumptions that the human operator is motionless, the slave is in free motion, and that a gravity compensation term is added to each controller, they prove that position coordination is achieved. That is, one manipulator asymptotically tracks the other. The second algorithm is a position-force algorithm. That is, the slave tracks the delayed master trajectory and the force exerted on the slave by the environment is reflected back to the master side. The slave controller in this approach is the same as the controller used in the position-position algorithm. The master controller contains the reflected slave torque and an additional damping term that is proportional to the master velocity. With this setup, the same results are proved for this algorithm as for the position-position algorithm. Simulation and experimental results are presented in [63] that verify the effectiveness of the approaches. Constant time delays of 0.7 seconds from master to slave and 0.9 seconds from slave to master are used in all cases. The simulations and experiments confirm the validity of the approach.

In [64], Nuno and Basanez present initial results toward the task of developing a general framework for performing stability analysis of teleoperator systems with time-varying delays. A general Lyapunov-like function is developed that, under slight modifications, may be used to analyze the stability of a variety of bilateral teleoperation architectures. The master and slave plants are both assumed to be nonlinear robot manipulators with gravity compensating controllers. The authors point out that this Lyapunov-like function is not actually a Lyapunov function, because it may not be possible to show that it is non-positive. However, stability of the system using this function is shown through integration of the Lyapunov-like function, as in [65]. The proposed Lyapunov-like function contains three terms. The first term represents the kinetic energy of both manipulators. The second term represents the interaction of the operator with the master manipulator and the environment with the slave manipulator. Note that the assumption that both the human operator and slave environment are passive is made throughout [64]. The third term in the function represents the energy in the communications between the master and slave sites. This term must be designed on a case by case basis depending on the teleoperation control strategy. The authors then present the use of this general Lyapunov-like function to show stability of a variety of teleoperation algorithms. The stability analysis is able to show boundedness of the position tracking error and velocity signals. Under the additional assumption that the human does not move and the slave is in free motion, asymptotic position tracking may be shown.

Sliding Mode Control

Sliding mode control is a popular control strategy when dealing with parametric uncertainties in the plant model [16]. Park and Cho [66] present a sliding mode control approach for a bilateral teleoperation system in the presence of varying time delays. This approach assumes linear dynamics for both the master and the slave, and uses an impedance controller on the master side. The sliding mode controller is implemented on the slave. Park and Cho make mention of the sliding mode controller's robustness to model parameter variations in helping to compensate for the effects of the time delay [66]. The gain term on the switching control law used in standard sliding mode control must be increased as the time delay increases, and as this gain is increased, the controller is more likely to cause chattering [66]. In order to deal with this, Park and Cho propose a modified sliding mode controller whose gain on the switching term can be designed independent of delay. Simulation results show the effectiveness of this modified sliding mode controller. From the simulations, the slave appears to track the delayed master position well. As well, the forces exerted on the slave are reflected back to the master quite accurately. An extension of these results, presented by Cho and Park [67], propose a similar algorithm, but include the slave environment force as a driving term in the slave desired impedance model. This approach allows for a reduced impact force when contacting the environment. In both cases, stability is proved under the assumption of a passive operator and a passive environment. Llewellyn's stability criteria is used to give necessary and sufficient conditions for absolute stability of the system [67].

Conventional sliding mode control is known to introduce chattering into a system, due to delays and the fact that actuators are not able to switch infinitely fast [16]. Second order sliding mode control is an approach developed that solves the problem of chattering [68]. In this approach, no discontinuous terms appear in the control law. Garcia-Valdovinos *et al.* [68] propose the use of a second order sliding mode controller for the slave in a bilateral teleoperation system with a constant but unknown time delay. As well, they make use of simple linear observers to estimate velocity and acceleration signals, in order to avoid measuring them. Linear models of the master and slave dynamics are used in the analysis and controller design. The master manipulator is controlled with a standard impedance controller. A proof of stability is given showing that the closed loop system, including the observers, is asymptotically stable. From the experimental results, it is evident that the slave tracks the delayed master position during free motion, and the force experienced by the slave is reflected back to the master during slave contact with the environment.

In many situations, controllers are implemented in a discrete-time computer. As a result, there is merit in developing controllers directly in discrete time. Khan *et al.*[69] present a bilateral teleoperation algorithm that makes use of a discrete-time sliding mode controller at the slave side. This algorithm is applied to a teleoperation system with a piezo actuator at the slave side, designed for micro manipulation tasks. Time delays are not considered in this work. In order to deal with modeling error at the slave side, the addition of a disturbance observer is made. This disturbance observer produces an estimate of all of the unmodeled terms acting on the plant by back-computing it from estimates of the position, velocity, and acceleration generated from a standard discrete-time sliding mode observer. Experimental results are presented to validate the algorithm. However, no stability analysis is performed.

Adaptive Control

Adaptive control schemes are useful in any situation where the parameters of a dynamic system are either unknown or time-varying. In many cases, the accuracy of a closed loop system in terms of meeting a tracking objective is dependent on how well the parameters of the plant are known. An adaptive controller may be used to learn the parameters of the plant online in order to improve closed loop tracking. This is equally true of bilateral teleoperation systems, when the manipulator parameters may not be perfectly known. As a result, adaptive control has been used in the literature to improve teleoperation system performance.

Lee and Chung [12] propose an adaptive controller for a bilateral teleoperator. The adaptation law exists on the slave side and is used to estimate the parameters of both the slave manipulator and the environment it is in contact with. All of the dynamics in the system are assumed to be linear, and an adaptation law is derived using a Lyapunov stability approach [12]. The closed loop is shown to be stable, and the simulation and experimental results show excellent performance. However, this algorithm does not take time delays into account.

Hashtrudi-Zaad and Salcudean [70] propose an adaptive controller for a bilateral teleoperator under time delay in which the adaptive scheme is used to estimate the slave side parameters. Linear dynamic models are used for the master, slave, and environment. A composite adaptive controller is used, which aims to drive both the tracking error and parameter estimation error to zero. Through different choices of the adaptation gain, controllers with different uses can be obtained. For instance, a fixed adaptation gain is a good choice in situations where the slave will be in intermittent contact with the environment, or when the environment is particularly hard. The controller at the master side is not adaptive, and is designed to achieve transparency in that the transmitted impedance will be equal to the environmental impedance at the slave side [70]. Simulation results show that, in the presence of time delays, the proposed control scheme performs well in terms of achieving transparency [70]. No proof of stability is given for this algorithm.

In [71], Fite *et al.* propose an adaptive bilateral teleoperation control scheme based on linear dynamic models, and in the presence of constant time delays. A classical control approach is taken in designing the controllers. A lead-lag compensator is used to perform frequency domain manipulation of the transparency and stability robustness properties of the closed loop [71]. In order to improve the stability margin in the presence of time delay, a Smith predictor is added to the system. The Smith predictor includes the slave environment model, so an adaptation law is proposed to learn the parameters of the environment. The parameter estimates are used to adapt the Smith predictor to ensure proper time delay compensation and improved transparency [71]. Experimental results show that the use of an adaptive predictor allowed for a desired transparency bandwidth even in the presence of variations in the environment. No tracking results were given.

Polushin *et al.* [72] propose an adaptive bilateral teleoperation control algorithm based on nonlinear dynamic models and the presence of a constant time delay through the communications. The adaptation law is based on the fact that nonlinear robot dynamics may be expressed as linear in the parameters. A standard adaptation law is presented to estimate the unknown inertial parameters of both the master and slave robots. Nonlinear control laws are proposed for each of the master and slave manipulators, based on the estimates of the inertial parameters. The scheme requires only position measurements, as a “dirty derivative” filter is used to estimate velocity and acceleration signals from the position signals [72]. Stability of the approach is shown through the use of a modified input-to-output stability small gain theorem for functional differential equations [72]. The simulation results presented show the stability of the proposed approach when entering into contact with the slave environment. As noted in [72], the quality of the position tracking could benefit from improvement.

In [73], Shahdi and Sirouspour propose a combined robust and adaptive control algorithm for bilateral teleoperation with constant time delays. Both the master and slave manipulators are modeled by nonlinear dynamics. The human and slave environment dynamics are modeled as second order decoupled linear systems in the robot task space. Adaptation laws are used at both the master and slave sides to learn the unknown robot parameters. Control design is performed in the robot task space, with feedback linearizing controllers that make use of the inertial parameter estimates from the adaptation laws used in order to render each of the master and slave linear in the closed loop. Since the robot dynamics have been rendered linear by their respective controllers, the problem of tracking and force reflection is addressed using the H_∞ framework. The goal of this teleoperation scheme is to have the slave track the delayed master trajectory, and have the master experience the delayed reflected environment force from the slave. The H_∞ controller is placed at the master side, and the entire teleoperator system is modeled as one large time delay system. Design of the H_∞ controller is based on this model of the system as a whole. Experimental results are presented for cases involving 200 ms and 300 ms of time delay in each direction. The results show that this approach works fairly well, although the plots suggest that as time delay increases, tracking error begins to degrade.

Predictive Control Strategies

Due to the time delays inherent in many teleoperation setups, it is desirable to attempt to include some form of prediction in the control scheme. A properly implemented predictor would help to mitigate the effects of the time delay from the perspective of the operator. Early work in predictive teleoperation focused mainly on the use of predictive displays [74, 75]. Typically, such systems did not include force feedback from the slave to the master.

In 1998, Brady and Tarn [76] presented a teleoperation scheme for use over the Internet that makes use of a time-forward observer at the master side to predict the state of the remote manipulator. The slave's position information is passed back through the delay to the master side, and it is used as the partial state measurement to drive the observer. No force feedback is included in this setup. The estimated slave state is then used to update a predictive display of the slave environment. An event based planner is included at the slave side to handle execution of a trajectory submitted to it from the teleoperator. Force feedback is not implemented, and no proof of stability is offered.

Prokopiou *et al.* [77] propose a predictive approach to bilateral teleoperation control that involves prediction of the master state. This is as opposed to predicting the slave state and presenting the prediction to the operator. The notion is to have the slave respond to the forward-time prediction of the master motion. This effectively leads to the slave operating ahead in time from the master [77]. The approach assumes equal and constant delays in both the forward and backward direction, but the authors state that when this is not the case additional buffering can be used to ensure equal and constant delays in both directions. An advantage of predicting the master state at the slave side is that video from the slave side may be passed back to the master without any additional prediction applied to the video sequence. Two similar approaches are proposed. The first is a model-based predictor, which makes use of both the human arm and master robot dynamics for prediction. A second approach is a straightforward trajectory extrapolation, based on interpolating the master position and force, and then using the simple model to extrapolate future values of the master position and force. Linear dynamic models are used. The choice of controllers results in perfect force tracking, and asymptotically convergent position tracking. Simulation results for both the trajectory extrapolation and model-based approaches are given in [77], showing the effectiveness of the approaches. Certain force profiles, that were easy to extrapolate, resulted in better performance than others.

When developing an accurate predictive approach that involves prediction of the slave state, it is important to have knowledge of the slave environment dynamics. Such dynamics can be nonlinear and difficult to model accurately using a linear model. In order to deal with this, Smith and Hashtrudi-Zaad [78, 79] propose the use of an online neural network (NN) based estimator for the slave manipulator and dynamics. The NN is trained with the delayed master position as an input, and the environmental force acting on the slave as the output. In order to use the NN as a predictor, an NN with an identical structure is implemented on the master side. Once the initial learning period is finished, changes to the NN weights will be minimal [79].

The NN on the master side takes the *non-delayed* master position as input and generates an estimate of the slave environment force. This NN is then used in a Smith predictor style architecture to mitigate the effect of the time delay. Experimental results show that the proposed algorithm performs in a superior way to several benchmark algorithms it is compared to. In the experimental results, there is still a delay between master and slave positions, and master and slave forces. As well, no proof of stability is given for this control scheme.

Model predictive control (MPC) is an attractive control technique when dealing with the control of systems subject to input and state constraints. In MPC, a cost function is minimized to generate the optimal control input sequence given the constraints. Several researchers have applied MPC techniques to the bilateral teleoperation control problem.

Sheng and Spong [80] propose a slightly modified MPC algorithm for the control of a bilateral teleoperator in the presence of uncertain time delay. Controllers for both free motion and constrained motion are designed. In the case of constrained motion, a quadratic programming problem must be solved to generate the optimal control inputs [80]. No formal stability analysis is presented in [80].

Sirouspour and Shahdi [81, 82] propose linear quadratic Gaussian (LQG) controllers for the control of a bilateral teleoperation system under constant time delay. In [81] the development is done in discrete time. Separate controllers are designed for the cases of free motion and rigid contact. A switching strategy is employed to choose the appropriate controller depending on whether the slave is in free motion or constrained. In this control scheme, the controller resides entirely on the master side. Slave control signals are generated at the master and passed through the communications to the slave. The Kalman filter component of the LQG controller provides prediction of master and slave motion. The control signals generated attempt to produce non-delayed tracking [81]. A stability analysis based on the Nyquist theorem is given. Simulation and experimental results show the effectiveness of the method in stabilizing the closed loop through a time delay and in providing good position and force tracking.

In [82] Sirouspour and Shahdi propose a modification of the work in [81] where the problem is solved, again using an LQG controller, in continuous time. A transformation is applied to the teleoperator system to bring it from an infinite dimensional system with time delays in the control action to a system without delay [82]. The rest of the development is parallel to the development in [81], but is done in continuous time. A stability analysis is performed using the Nyquist theorem, and experimental results are given that show good performance. This approach has the benefit over [81] that the state vector does not increase in size with increasing delay.

Ni and Wang [83] propose the use of a fuzzy logic approach to tuning the controller in a bilateral teleoperation system. This work is focused on a human-to-human system. That is, there are humans operating both the master and slave manipulators. As well, position error from the slave to master is used to provide the force feedback component instead of actually measuring the environmental force at the slave side. In such systems, one desires different sets of gains depending on the operating mode. When one manipulator is operating in free motion, it is desirable to have low gains at the side operated by the human and high gains at the remote site to ensure good position tracking . However, when both manipulators are operated by humans, high gains are desired at both ends. In order to intelligently select the appropriate gains depending on the situation, a set of fuzzy rules is used to adjust the gains. By examining the power variable (the inner product of the control input and velocity output of the plant), and by estimating the total inertia of each manipulator in an online fashion, a set of fuzzy rules is developed to provide suitable PD controller gains. Experimental results show, from a qualitative point of view, an increase in the performance of the teleoperator system when tuning the PD controller gains using the fuzzy logic rules.

Munir [84] proposes a predictive bilateral teleoperation algorithm based on the use of a state estimator and a time forward observer. The development is performed using state space models of the master and slave dynamics. Initially, the state models have delays in both the control inputs and the state variables. The system is reduced to one having no delays in the control inputs and state variables. Either a Luenberger observer or a Kalman filter may be used to estimate the state of the slave side, based on the force feedback measurement. The estimated state is marched forward in time by the time forward observer. From the predicted state, the predicted force feedback term is obtained.

The master side uses this in the force feedback. Algorithms are presented for constant and varying time delays. No proof of stability is given, but simulation results show very good performance. However, Munir notes [84] that even small errors in the model for this approach can render the closed loop unstable.

In order to produce a control scheme that provides good transparency in the presence of time delay, Pan *et al.* [85] propose an approach based on observers at both the master and slave sides of the teleoperation system. Their focus is to find a less conservative control design than other approaches while preserving closed loop stability. It is noted that such prediction based approaches require a good knowledge of the dynamics of both the master and slave systems [85]. This approach assumes a time varying, but bounded, delay. Linear dynamic models for master and slave are assumed in the control design, but a nonlinear slave environment model is used. The prediction scheme makes use of two predictors *each* at the master and slave sides. Due to a coupling between certain states in the combined master-slave system, predictors for both the master and slave must be implemented at each side [85]. The predictors require an estimate of the time delay in order to artificially delay some of the signals on each side of the teleoperator. These delayed signals serve as inputs to the predictors, which generate estimates of the signals at the current time. Only a position measurement is required in the predictors, as full order observers are also implemented at both master and slave sides to estimate the velocity from the position measurements. A proof of stability is given for the closed loop system, showing that the tracking error is uniformly bounded and tends to a ball of a given radius [85] in finite time.

Yoshida *et al.* [86] present a predictive bilateral teleoperation algorithm using state predictors for cases involving time-varying delays in the communications. This algorithm aims to achieve trajectory synchronization from the master to the slave in the absence of delays, and accurate static force reflection. That is, when neither manipulator is moving, the force acting on the slave is accurately transmitted to the master such that those two forces are equal. In order to learn the robot inertial parameters, an adaptation law is used at each of the master and slave sides along with feedback linearizing controllers. Predictors are placed at both the master and slave sides to compensate for the time delays. These predictors are based on the solution equations for the closed loop master and slave manipulators. The solution equations make it possible to obtain an estimate of the remote signals delayed through the communications. In the case where the human and environment forces are constant, it is possible to show that the error on the predicted signals reaches zero in finite time [86]. A stability analysis is performed showing that the position and velocity errors in the system are bounded under this control strategy. Simulation and experimental results are given to verify the approach. Time delays of 0.5 seconds in each direction are used in all cases. From the results of the simulations and experiments, it is apparent that the predictors help to minimize position tracking error, but do not completely remove the effects of the time delays.

2.3.3 Summary of the Literature

Many different approaches to teleoperation have been examined in this section. Approaches involving time delays, as well as approaches involving delay-free teleoperation have been discussed. Initially, passivity-based approaches were presented. Early approaches based on passivity involved modeling the teleoperation system as an n-port network and designing control laws to ensure that the closed loop remains passive in the presence of time delays were discussed. Shortly after the introduction of these control approaches, the wave variables concept was developed. The wave variables provide a transformation of the velocity and force signals that ensures that the closed loop remains passive. Within the realm of the wave variables approach, predictive methods have been developed. Approaches to model the time delays in the case of varying time delays were presented. These approaches make use of reconstruction filters in order to ensure that the system remains passive even in the presence of varying time delays. Other predictive approaches have been developed to compensate for time delays in the closed loop. These approaches involve prediction of the wave variables that are transmitted across the communications. Smith predictors and Kalman filters have been used to perform the wave variable prediction. Other passivity-based teleoperation methods were presented that do not rely on the wave variable transformation.

A number of teleoperation algorithms that do not rely on passivity to ensure stability were discussed. Some of these results rely on small gain arguments to guarantee stability of the closed loop. Algorithms involving Lyapunov-like functions to ensure stability were also discussed. Following that, approaches based on sliding mode control were presented. Both standard first order sliding mode control and second order sliding mode control algorithms were discussed. Adaptive control techniques are also used in bilateral teleoperation. These adaptive algorithms typically involve standard adaptation laws to estimate plant parameters. Approaches that learn the slave side environment dynamics were also presented. Finally, predictive approaches not based on passivity were discussed. These approaches often rely on time forward observers or other model-based estimators to predict the state of the other side of the teleoperation system. The goal of these predictors is to compensate for the time delays by predicting ahead in time. An approach involving the use of a neural network at the master side to predict the state of the slave system was also given. Model predictive approaches were also presented. These approaches are particularly well-suited to predictive control, as the goal of model predictive control is to predict the state of the system ahead in time.

Having provided a background of the theory necessary to understand the material to be presented, and given a review of the state of the art in bilateral teleoperation, the next chapter will present the theory behind the novel bilateral teleoperation algorithm developed in this work.

Chapter 3

Time Delayed Output Feedback Bilateral Teleoperation

3.1 Introduction

This chapter presents a novel bilateral teleoperation algorithm for n degree of freedom (DOF) nonlinear manipulators that provides the benefits of a position-force architecture in terms of transparency and force tracking, but like the position-position architecture does not require the use of force sensors. Unknown input sliding mode observers provide a useful framework for force estimation in robotics, as shown in our earlier work for 1-DOF systems in [87]. By treating the external forces acting on a manipulator as unknown inputs, those forces can be recovered by the observer in finite time. This work makes use of second order sliding mode observers, based on the work of Davila *et al.* [31].

Force feedback is a key element of bilateral teleoperation. It is what allows the operator to have a sense of presence at the remote environment. By being able to feel the interacting forces, one may be better able to work with the remote environment. Consider the case of remote surgeries. Without a sense of the remote forces, it would be possible for a surgeon to apply too much force with the surgical tools, potentially injuring a patient [88]. However, the introduction of force feedback adds additional challenges in terms of ensuring stability in the presence of time delays for the bilateral teleoperation system. As a result, after presenting the new algorithm, a rigorous stability analysis will be performed of both the master and slave manipulators, as well as an analysis of the entire closed loop system in contact with various environmental models.

This bilateral teleoperation control algorithm uses force feedback with n-DOF master and slave manipulators at each side of the communications. Although similar to Cho *et al.* [89], which deals with linear 1-DOF systems and requires measurement of positions, velocities, and external forces, the algorithm presented here is developed for nonlinear n-DOF plants and requires only position measurements. A sliding mode controller is used at the slave side to ensure a desired closed loop impedance and tracking of the delayed master trajectory. A computed torque method impedance controller at the master side is used to give the master a desired impedance and to apply the reflected slave environment force back to the master. This is an output feedback algorithm, so robot position measurements drive observers that estimate both the state and the external forces. The control algorithm in this work is designed in the manipulator Cartesian space and the slave environment is modeled as an n-DOF system acting on the slave end effector. This is done since it is useful to be able to specify a desired impedance for each degree of freedom of the end effector. A block diagram representation of this system is given in Figure 3.1. The input to the master system is the force applied by the human operator. The master trajectory estimate is transmitted through the delayed communications to the slave. The slave interacts with the environment through its state output, and receives as input the force applied by the environment. Finally, the force estimate exerted on the slave by the environment is reflected back to the master through the delayed communications.

Since the focus of teleoperation is in using robots to accomplish tasks, a discussion of the robot dynamic models used will first be given in Section 3.2. The new bilateral teleoperation algorithm is then presented in Section 3.3. Sections 3.4 and 3.5 develop proofs of stability for the closed loop teleoperation system under various time delay restrictions. A transparency analysis of the closed loop teleoperator is given in Section 3.7. The issue of robustness to unmodeled dynamics is addressed in Section 3.8.

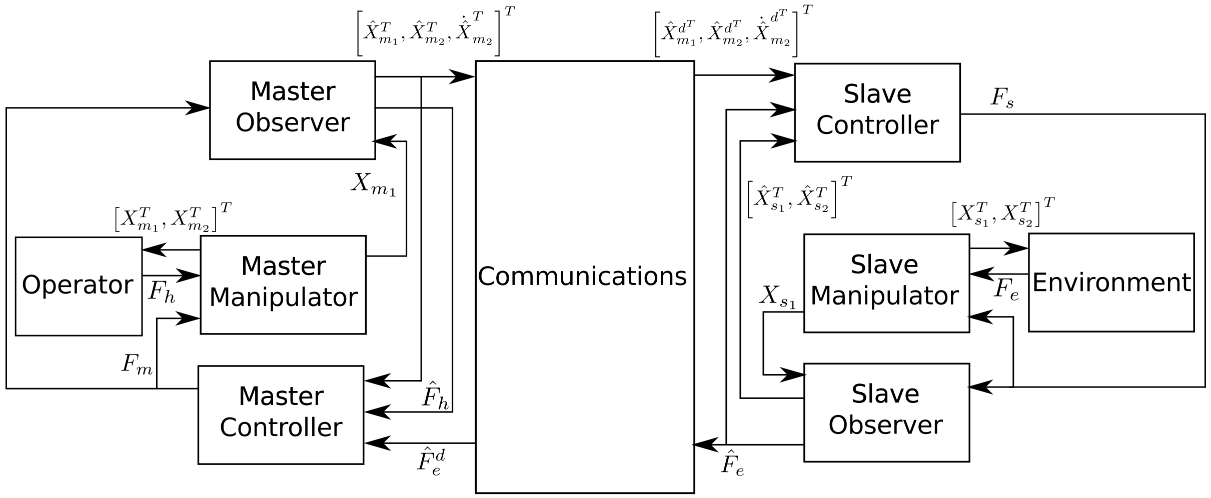

$\left[\hat{X}_{m_1}^T, \hat{X}_{m_2}^T, \dot{\hat{X}}_{m_2}^T\right]^T$
$\left[\hat{X}_{m_1}^{d^T}, \hat{X}_{m_2}^{d^T}, \dot{\hat{X}}_{m_2}^{d^T}\right]^T$
Master Observer
Slave Controller
F_s
X_{m_1}
$\left[\hat{X}_{s_1}^T, \hat{X}_{s_2}^T\right]^T$
$[X_{m_1}^T, X_{m_2}^T]^T$
$[X_{s_1}^T, X_{s_2}^T]^T$
Operator
F_h
Master Manipulator
Communications
X_{s_1}
Slave Manipulator
F_e
Environment
F_m
Master Controller
$\hat{F}_h$
Slave Observer
$\hat{F}_e^d$
$\hat{F}_e$

Figure 3.1: Block diagram of the entire teleoperation system.

3.2 Robot Dynamics

Consider the following master manipulator dynamics in joint space,

$$T_m = M_m(q_m)\ddot{q}_m + h_m(q_m, \dot{q}_m) - T_h \tag{3.1}$$

where $q_m \in R^n$ is the vector of joint positions, $T_m \in R^n$ is the vector of input torques, $M_m(q_m) \in R^{n \times n}$ is the mass matrix, $h_m(q_m, \dot{q}_m) \in R^n$ is a vector of other nonlinear terms, including gravity and Coriolis effects, and $T_h \in R^n$ is the vector of external torques applied by the human. Similarly, the slave dynamics in joint space are given as,

$$T_s = M_s(q_s)\ddot{q}_s + h_s(q_s, \dot{q}_s) + T_e \tag{3.2}$$

where $q_s \in R^n$ is the vector of joint positions, $T_s \in R^n$ is the vector of input torques, $M_s(q_s) \in R^{n \times n}$ is the mass matrix, $h_s(q_s, \dot{q}_s) \in R^n$ is a vector of other nonlinear terms, including gravity and Coriolis effects, and $T_e \in R^n$ is the vector of external torques applied by the environment. It is assumed that these manipulators are not kinematically redundant, leading to a square Jacobian matrix.

Control design will be performed in the robot task space. In order to determine the the task space dynamics, consider first the master dynamics (3.1). Following [90], the relationship between the end effector coordinates $X_m \in R^n$ and the joint angles $q_m \in R^n$ is expressed through the forward kinematics function $f_m(q_m)$ as,

$$X_m = f_m(q_m) \tag{3.3}$$

Note that this expression is only valid in areas that do not pass through kinematic singularities. So for this work, the assumption is made that robots avoid regions of singularities. This can be assured by proper trajectory planning for the manipulators. The relationship between the end effector velocities and joint velocities is obtained by differentiating (3.3). This yields,

$$\dot{X}_m = J_m(q_m)\dot{q}_m \tag{3.4}$$

where $J_m(q_m)$ is the manipulator Jacobian. The accelerations in joint space and task space are related through the time derivative of (3.4) as,

$$\ddot{X}_m = J_m(q_m)\ddot{q}_m + \dot{J}_m(q_m)\dot{q}_m \tag{3.5}$$

Note that the reaction torque T_h from the human is related to the reaction force at the end effector through the Jacobian. That is,

$$T_h = J_m^T(q_m)F_h$$

Next express the master dynamics (3.1) in terms of the acceleration as,

$$\ddot{q}_m = M_m^{-1}(q_m)\left(T_m - h_m(q_m, \dot{q}_m) + J_m^T(q_m)F_h\right) \tag{3.6}$$

Substituting (3.6) into (3.5) yields,

$$\begin{aligned}\ddot{X}_m &= J_m(q_m)M_m^{-1}(q_m)T_m - J_m(q_m)M_m^{-1}(q_m)h_m(q_m,\dot{q}_m) + J_m(q_m)M_m^{-1}(q_m)J_m^T(q_m)F_h \\ &\quad +\dot{J}_m(q_m)\dot{q}_m \end{aligned} \tag{3.7}$$

Multiplying both sides of (3.7) by,

$$(J_m(q_m)^T)^{-1}M_m(q_m)J_m(q_m)^{-1}$$

yields,

$$\begin{aligned}(J_m(q_m)^T)^{-1}M_m(q_m)J_m(q_m)^{-1}\ddot{X}_m &= (J_m(q_m)^T)^{-1}T_m - (J_m(q_m)^T)^{-1}h_m(q_m,\dot{q}_m) + F_h \\ &\quad + (J_m(q_m)^T)^{-1}M_m(q_m)J_m(q_m)^{-1}\dot{J}_m(q_m)\dot{q}_m\end{aligned}$$

Note that, from (3.4), $\dot{q}_m = J_m^{-1}(q_m)\dot{X}_m$. Also note that the term $(J_m(q_m)^T)^{-1}T_m$ is exactly the expression for the input force F_m acting on the end effector. Defining,

$$\begin{aligned}\bar{M}_m(X_m) &= (J_m(q_m)^T)^{-1}M_m(q_m)J_m(q_m)^{-1} \\ \bar{h}_m(X_m,\dot{X}_m) &= (J_m(q_m)^T)^{-1}h_m(q_m,\dot{q}_m) - \bar{M}_m(X_m)\dot{J}_m(q_m)J_m(q_m)^{-1}\dot{X}_m\end{aligned}$$

leads to the following expression for the master dynamics,

$$\bar{M}_m(X_m)\ddot{X}_m = F_m - \bar{h}_m(X_m,\dot{X}_m) + F_h \tag{3.8}$$

Rearranging this expression yields,

$$\bar{M}_m(X_m)\ddot{X}_m + \bar{h}_m(X_m,\dot{X}_m) = F_m + F_h \tag{3.9}$$

This equation represents the master manipulator equations expressed in the task space.

The same equation development can be performed for the slave manipulator. The only difference is the sign of the force term in the slave dynamics. This leads to the following task space description for the slave dynamics,

$$\bar{M}_s(X_s)\ddot{X}_s + \bar{h}_s(X_s, \dot{X}_s) = F_s - F_e \tag{3.10}$$

where the terms with a subscript s are defined in the same way as for the master dynamics. The term $F_e \in R^n$ is the vector of external forces applied to the end effector by the environment. Expressing these dynamics in state space gives, for the master,

$$\begin{aligned} \dot{X}_{m_1} &= X_{m_2} && (3.11) \\ \dot{X}_{m_2} &= \bar{M}_m^{-1}(X_{m_1})\left(-\bar{h}_m(X_{m_1}, X_{m_2}) + F_m + F_h\right) && (3.12) \end{aligned}$$

where $X_{m_1} \in R^n$ is the vector of positions and $X_{m_2} \in R^n$ is the vector of velocities. Defining the slave states similarly, the slave state space representation of the dynamics is given as,

$$\begin{aligned} \dot{X}_{s_1} &= X_{s_2} && (3.13) \\ \dot{X}_{s_2} &= \bar{M}_s^{-1}(X_{s_1})\left(-\bar{h}_s(X_{s_1}, X_{s_2}) + F_s - F_e\right) && (3.14) \end{aligned}$$

It is these state space expressions of the robot dynamics expressed in task space that will be used in all further development of the algorithm.

3.3 New Algorithm

Having developed the expressions for the robot dynamics that are used in the control design, the teleoperation algorithm will now be presented. This algorithm requires only position measurements, and as a result relies on second order sliding mode observers for both state and external force estimation. The observers used in the algorithm will first be presented, followed by the control laws used at the master and slave sides.

The observers used here are direct applications of the MIMO observers presented in Section 2.2.4, where the force signals are viewed as unknown inputs to the plants. The observer for the master plant is given as,

$$\begin{aligned}\dot{\hat{X}}_{m_1} &= \hat{X}_{m_2} + z_{m_1} &(3.15)\\ \dot{\hat{X}}_{m_2} &= \bar{M}_m^{-1}(\hat{X}_{m_1})\left(-\bar{h}_m(\hat{X}_{m_1}, \hat{X}_{m_2}) + F_m\right) + z_{m_2} &(3.16)\end{aligned}$$

where $z_{m_1} \in R^n$ and $z_{m_2} \in R^n$. The i-th element of vector z_{m_1} is given as,

$$z_{m_{1i}} = \lambda_{m_i}|X_{m_{1i}} - \hat{X}_{m_{1i}}|^{1/2}\text{sign}(X_{m_{1i}} - \hat{X}_{m_{1i}}) \tag{3.17}$$

and the i-th element of vector z_{m_2} is given as,

$$z_{m_{2i}} = \alpha_{m_i}\text{sign}(X_{m_{1i}} - \hat{X}_{m_{1i}}) \tag{3.18}$$

where λ_{m_i} and α_{m_i} are constants whose values must be chosen according to (2.57) and (2.56). Note that the human force exerted on the end effector does not appear at all in the observer. Regardless of this, finite time convergence of the state estimates is achieved, and an estimate of the human force is obtained as well. It is this force estimate that is used in the control law. The human force estimate is obtained from the equivalent output injection term z_{m_2} as,

$$\hat{F}_h = \bar{M}_m(\hat{X}_{m_1})z_{m_2eq} \tag{3.19}$$

where z_{m_2eq} represents a low pass filtering operation on z_{m_2} in order to obtain the equivalent output injection signal [20]. The second order sliding mode observer for the slave dynamics takes the same form as the master side observer. The equations for the slave observer are expressed as,

$$\begin{aligned}\dot{\hat{X}}_{s_1} &= \hat{X}_{s_2} + z_{s_1} && (3.20)\\ \dot{\hat{X}}_{s_2} &= \bar{M}_s^{-1}(\hat{X}_{s_1})\left(-\bar{h}_s(\hat{X}_{s_1}, \hat{X}_{s_2}) + F_s\right) + z_{s_2} && (3.21)\end{aligned}$$

where $z_{s_1} \in R^n$ and $z_{s_2} \in R^n$. The i-th element of vectors z_{s_1} and z_{s_2} are defined analogously to the i-th elements of z_{m_1} and z_{m_2}, respectively. The estimate of the environmental force acting on the slave is obtained from the equivalent output injection term z_{s_2} as,

$$\hat{F}_e = -\bar{M}_s(\hat{X}_{s_1})z_{s_2eq} \tag{3.22}$$

The master and slave control laws are next presented. Before presenting these controllers, some notation is introduced. A signal $x(t)$ delayed by T_1 seconds is represented as,

$$x^{d_1}(t) \equiv x(t - T_1)$$

A signal $x(t)$ delayed by T_2 seconds is represented as,

$$x^{d_2}(t) \equiv x(t - T_2)$$

Similarly, a signal delayed by two times delays, T_1 and T_2, is represented as,

$$x^{dd}(t) \equiv x(t - T_1 - T_2)$$

The master control law is a computed torque method controller to decouple and linearize each degree of freedom in the task space. The outer loop controller is specified as,

$$F_m = \bar{M}_m(\hat{X}_{m_1})v_m + \bar{h}_m(\hat{X}_{m_1}, \hat{X}_{m_2}) - \hat{F}_h \tag{3.23}$$

The inner impedance controller, to provide each degree of freedom with desired impedance characteristics, is given as,

$$v_m = \tilde{M}_m^{-1}\left(-\tilde{B}_m\hat{X}_{m_2} - \tilde{K}_m\hat{X}_{m_1} + \hat{F}_h - \hat{F}_e^{d_2}\right) \tag{3.24}$$

where $\tilde{M}_m \in R^{n\times n}$ is the diagonal constant matrix that specifies the desired mass characteristic for each degree of freedom. The desired mass for the i-th degree of freedom is given by the i-th diagonal of $\tilde{M}_m$. The matrices $\tilde{B}_m \in R^{n\times n}$ and $\tilde{K}_m \in R^{n\times n}$ are also diagonal constant matrices representing the desired damping and stiffness values for each degree of freedom. Note that, in general, it may not always be desirable to have a non-zero value for $\tilde{K}_m$. This spring term draws the manipulator back to the origin. Viewing the manipulator as a tool, the operator may want to be able to place the manipulator in the task space and have it stay there. The spring term prevents this from happening. In practice, this term may be set to zero. Note, though, that without a spring force the manipulator is not asymptotically stable. The control law (3.23) and (3.24) ensures that, after convergence of the observers, the master manipulator has the desired mass-spring-damper impedance characteristics for each degree of freedom. As with any standard computed torque method controller, there will not be a perfect cancellation of nonlinearities in practice. A robustness analysis of the algorithm is presented in Section 3.8.

In order to present the slave controller, define the master-slave position and velocity tracking error as $e_{r_1} = X_{s_1} - X_{m_1}^{d_1} \in R^n$ and $e_{r_2} = X_{s_2} - X_{m_2}^{d_1} \in R^n$. The controller is designed in order to give each degree of freedom of the end effector a desired impedance characteristic. Intuitively, there is value in giving impedance characteristics to the end effector, which will actually be interacting with the task. Controlling the mass, spring, and damping characteristics of the end effector allows the operator to tune the system based on the particular application. The desired impedance model for the slave end effector is expressed as,

$$Z_s(s) = \frac{F_e(s)}{V_e(s)} = -\left(\tilde{M}_s s + \tilde{B}_s + \frac{\tilde{K}_s}{s}\right) \tag{3.25}$$

where $F_e(s)$ represents the Laplace transform of the environmental force, $V_e(s)$ represents the Laplace transform of the tracking error velocity, and the matrices $\tilde{M}_s$, $\tilde{B}_s$, and $\tilde{K}_s$ are defined as in the master controller, but for the slave impedances. This impedance model gives rise to the following n-DOF equation of dynamics,

$$I = \tilde{M}_s \dot{e}_{r_2} + \tilde{B}_s e_{r_2} + \tilde{K}_s e_{r_1} + F_e = 0 \tag{3.26}$$

This equation is obtained from (3.25) by taking an inverse Laplace transform. When (3.26) is satisfied, the slave has the desired closed loop impedance and asymptotically tracks the delayed master trajectories. In order to ensure that this desired impedance characteristic is satisfied, a sliding surface for the slave controller is defined as,

$$s = \int_0^t \tilde{M}_s^{-1} I(\tau)\, d\tau = 0 \tag{3.27}$$

Once the slave sliding mode controller has driven the system trajectories to $s = 0$ then (3.26) will be satisfied and the slave manipulator will have the desired closed loop behaviour at the end effector.

However, this work examines output feedback control. An output feedback version of the end effector dynamics that yield the desired impedance model is defined as,

$$\hat{I} = \tilde{M}_s \dot{\hat{e}}_{r_2} + \tilde{B}_s \hat{e}_{r_2} + \tilde{K}_s \hat{e}_{r_1} + \hat{F}_e = 0 \tag{3.28}$$

where $\hat{e}_{r_1} = \hat{X}_{s_1} - \hat{X}_{m_1}^{d_1}$ and $\hat{e}_{r_2} = \hat{X}_{s_2} - \hat{X}_{m_2}^{d_1}$. Now, the sliding surface in the output feedback case is defined as,

$$\begin{aligned}
\hat{s} &= \int_0^t \tilde{M}_s^{-1} \hat{I}(\tau)\, d\tau && (3.29)\\
&= \hat{e}_{r_2} + \int_0^t \left(\tilde{M}_s^{-1} \tilde{B}_s \hat{e}_{r_2} + \tilde{M}_s^{-1} \tilde{K}_s \hat{e}_{r_1} + \tilde{M}_s^{-1} \hat{F}_e \right) d\tau && (3.30)\\
&= 0 && (3.31)
\end{aligned}$$

Then, the slave side sliding mode controller is given as,

$$
\begin{aligned}
F_s \;=\; & -\bar{M}_s(\hat{X}_{s_1})\Big[\tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{s_1} + \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{s_2} - \bar{M}_s^{-1}(\hat{X}_{s_1})\bar{h}_s(\hat{X}_{s_1}, \hat{X}_{s_2}) \\
& + (\tilde{M}_m^{-1}\tilde{K}_m - \tilde{M}_s^{-1}\tilde{K}_s)\hat{X}_{m_1}^{d_1} + (\tilde{M}_m^{-1}\tilde{B}_m - \tilde{M}_s^{-1}\tilde{B}_s)\hat{X}_{m_2}^{d_1} - (\tilde{M}_m^{-1} - \bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1}))\hat{F}_h^{d_1} \\
& +\tilde{M}_m^{-1}\hat{F}_e^{dd} + \tilde{M}_s^{-1}\hat{F}_e + z_{s_2eq} - z_{m_2eq}^{d_1} + K_g\text{sign}(\hat{s})\Big] \qquad (3.32)
\end{aligned}
$$

where $K_g = k_g I \in R^{n\times n}$ and k_g is a scalar whose value will be specified later. This control law comes out of a Lyapunov stability analysis of the sliding mode dynamics. The control law is chosen to ensure that the sliding mode dynamics are stable and additionally that they satisfy the property of finite time convergence. This analysis is presented in Section 3.3.1

To summarize, for master plant (3.11), (3.12) and slave plant (3.13), (3.14) connected bilaterally through a time delay of T_1 seconds from the master to the slave and T_2 seconds from the slave to the master, the system may be controlled using the master control law (3.23) and (3.24) with master side observer (3.15), (3.16) and slave sliding mode control law (3.32) with slave side observer (3.20), (3.21).

Having presented the bilateral teleoperation algorithm in its entirety, the next section will develop the stability analysis of this system.

3.3.1 Master and Slave Stability Analysis

In order to show stability of this system, the following assumption is made.

Assumption 3.3.1. *The external forces acting on both master and slave are bounded for all time with some known upper bounds.*

The next theorem presents the main stability result for each of the master and slave sides, showing asymptotic stability of each side under output feedback with force estimation, and in the presence of time delays in the communications.

Theorem 3.3.1. *Consider master plant (3.11), (3.12) and slave plant (3.13), (3.14) connected bilaterally through a time delay of T_1 seconds from the master to the slave and T_2*

seconds from the slave to the master, with master control law (3.23) and (3.24), master side observer (3.15), (3.16) and slave sliding mode control law (3.32) with slave side observer (3.20), (3.21). Then, there exists observer gains λ_m, α_m, λ_s, α_s and a sliding mode controller gain $K_g = k_g I$ where,

$$k_g \geq 2\sqrt{n}\left(\max_i \alpha_{s_i} + \max_i \alpha_{m_i}\right) + \varepsilon_g \tag{3.33}$$

for any $\varepsilon_g > 0$ such that the state estimates recover the true state in finite time, and the master and slave plants have the desired impedance model.

Proof. The first step is to show that the estimated states converge to the true states in finite time. Observer convergence for both the master and slave is guaranteed from Corollary 2.2.1, provided that the observer gains are chosen according to (2.56) and (2.57). The master observer states will be exactly the master states after T_m seconds, and likewise for the slave observer states after T_s seconds.

In order to show stability of the slave system, a Lyapunov function is used, and the controller is selected to ensure that the sliding mode dynamics are finite time stable. Due to the definition of the sliding surface (3.30), expressions for the dynamics of both the master and slave observers are required. The master observer dynamics, after substituting the master control law (3.23) and (3.24) into the original expression for the master observer dynamics (3.16), are given by,

$$\begin{aligned}
\dot{\hat{X}}_{m_1} &= \hat{X}_{m_2} + z_{m_1} && (3.34)\\
\dot{\hat{X}}_{m_2} &= -\tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2} - \tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1} + \tilde{M}_m^{-1}\hat{F}_h \\
&\quad -\tilde{M}_m^{-1}\hat{F}_e^{d_2} - \bar{M}_m^{-1}(\hat{X}_{m_1})\hat{F}_h + z_{m_2} && (3.35)
\end{aligned}$$

Substituting the delayed version of the closed loop master observer dynamics (3.35) and the slave observer dynamics (3.21) into the sliding surface (3.30) and simplifying, one arrives at,

$$\begin{aligned}\hat{s} \;=\; & \int_0^t \Big(-\bar{M}_s^{-1}(\hat{X}_{s_1})\bar{h}_s(\hat{X}_{s_1},\hat{X}_{s_2}) + \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{s_1} \\ & +\tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{s_2} - \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{m_1}^{d_1} - \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{m_2}^{d_1} \\ & + \tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1}^{d_1} + \tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2}^{d_1} \\ & -(\tilde{M}_m^{-1} - \bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1}))\hat{F}_h^{d_1} + \tilde{M}_m^{-1}\hat{F}_e^{dd} \\ & +\tilde{M}_s^{-1}\hat{F}_e + \bar{M}_s^{-1}(\hat{X}_{s_1})F_s + z_{s_2} - z_{m_2}^{d_1}\Big)\; d\tau \end{aligned} \tag{3.36}$$

Now consider the Lyapunov function candidate,

$$V_s = \frac{1}{2}\hat{s}^T\hat{s} \tag{3.37}$$

Taking the derivative of V_s along the trajectories of the system,

$$\dot{V}_s \;=\; \hat{s}^T\dot{\hat{s}} \tag{3.38}$$

$$\begin{aligned} \;=\; & \hat{s}^T\Big[\Big(-\bar{M}_s^{-1}(\hat{X}_{s_1})\bar{h}_s(\hat{X}_{s_1},\hat{X}_{s_2}) + \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{s_1} \\ & +\tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{s_2} - \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{m_1}^{d_1} - \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{m_2}^{d_1} \\ & +\tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1}^{d_1} + \tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2}^{d_1} \\ & -(\tilde{M}_m^{-1} - \bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1}))\hat{F}_h^{d_1} + \tilde{M}_m^{-1}\hat{F}_e^{dd} \\ & +\tilde{M}_s^{-1}\hat{F}_e + \bar{M}_s^{-1}(\hat{X}_{s_1})F_s + z_{s_2} - z_{m_2}^{d_1}\Big)\Big] \end{aligned} \tag{3.39}$$

Now choosing the control law as (3.32), substituting into (3.39) and simplifying yields,

$$\dot{V}_s = -\hat{s}^T k_g I \text{sign}(\hat{s}) + \hat{s}^T\left(z_{s_2} - z_{s_2eq} - z_{m_2}^{d_1} + z_{m_2eq}^{d_1}\right) \tag{3.40}$$

Note that $\hat{s}^T\text{sign}(\hat{s}) = \sum_{i=1}^n |\hat{s}_i| = ||\hat{s}||_1$. Then,

$$\begin{aligned}\dot{V}_s \;&=\; -k_g||\hat{s}||_1 + \hat{s}^T\left(z_{s_2} - z_{s_2eq} - z_{m_2}^{d_1} + z_{m_2eq}^{d_1}\right) \\ &\leq\; -k_g||\hat{s}||_1 + |\hat{s}^T\left(z_{s_2} - z_{s_2eq} - z_{m_2}^{d_1} + z_{m_2eq}^{d_1}\right)| \end{aligned}$$

The second term on the right hand side of the above inequality represents an inner product of two vectors. By the Cauchy-Schwartz inequality,

$$\dot{V}_s \leq -k_g||\hat{s}||_1 + ||\hat{s}||_2 \, ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2 \tag{3.41}$$

Next, recall that $||x||_2 \leq ||x||_1$ in R^n. This leads to,

$$\begin{aligned} \dot{V}_s &\leq -k_g||\hat{s}||_1 + ||\hat{s}||_1 \, ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2 \\ &= -||\hat{s}||_1 \left(k_g - ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2\right) \end{aligned}$$

Note that for a vector $x \in R^n$, $||x||_2 \leq \sqrt{n}||x||_\infty$. Then consider the term $||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2$. It follows from the triangle inequality that,

$$\begin{aligned} ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2 &\leq ||z_{s_2}||_2 + ||z_{s_2eq}||_2 + ||z^{d_1}_{m_2}||_2 + ||z^{d_1}_{m_2eq}||_2 \\ &\leq \sqrt{n}||z_{s_2}||_\infty + \sqrt{n}||z_{s_2eq}||_\infty + \sqrt{n}||z^{d_1}_{m_2}||_\infty + \sqrt{n}||z^{d_1}_{m_2eq}||_\infty \end{aligned}$$

Next note that $||z_{s_2eq}||_\infty \leq ||z_{s_2}||_\infty$ and $||z^{d_1}_{m_2eq}||_\infty \leq ||z^{d_1}_{m_2}||_\infty$. As well, $||z_{s_2}||_\infty = \max_i \alpha_{s_i}$ and $||z^{d_1}_{m_2}||_\infty = \max_i \alpha_{m_i}$, where α_{s_i} and α_{m_i} are the slave and master observer gains, respectively. Then, one can conclude that,

$$\begin{aligned} ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2 &\leq 2\sqrt{n}\max_i \alpha_{s_i} + 2\sqrt{n}\max_i \alpha_{m_i} \\ &= 2\sqrt{n}\left(\max_i \alpha_{s_i} + \max_i \alpha_{m_i}\right) \end{aligned}$$

Choosing k_g as in (3.33) then ensures that $k_g > ||z_{s_2} - z_{s_2eq} - z^{d_1}_{m_2} + z^{d_1}_{m_2eq}||_2$. From there, one arrives at the following inequality,

$$\dot{V}_s < -||\hat{s}||_1\varepsilon_g < 0 \qquad \forall \, ||\hat{s}||_1 \neq 0 \tag{3.42}$$

This guarantees stability, and additionally finite time convergence of the system trajectories to the sliding surface. Finite time convergence is shown using Lemma 2.2.1. In order to show this, define the function $W_1 = \sqrt{2V_s} = ||\hat{s}||_2$. Then,

$$\begin{aligned} D^+W_1 &= \dot{W}_1 \\ &= \frac{dW_1}{dV_s}\dot{V}_s \\ &= \frac{1}{\sqrt{2V_s}}\dot{V}_s \\ &= \frac{1}{||\hat{s}||_2}\dot{V}_s \\ &\leq \frac{-||\hat{s}||_1}{||\hat{s}||_2}\varepsilon_g \\ &\leq \frac{-||\hat{s}||_2}{||\hat{s}||_2}\varepsilon_g \\ &= -\varepsilon_g \end{aligned}$$

So $D^+W_1 \leq -\varepsilon_g$. Define $f_1(t, W_1) = -\varepsilon_g$. Then $D^+W_1 \leq f_1(t, W_1)$. Now consider $\dot{u} = f_1(t, u) = -\varepsilon_g$, $u(t_0) = W_1(\hat{s}(t_0))$. Then,

$$\begin{aligned} \int_{t_0}^{t} \frac{du}{d\tau} d\tau &= -\varepsilon_g \int_{t_0}^{t} d\tau \\ u(t) - u(t_0) &= -\varepsilon_g(t - t_0) \\ u(t) &= u(t_0) - \varepsilon_g(t - t_0) \end{aligned}$$

Making use of Lemma 2.2.1, the result $W_1(t) \leq u(t)$ is obtained. Then,

$$W_1(t) \leq W_1(\hat{s}(t_0)) - \varepsilon_g(t - t_0)$$

Since $W_1(t) = ||\hat{s}(t)||_2$, it is clear from the above that there will be some finite time when $||\hat{s}||_2$ reaches zero. In order to solve for the upper bound on this time, set $W_1(\hat{s}(t_0)) - \varepsilon_g(t - t_0) = 0$. Rearranging, one arrives at,

$$T_g \triangleq \frac{||\hat{s}(t_0)||_2}{\varepsilon_g} + t_0$$

Therefore by the time $t = T_g$ the state $\hat{s}$ will have converged to the sliding surface $\hat{s} = 0$.

Satisfying the conditions of Corollary 2.2.1 ensures that both the master and slave estimation errors will converge to zero within T_m seconds and T_s seconds, respectively. As well, the system trajectories will reach the sliding surface $\hat{s} = 0$ within T_g seconds. Therefore, at $t = \max(T_s, T_m, T_g)$, all observers will have converged, the slave plant will reach the sliding surface $\hat{s} = 0$, and all force estimates will be equal to the actual external forces acting on the plants.

Using the method of equivalent control, one can show that the tracking error dynamics will have exactly the desired characteristic impedance. The equivalent slave control signal, that is the control signal when the system is exactly on the sliding surface, is found by setting $\dot{\hat{s}} = 0$ and solving for the control signal. The equivalent control signal is found as,

$$\begin{aligned} F_{s_{eq}} \quad = \quad & -\bar{M}_s(\hat{X}_{s_1}) \Big[\tilde{M}_s^{-1} \tilde{K}_s \hat{X}_{s_1} + \tilde{M}_s^{-1} \tilde{B}_s \hat{X}_{s_2} \\ & -\bar{M}_s^{-1}(\hat{X}_{s_1}) \bar{h}_s(\hat{X}_{s_1}, \hat{X}_{s_2}) \\ & + (\tilde{M}_m^{-1} \tilde{K}_m - \tilde{M}_s^{-1} \tilde{K}_s) \hat{X}_{m_1}^{d_1} \\ & + (\tilde{M}_m^{-1} \tilde{B}_m - \tilde{M}_s^{-1} \tilde{B}_s) \hat{X}_{m_2}^{d_1} - (\tilde{M}_m^{-1} \\ & - \bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1})) \hat{F}_h^{d_1} \\ & + \tilde{M}_m^{-1} \hat{F}_e^{dd} + \tilde{M}_s^{-1} \hat{F}_e + z_{s_2} - z_{m_2}^{d_1} \Big] \end{aligned} \tag{3.43}$$

In order to analyze the dynamics on the sliding surface, substitute the slave equivalent control (3.43) into the slave dynamics (3.13), (3.14). Simplifying this expression yields,

$$
\begin{align}
\dot{X}_{s_1} &= X_{s_2} \tag{3.44}\\
\dot{X}_{s_2} &= \bar{M}_s^{-1}(X_{s_1})\bar{M}_s(\hat{X}_{s_1})\left[-\tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{s_1} - \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{s_2} - \tilde{M}_s^{-1}\hat{F}_e\right] \nonumber\\
&\quad -\bar{M}_s^{-1}(X_{s_1})(\bar{h}_s(X_{s_1},X_{s_2}) - \bar{h}_s(\hat{X}_{s_1},\hat{X}_{s_2})) - (\bar{M}_s^{-1}(X_{s_1})F_e + \bar{M}_s^{-1}(X_{s_1})\bar{M}_s(\hat{X}_{s_1})z_{s_2}) \nonumber\\
&\quad +\bar{M}_s^{-1}(X_{s_1})\bar{M}_s^{-1}(\hat{X}_{s_1})\left[-\tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1}^{d_1} - \tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2}^{d_1} + \tilde{M}_m^{-1}\hat{F}_h^{d_1} - \tilde{M}_m^{-1}\hat{F}_e^{dd}\right. \nonumber\\
&\quad \left.-\bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1})\hat{F}_h^{d_1} + z_{m_2}^{d_1} + \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{m_2}^{d_1} + \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{m_1}^{d_1}\right] \tag{3.45}
\end{align}
$$

Now note that the equation (3.35), when a delay is inserted into it, is expressed as,

$$
\begin{align}
\dot{\hat{X}}_{m_2}^{d_1} &= -\tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2}^{d_1} - \tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1}^{d_1} + \tilde{M}_m^{-1}\hat{F}_h^{d_1} - \tilde{M}_m^{-1}\hat{F}_e^{dd} \tag{3.46}\\
&\quad -\bar{M}_m^{-1}(\hat{X}_{m_1}^{d_1})\hat{F}_h^{d_1} + z_{m_2}^{d_1} \tag{3.47}
\end{align}
$$

Substituting this equation into (3.45) gives the following slave dynamics on the sliding surface $\hat{s} = 0$,

$$
\begin{align}
\dot{X}_{s_1} &= X_{s_2} \tag{3.48}\\
\dot{X}_{s_2} &= \bar{M}_s^{-1}(X_{s_1})\bar{M}_s(\hat{X}_{s_1})\left[-\tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{s_1} - \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{s_2} - \tilde{M}_s^{-1}\hat{F}_e\right] \nonumber\\
&\quad -\bar{M}_s^{-1}(X_{s_1})(\bar{h}_s(X_{s_1},X_{s_2}) - \bar{h}_s(\hat{X}_{s_1},\hat{X}_{s_2})) - (\bar{M}_s^{-1}(X_{s_1})F_e + \bar{M}_s^{-1}(X_{s_1})\bar{M}_s(\hat{X}_{s_1})z_{s_2}) \nonumber\\
&\quad +\bar{M}_s^{-1}(X_{s_1})\bar{M}_s^{-1}(\hat{X}_{s_1})\left[\dot{\hat{X}}_{m_2}^{d_1} + \tilde{M}_s^{-1}\tilde{B}_s\hat{X}_{m_2}^{d_1} + \tilde{M}_s^{-1}\tilde{K}_s\hat{X}_{m_1}^{d_1}\right] \tag{3.49}
\end{align}
$$

Next, the closed loop expression for the master dynamics is found by substituting the master controller (3.23) and (3.24) into the master plant dynamics (3.11) and (3.12). Making this substitution and simplifying yields,

$$
\begin{align}
\dot{X}_{m_1} &= X_{m_2} \tag{3.50}\\
\dot{X}_{m_2} &= \bar{M}_m^{-1}(X_{m_1})\bar{M}_m(\hat{X}_{m_1})\left[-\tilde{M}_m^{-1}\tilde{B}_m\hat{X}_{m_2} - \tilde{M}_m^{-1}\tilde{K}_m\hat{X}_{m_1} + \tilde{M}_m^{-1}\hat{F}_h - \tilde{M}_m^{-1}\hat{F}_e^{d_2}\right] \nonumber\\
&\quad -\bar{M}_m^{-1}(X_{m_1})\left(\bar{h}_m(X_{m_1},X_{m_2}) - \bar{h}_m(\hat{X}_{m_1},\hat{X}_{m_2})\right) \nonumber\\
&\quad +\bar{M}_m^{-1}(X_{m_1})\left(F_h - \hat{F}_h\right) \tag{3.51}
\end{align}
$$

Once time $t = \max(T_s, T_m, T_g)$ is reached, all observers will have converged, the slave plant will reach the sliding surface $\hat{s} = 0$, and all force estimates will be equal to the actual external forces acting on the plants. At this point, the slave closed loop dynamics reduce to,

$$\begin{aligned}\dot{X}_{s_1} &= X_{s_2} &(3.52)\\ \dot{X}_{s_2} &= -\tilde{M}_s^{-1}\tilde{K}_s X_{s_1} - \tilde{M}_s^{-1}\tilde{B}_s X_{s_2} - \tilde{M}_s^{-1}F_e + \dot{X}^{d_1}_{m_2} + \tilde{M}_s^{-1}\tilde{B}_s X^{d_1}_{m_2} \\ &\quad + \tilde{M}_s^{-1}\tilde{K}_s X^{d_1}_{m_1} &(3.53)\end{aligned}$$

Similarly, the master closed loop dynamics become,

$$\begin{aligned}\dot{X}_{m_1} &= X_{m_2} &(3.54)\\ \dot{X}_{m_2} &= -\tilde{M}_m^{-1}\tilde{K}_m X_{m_1} - \tilde{M}_m^{-1}\tilde{B}_m X_{m_2} + \tilde{M}_m^{-1}F_h - \tilde{M}_m^{-1}F_e^{d_2} &(3.55)\end{aligned}$$

In order to determine the tracking error dynamics $e_{r_1} = X_{s_1} - X^{d_1}_{m_1}$ and $e_{r_2} = X_{s_s} - X^{d_1}_{m_2}$, subtract the delayed version of (3.54) from (3.52) and the delayed version of (3.55) from (3.53). This yields,

$$\begin{aligned}\dot{e}_{r_1} &= e_{r_2} &(3.56)\\ \dot{e}_{r_2} &= -\tilde{M}_s^{-1}\tilde{K}_s e_{r_1} - \tilde{M}_s^{-1}\tilde{B}_s e_{r_2} - \tilde{M}_s^{-1}F_e &(3.57)\end{aligned}$$

This result is exactly the equation of dynamics representing the desired characteristic impedance for the tracking error dynamics. This equivalent control analysis has shown that on the sliding surface the master and slave dynamics remain stable, and the desired characteristic impedance is achieved.

□

Theorem 3.3.1 guarantees stability of each of the master and slave plants. However, stability of the entire closed loop is not addressed in this result. The next section addresses the issue of closed loop stability under a variety of environmental conditions.

3.4 Closed Loop Stability Independent of Delay

Having guaranteed stability for each of the master and slave manipulators with their associated observers and controllers, it remains to show that the entire closed loop can be stabilized in the presence of time delays. This section will present closed loop stability results for two situations. These cases deal with the situation where the system may be stabilized for any size of delay, and the delay need not be known *a priori*. Different results are developed for cases where the environment is a linear spring damper and where the environment is a nonlinear finite-gain stable system.

3.4.1 Linear Slave Side Environment

This case examines the situation where the slave end effector is in constant contact with a linear environment. This environment is modeled as a set of n spring-dampers in connection with each degree of freedom of the end effector. The environment is located at $X_{s_1} = X_c$ and is described by the model,

$$F_e = K_e(X_{s_1} - X_c) + B_e X_{s_2} \tag{3.58}$$

where $K_e \in R^{n \times n}$ is a diagonal matrix of spring constants and $B_e \in R^{n \times n}$ is a diagonal matrix of damping constants. This sort of linear environment model may be found in the teleoperation literature. See for example [71, 78, 91]. When this environment is in contact with the slave manipulator, the equilibrium point of the slave system is no longer the origin.

Figure 3.2 shows the layout of each of the components of this closed loop system. In order to perform a stability analysis, a new state is defined to express the slave as a system with an equilibrium at the origin. Define the new state as,

$$X'_{s_1} = X_{s_1} - \left(\tilde{K}_s + K_e\right)^{-1} K_e X_c \tag{3.59}$$

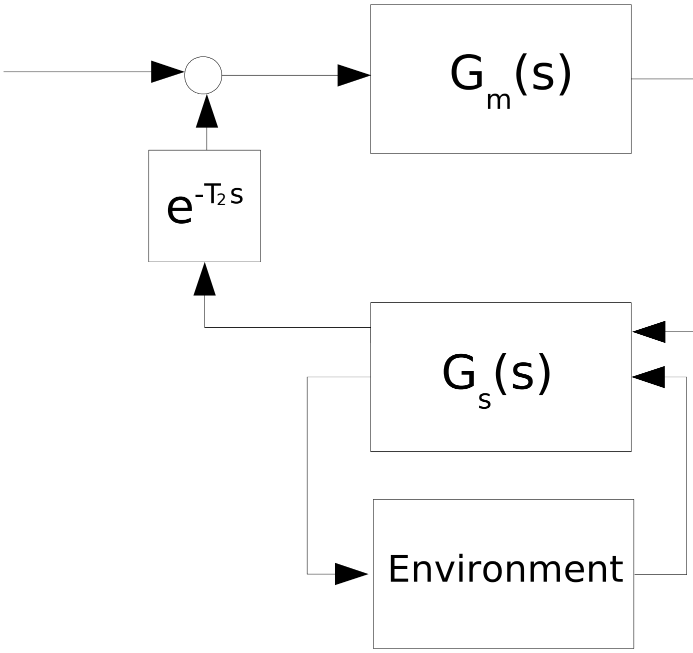


Figure 3.2: Block diagram of the master system and the slave and environment subsystem.

Applying this transformation to the slave plant ensures that the origin of the system in the new states $X_s = [X_{s_1}'^T, X_{s_2}^T]^T$ is at the origin.

In this section, the Small Gain Theorem is used to develop sufficient conditions to ensure closed loop stability regardless of the time delays. Before presenting the results, the Small Gain Theorem [16], is given. Figure 3.3 shows two systems, $H_1 : L_e^m \to L_e^q$ and $H_2 : L_e^q \to L_e^m$, in the standard feedback configuration. Both systems are finite-gain L stable, with gains γ_1 and γ_2 respectively, and the feedback system is well defined. Then, the following result from [16] can be stated.

Theorem 3.4.1. *Under the preceding assumptions, the feedback connection of Figure 3.3 is finite-gain L stable if $\gamma_1 \gamma_2 < 1$.*

In order to develop the stability results in this case, a transfer function representation of the master system is used. The master system has as inputs the sum of the human force and the delayed environmental force. Define this input signal as $e_1 = F_h - F_e(t - T_2) \in R^n$. It produces as outputs position, velocity, and acceleration. Define the master output $Y_m =$

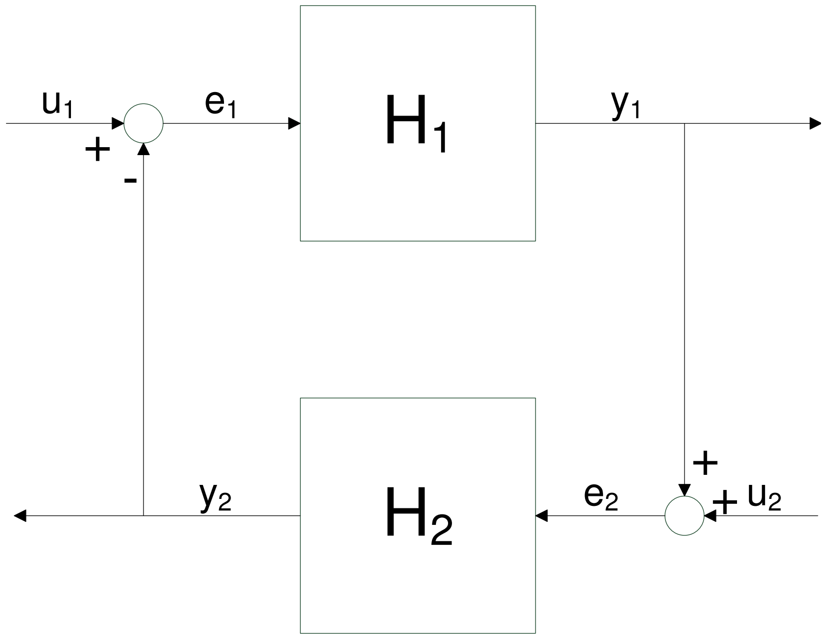


Figure 3.3: The standard feedback configuration for the Small Gain Theorem.

$[X_{m_1}^T, X_{m_2}^T, \dot{X}_{m_2}^T]^T$. Expressing the master dynamics in the familiar state-space notation for linear systems one arrives at,

$$\begin{bmatrix} \dot{X}_{m_1} \\ \dot{X}_{m_2} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -\tilde{M}_m^{-1}\tilde{K}_m & -\tilde{M}_m^{-1}\tilde{B}_m \end{bmatrix}}_{A_m} \begin{bmatrix} X_{m_1} \\ X_{m_2} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \tilde{M}_m^{-1} \end{bmatrix}}_{B_m} e_1 \tag{3.60}$$

$$Y_m = \underbrace{\begin{bmatrix} I & 0 \\ 0 & I \\ -\tilde{M}_m^{-1}\tilde{K}_m & -\tilde{M}_m^{-1}\tilde{B}_m \end{bmatrix}}_{C_m} \begin{bmatrix} X_{m_1} \\ X_{m_2} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \tilde{M}_m^{-1} \end{bmatrix}}_{D_m} e_1 \tag{3.61}$$

The slave+environment system receives $e_2 = Y_m(t - T_1) \in R^{3n}$ as input. The environmental forces acting on the slave end effector are taken as its output, that is, $Y_s = K_e X'_{s_1} + B_e X_{s_2}$. For the slave+environment, the standard linear state-space form for the dynamics (given in terms of the transformed state X'_{s_1} so the equilibrium point is at the origin) is expressed as,

$$\left[\begin{array}{c} \dot{X}'_{s_1} \\ \dot{X}_{s_2} \end{array}\right] = \underbrace{\left[\begin{array}{cc} 0 & I \\ -\tilde{M}_s^{-1}\left(\tilde{K}_s + K_e\right) & -\tilde{M}_s^{-1}\left(\tilde{B}_s + B_e\right) \end{array}\right]}_{A_s}\left[\begin{array}{c} X'_{s_1} \\ X_{s_2} \end{array}\right] \tag{3.62}$$

$$+\underbrace{\left[\begin{array}{ccc} 0 & 0 & 0 \\ \tilde{M}_s^{-1}\tilde{K}_s & \tilde{M}_s^{-1}\tilde{B}_s & I \end{array}\right]}_{B_s} e_2 \tag{3.63}$$

$$Y_s = \underbrace{\left[\begin{array}{cc} K_e & B_e \end{array}\right]}_{C_s}\left[\begin{array}{c} X'_{s_1} \\ X_{s_2} \end{array}\right] \tag{3.64}$$

The master transfer function matrix $G_m(s) \in C^{3n\times n}$ can be found as,

$$G_m(s) = \left[\begin{array}{c} G_{m_1}(s) \\ G_{m_2}(s) \\ G_{m_3}(s) \end{array}\right] \tag{3.65}$$

where,

$$G_{m_1}(s) = \operatorname{diag}\left(\frac{1}{m_{m_i}s^2 + b_{m_i}s + k_{m_i}}\right) \tag{3.66}$$

$$G_{m_2}(s) = \operatorname{diag}\left(\frac{s}{m_{m_i}s^2 + b_{m_i}s + k_{m_i}}\right) \tag{3.67}$$

$$G_{m_3}(s) = \operatorname{diag}\left(\frac{s^2}{m_{m_i}s^2 + b_{m_i}s + k_{m_i}}\right) \tag{3.68}$$

for $i = 1 \ldots n$, where m_{m_i}, b_{m_i}, and k_{m_i} are the i-th diagonals of $\tilde{M}_m$, $\tilde{B}_m$, and $\tilde{K}_m$ respectively. This transfer function matrix takes one input vector, the sum of the applied human force and the delayed slave side force, and has three output vectors – master position, velocity, and acceleration. The time delays in both the feedforward and feedback paths may be combined with each transfer function matrix to yield the following transfer matrices,

$$G_m^{d_1}(s) = G_m(s)e^{-T_1 s},\ G_s^{d_2}(s) = G_s(s)e^{-T_2 s} \tag{3.69}$$

Stability of the feedback interconnection of $G_m^{d_1}(s)$ and $G_s^{d_2}(s)$ will now be shown.

Assumption 3.4.1. *An upper bound on the ∞-norm of the slave+environment system $G_s(s)$ is known.*

Theorem 3.4.2. *Consider master system (3.54) and (3.55) connected through time delays to slave system (3.52) and (3.53). Define,*

$$\gamma_{s+e} = \frac{1}{||G_s(s)||_\infty}$$

where $||G_s(s)||_\infty$ is the L_2 norm of the slave+environment system. If the master impedance parameters are chosen as,

$$m_{m_i} > \frac{1}{\gamma_{s+e}}\sqrt{\frac{3}{2}},\ k_{m_i} = m_{m_i},\ b_{m_i} = \sqrt{2k_{m_i}m_{m_i}} \tag{3.70}$$

then the closed loop teleoperator system is L_2 stable independent of delay.

Proof. Since the transfer function matrix for the slave+environment system given by $G_s(s)$ has no pole in C_+ regardless of the values for K_e and B_e, it therefore belongs to the space H_∞ and has a finite H_∞ norm. As a result, an upper bound can always be found to satisfy Assumption 3.4.1.

As a consequence of the Small Gain Theorem [16], if the product of the L_2 gains of $G_m^{d_1}(s)$ and $G_s^{d_2}(s)$ can be made strictly less than one, then the closed loop system will be internally stable. Note that the H_∞ norm satisfies the sub-multiplicative property,

$$||GH||_\infty \leq ||G||_\infty ||H||_\infty$$

Therefore,

$$||G_m^{d_1}(s)||_\infty = ||G_m(s)e^{-T_1 s}||_\infty \leq ||G_m(s)||_\infty ||e^{-T_1 s}||_\infty$$

for the master system. Similarly, this is true for the slave system.

$$||G_s^{d_2}(s)||_\infty = ||G_s(s)e^{-T_2 s}||_\infty \leq ||G_s(s)||_\infty ||e^{-T_2 s}||_\infty$$

The H_∞ norm of a pure delay is one. As a result, one need only show that the product of the gains of $G_m(s)$ and $G_s(s)$ is less than unity in order to ensure internal stability independent of time delays.

The H_∞ norm of $G_m(s)$ may be expressed as,

$$||G_m(s)||_\infty = \sup_\omega \sqrt{\lambda_{\max}(G_m^T(-j\omega)G_m(j\omega))} \tag{3.71}$$

Since $G_m(s) \in C^{3n\times n}$ then $G_m^T(-j\omega)G_m(j\omega) \in C^{n\times n}$. The i^{th} diagonal of $G_m^T(-j\omega)G_m(j\omega) \in C^{n\times n}$ is found as,

$$\frac{\omega^4 + \omega^2 + 1}{m_{m_i}^2\omega^4 + (b_{m_i}^2 - 2m_{m_i}k_{m_i})\omega^2 + k_{m_i}^2} \tag{3.72}$$

The H_∞ norm of $G_m(s)$ is the supremum over ω of the square root of the maximum of (3.72). This is expressed as,

$$\begin{aligned} &||G_m(s)||_\infty = \\ &\sup_\omega \sqrt{\max_i \left(\frac{\omega^4 + \omega^2 + 1}{m_{m_i}^2\omega^4 + (b_{m_i}^2 - 2m_{m_i}k_{m_i})\omega^2 + k_{m_i}^2} \right)} \end{aligned} \tag{3.73}$$

Since $G_m(s)$ is a stable transfer function matrix, the H_∞ norm is finite. With complete control over the parameters of the system, the norm can be made as small as possible. Consider the following choices for the parameters,

$$k_{m_i} = m_{m_i},\ b_{m_i} = \sqrt{2k_{m_i}m_{m_i}} \tag{3.74}$$

With these choices,

$$||G_m(s)||_\infty = \sup_\omega \sqrt{\max_i \frac{1}{m_{m_i}^2}\left(\frac{\omega^4+\omega^2+1}{\omega^4+1}\right)} \tag{3.75}$$

Without loss of generality, assume that the j-th diagonal of $G_m^T(-j\omega)G_m(j\omega)$ is its largest eigenvalue, leading to the maximum in (3.75). Then define,

$$W(\omega) = \frac{1}{m_{m_j}^2}\frac{\omega^4+\omega^2+1}{\omega^4+1} \tag{3.76}$$

It is straightforward to analytically determine the maximum over ω of $W(\omega)$. The maximum of (3.76) will occur when its derivative is zero. It turns out that the derivative of (3.76) has roots at $\omega = 0$, $\omega = \pm 1$, and $\omega = \pm j$. Evaluating $W(\cdot)$ at each of these points yields,

$$W(0) = \frac{1}{m_{m_j}^2},\ W(\pm 1) = \frac{3}{2m_{m_j}^2},\ W(\pm j) = \frac{1}{2m_{m_j}^2} \tag{3.77}$$

Therefore, the maximum of $W(\omega)$ occurs at $\omega = \pm 1$. So for the choices of parameters given in (3.70) it can be said that,

$$||G_m(s)||_\infty = \sqrt{W(1)} = \frac{1}{m_{m_j}}\sqrt{\frac{3}{2}} \tag{3.78}$$

In order to satisfy the Small Gain Theorem, one must ensure that $||G_m(s)||_\infty \cdot ||G_s(s)||_\infty < 1$. Equivalently, $||G_m(s)||_\infty < \gamma_{s+e}$. From (3.78) one can express the small gain condition as,

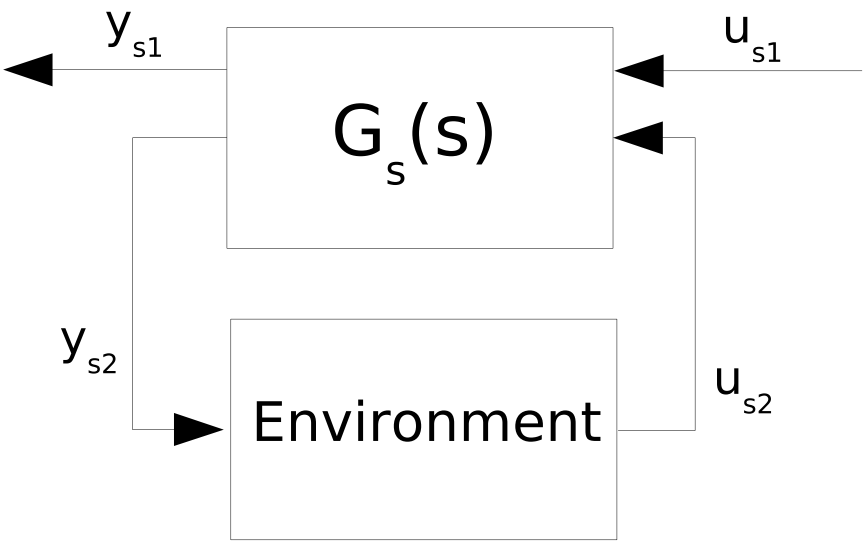


Figure 3.4: Block diagram of the slave and environment systems.

$$\frac{1}{m_{m_j}}\sqrt{\frac{3}{2}} < \gamma_{s+e} \tag{3.79}$$

Rearranging (3.79) leads to the condition on m_{m_i} in (3.70). Therefore, choosing the master impedance parameters as in (3.70) ensures that the master-slave closed loop teleoperator is stable for all time delays by the Small Gain Theorem. □

3.4.2 Nonlinear Slave Side Environment

In general, one may not be able to assume a known structure for the slave side environment. In fact, it may not even be possible to determine one. This section will show that, for a general nonlinear environment with a finite gain and a small modification to the slave closed loop plant, stability independent of delay for an unknown nonlinear environment may be achieved. The use of nonlinear slave side environments may be found in other works as well, such as [58, 59]. Consider Figure 3.4. This section will begin by deriving conditions under which the slave+environment loop is stable. From there, stability of the entire closed loop is examined.

The slave has two vector valued inputs. The first, $u_{s_1} \in R^{3n}$, is the delayed master trajectory. That is,

$$u_{s_1} = \begin{bmatrix} X_{m_1}^{d_1} \\ X_{m_2}^{d_1} \\ \dot{X}_{m_2}^{d_1} \end{bmatrix}$$

Then, $u_{s_2} \in R^n$ is the environmental force applied to the slave. So, $u_{s_2} = F_e$. The output $y_{s_1} \in R^n$ is the vector of environmental forces reflected back to the master. That is, $y_{s_1} = F_e$. The output $y_{s_2} \in R^{2n}$ is the slave state, which is the input to the environment operator. Defining $G_s : L_{2e}^{4n} \rightarrow L_{2e}^{3n}$ as the operator representing the slave system, one may express the slave system as,

$$\begin{bmatrix} y_{s_1} \\ y_{s_2} \end{bmatrix} = G_s \begin{bmatrix} u_{s_1} \\ u_{s_2} \end{bmatrix} \tag{3.80}$$

The whole input u_s to the slave is defined as $u_s = [u_{s_1}^T, u_{s_2}^T]^T \in R^{4n}$. Similarly, the output y_s is defined as $y_s = [y_{s_1}^T, y_{s_2}^T]^T$.

Defining the environment as some relation,

$$u_{s_2} = Hy_{s_2} \tag{3.81}$$

where $H : L_{2e}^{2n} \rightarrow L_{2e}^{n}$ is the mapping relating slave state (defined here as y_{s_2}) to environmental force (defined here as u_{s_2}), this subsystem must be finite-gain L_2 stable. That is, there must exist non-negative constants γ_e and β_e such that,

$$||(Hy_{s_2})_\tau||_{L_2} \leq \gamma_e ||(y_{s_2})_\tau||_{L_2} + \beta_e \tag{3.82}$$

for all $y_{s_2} \in L_{2e}^{2n}$ and $\tau \in [0, \infty)$.

One would like to be able to choose the slave plant L_2 gain arbitrarily small to ensure a loop gain of less than one. However, the H_∞ norm of the slave system cannot be made arbitrarily small without some modification. In order to overcome this, a scaling term k_p on the delayed master states is introduced. As well, the desired impedance parameters are re-defined. Define,

$$\tilde{M}_s = \epsilon_p \tilde{M}_s',\ \tilde{B}_s = \epsilon_p \tilde{B}_s',\ \tilde{K}_s = \epsilon_p \tilde{K}_s' \tag{3.83}$$

where ϵ_p is a positive scalar to be set by the designer and $\tilde{M}_s'$, $\tilde{B}_s'$, and $\tilde{K}_s'$ are positive diagonal matrices that may be freely chosen as well. Introducing these parameters has the effect of giving full control of the H_∞ norm of the slave to the designer while maintaining the desired dynamic behaviour. Introducing the scalar master trajectory scaling factor k_p and using the re-definitions of the impedance parameters, one can express the slave dynamics as,

$$\begin{bmatrix} \dot{X}_{s_1} \\ \dot{X}_{s_2} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -\tilde{M}_s'^{-1}\tilde{K}_s' & -\tilde{M}_s'^{-1}\tilde{B}_s' \end{bmatrix}}_{A_s} \begin{bmatrix} X_{s_1} \\ X_{s_2} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ k_p\tilde{M}_s'^{-1}\tilde{K}_s' & k_p\tilde{M}_s'^{-1}\tilde{B}_s' & k_p I & -\epsilon_p^{-1}\tilde{M}_s'^{-1} \end{bmatrix}}_{B_s} \begin{bmatrix} X_{m_1}^{d_1} \\ X_{m_2}^{d_1} \\ \dot{X}_{m_2}^{d_1} \\ F_e \end{bmatrix} \tag{3.84}$$

$$Y_s = \underbrace{\begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix}}_{C_s} \begin{bmatrix} X_{s_1} \\ X_{s_2} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & \epsilon_p^{-1} I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{D_s} \begin{bmatrix} X_{m_1}^{d_1} \\ X_{m_2}^{d_1} \\ \dot{X}_{m_2}^{d_1} \\ F_e \end{bmatrix} \tag{3.85}$$

Lemma 3.4.1. *Given the slave system (3.84) and (3.85) with the scaling factor k_p on the delayed master trajectory inputs, set $\tilde{K}_s = \epsilon_p \tilde{K}_s'$, $\tilde{B}_s = \epsilon_p \tilde{B}_s'$, and $\tilde{M}_s = \epsilon_p \tilde{M}_s'$. Then, the H_∞ norm of the slave system may be made arbitrarily small.*

Proof. One desires to be able to meet the condition,

$$||G_s(s)||_\infty < \epsilon_s$$

where ϵ_s is some arbitrarily small constant. So one requires,

$$||G_s(s)||_\infty = ||C_s(sI - A_s)^{-1}B_s + D_s||_\infty < \epsilon_s \tag{3.86}$$

It is possible to express B_s as,

$$\begin{aligned} B_s = k_p & \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ \tilde{M}_s'^{-1}\tilde{K}_s' & \tilde{M}_s'^{-1}\tilde{B}_s' & I & 0 \end{bmatrix}}_{B_1} \\ + \epsilon_p^{-1} & \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\tilde{M}_s'^{-1} \end{bmatrix}}_{B_2} \end{aligned} \tag{3.87}$$

As well, D_s may be expressed as,

$$D_s = \epsilon_p^{-1} \underbrace{\begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{D'} \tag{3.88}$$

Substituting the right hand sides of (3.87) and (3.88) into (3.86) and using the triangle inequality gives,

$$\begin{aligned} ||k_pC_s(sI - A_s)^{-1}B_1 + \epsilon_p^{-1}C_s(sI - A_s)^{-1}B_2 + D_s||_\infty \leq \\ \underbrace{k_p||C_s(sI - A_s)^{-1}B_1||_\infty}_{\alpha} + \underbrace{\epsilon_p^{-1}\left(||C_s(sI - A_s)^{-1}B_2||_\infty + ||D'||_\infty\right)}_{\beta} \end{aligned} \tag{3.89}$$

If it is possible to show that the right hand side of (3.89) is less than ϵ_s, it is then guaranteed that $||G_s(s)||_\infty < \epsilon_s$ as well. By adjusting k_p the term α may be decreased to an arbitrarily small value. As well, as ϵ_p approaches infinity, the term β may be made arbitrarily small. As a result, it is always possible to ensure,

$$||G_s(s)||_\infty < \epsilon_s \tag{3.90}$$

□

Theorem 3.4.3. *Consider the finite-gain stable environment operator (3.81), with known gain γ_e, connected in feedback with the slave subsystem (3.84) and (3.85). One can always ensure that this feedback connection is finite-gain stable by ensuring that the slave L_2 gain $\gamma_s < 1/\gamma_e$.*

Proof. By Lemma 3.4.1 one can choose slave parameters such that γ_s may be made arbitrarily small. The condition $\gamma_s < 1/\gamma_e$ may be equivalently expressed as,

$$\gamma_s \gamma_e < 1 \tag{3.91}$$

This is a small gain condition. When (3.91) is satisfied, the slave-environment loop is finite-gain L_2 stable by the Small Gain Theorem. □

The slave-environment closed loop gain is defined as γ_{s+e}. It has already been shown above that a suitable choice of master plant parameters may always be made to ensure the closed-loop stability of the master-slave system. In this case one can choose parameters to ensure that,

$$||G_m(s)||_\infty < \frac{1}{\gamma_{s+e}} \tag{3.92}$$

Then the feedback connection of the finite-gain L_2 stable master with the finite-gain L_2 stable slave+environment subsystem is guaranteed to be stable by the Small Gain Theorem.

3.5 Closed Loop Delay-Dependent Stability with a Linear Environment

While the results of the previous two subsections show that sufficient conditions may be found to stabilize the closed loop system for any delay, those choices may be overly conservative for small delays. This section makes use of a result in [92] to determine stability of the closed loop system for a particular range of delays. That is, for a given choice of the master and slave parameters, the maximum delay for which the system is stable may be found. With this result, the designer may determine if the particular parameters chosen will maintain stability for the delay considered in a particular case.

This result applies for linear systems with *commensurate* delays, that is the delays in the system must be integer multiples of a certain positive value τ. In a teleoperation system, this is achievable as it is always possible to put buffers in the system on each side so that incoming signals may be released to each of the master and slave at a known time interval. As a result, it is possible to assume that the delays from the master to the slave, and from the slave to the master, are equal. In this section it is assumed that the environment is a linear set of spring-dampers having the form (3.58).

The theorem from [92] to be presented is valid for linear retarded function differential equation (RFDE) systems of the form,

$$\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{m} A_k x(t - k\tau), \qquad \tau \geq 0 \tag{3.93}$$

where $x(t) \in R^n$ and $A_i \in R^{n \times n}$ for all $i = 0, \ldots, m$.

The main delay dependent stability result, found in [92], is now presented.

Theorem 3.5.1. *Suppose that the system (3.93) is stable at $\tau = 0$, and let $q = \text{rank}(A_m)$. Furthermore, define*

$$\bar{\tau}_i := \begin{cases} \min_{1\leq k\leq n} \frac{\theta_k^i}{\omega_k^i} & \textit{if } \lambda_i(G(j\omega_k^i), H(j\omega_k^i)) = \\ & e^{-j\theta_k^i} \textit{ for some } \omega_k^i \in (0,\infty), \\ & \theta_k^i \in [0, 2\pi] \\ \infty & \textit{if } \underline{\rho}(G(j\omega), H(j\omega)) > 1, \\ & \forall\omega \in (0,\infty) \end{cases}$$

where

$$G(s) := \begin{bmatrix} 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \\ -(sI - A_0) & A_1 & \cdots & A_{m-1} \end{bmatrix}$$

$$H(s) := \mathrm{diag}(I, \ldots, I, -A_m)$$

Then,

$$\bar{\tau} := \min_{1\leq i\leq q+n(m-1)} \bar{\tau}_i$$

The system (3.93) is stable for all $\tau \in [0, \bar{\tau})$, but becomes unstable at $\tau = \bar{\tau}$.

Note that the notation $\lambda_i(A, B)$ represents the i-th generalized eigenvalue of two square matrices A and B. As well,

$$\underline{\rho}(A, B) := \min\{|\lambda| \,|\, \det(A - \lambda B) = 0\}$$

This theorem requires a frequency sweeping test. Strictly speaking, the theorem requires the evaluation of generalized eigenvalues continuously between 0 rad/sec and ∞ rad/sec. In practice, a numerical approach is used [92]. The frequency axis is broken up into a grid, and at each grid point the generalized eigenvalues of $G(j\omega_k^i)$ and $H(j\omega_k^i)$ are computed. If $\underline{\rho}(G(j\omega), H(j\omega)) > 1$ for all ω, the system has a delay margin of ∞. If this is not the case, the computation results in pairs (ω_k^i, θ_k^i) that will provide estimates of the delay margin $\bar{\tau}_i$. The smallest $\bar{\tau}_i$ will be the delay margin for the system. The pairs are found in the cases where the generalized eigenvalues have magnitude one. At these points, the phase θ_k^i is found, since the generalized eigenvalues are complex in general.

To make use of Theorem 3.5.1 for the bilateral teleoperation system, the closed loop master and slave dynamics must be expressed as a set of RFDEs of the form (3.93). Substituting the environment dynamics into the slave dynamics and performing the change of variable (3.59) to shift the equilibrium of the slave+environment dynamics to the origin yields the following matrices for the RFDE (3.93)

$$A_0 = \begin{bmatrix} 0 & I & 0 & 0 \\ -\tilde{M}_s^{-1}(\tilde{K}_s + K_e) & -\tilde{M}_s^{-1}(\tilde{B}_s + B_e) & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -\tilde{M}_m^{-1}\tilde{K}_m & -\tilde{M}_m^{-1}\tilde{B}_m \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{M}_s^{-1}\tilde{K}_s - \tilde{M}_m^{-1}\tilde{K}_m & \tilde{M}_s^{-1}\tilde{B}_s - \tilde{M}_m^{-1}\tilde{B}_m \\ 0 & 0 & 0 & 0 \\ -\tilde{M}_m^{-1}K_e & -\tilde{M}_m^{-1}B_e & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\tilde{M}_m^{-1}K_e & -\tilde{M}_m^{-1}B_e & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where the state of the RFDE system is defined as $x(t) = [X'^T_{s_1}, X^T_{s_2}, X^T_{m_1}, X^T_{m_2}]^T$. For n-DOF master and slave plants, $A_i \in R^{4n \times 4n}$ for all $i = 0, \ldots, 2$. In this system, τ represents the delay from the master to the slave, and the same delay from the slave to the master. Some terms are delayed by 2τ because the environment force is sent from the slave to the master, and then back from the master to the slave. This accounts for the matrix A_2.

With these matrices defined, the delay dependent stability of the closed loop teleoperation system may be checked according to the conditions of Theorem 3.5.1.

3.5.1 Numerical Example

This section will examine the case of 6-DOF master and slave manipulators connected through time delays. After application of the control laws given in Section 3.3 the master and slave manipulators have closed loop dynamics given by (3.54), (3.55) and (3.52), (3.53) respectively. The slave is in contact with a 6-DOF linear spring-damper environment described by (3.58). The closed loop master and slave impedance parameters are chosen as,

$$\tilde{M}_m = \tilde{M}_s = I, \quad \tilde{B}_m = \tilde{B}_s = 4I, \quad \tilde{K}_m = \tilde{K}_s = 4I$$

The environment on the slave side, modeled as a set of spring-dampers on each end effector degree of freedom, has the following stiffness and damping matrices: $K_e = 150I$, $B_e = 0$, where $I \in R^{6\times 6}$ is the identity matrix and $0 \in R^{6\times 6}$ is the zero matrix. Matrices A_0, A_1, A_2, $G(s)$, and $H(s)$ are constructed according to the previous section. Checking the conditions of Theorem 3.5.1 yields information about the stability of this system. The top graph in Figure 3.5 shows a plot of $\underline{\rho}$ over the frequency range. It is clear that it is not greater than one for all ω, therefore there is a finite delay margin. Computing this delay margin according to Theorem 3.5.1 shows that the system will be stable for delays up to and including $\tau = 0.013571$ seconds in each direction, but will be unstable for delays beyond that value. The bottom plot in Figure 3.5 shows the position and velocity of the first master degree of freedom in response to a step input for a time delay of $\tau = 0.0122$ seconds.

Computing the H_∞ norm of both the master and slave+environment for this choice of parameters yields $||G_m(s)||_\infty = 1$ and $||G_s(s)||_\infty = 17.5891$. Clearly, the sufficient condition of the Small Gain Theorem is violated, but a less conservative result for small delays has been achieved with the result of Theorem 3.5.1.

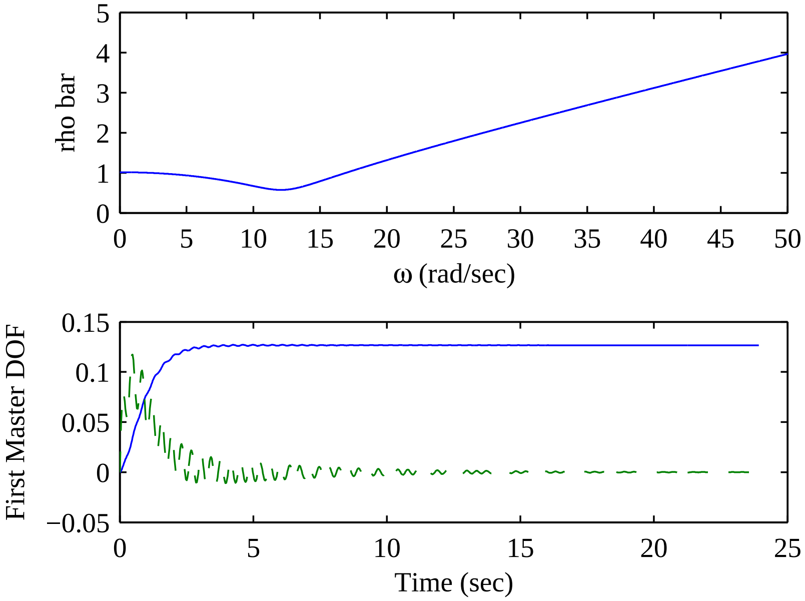


Figure 3.5: rho bar as a function of frequency for the numerical example (top plot). Position (solid) and velocity (dashed) of first master DOF in response to step input (bottom plot).

By contrast, one can make use of Theorem 3.4.2 to choose master impedance parameters that will ensure closed loop stability independent of delay. In this case, $\gamma_s = 0.0569$. Therefore, one must choose $m_{m_i} > 21.54216$. Choosing $\tilde{M}_m = 22I$ then yields $\tilde{K}_m = 22I$ and $\tilde{B}_m = 31.1127I$. Computing the H_∞ norm of the master for this choice of parameters yields $||G_m(s)||_\infty = 0.0557$. Then,

$$||G_m(s)||_\infty \cdot ||G_s(s)||_\infty = 0.9792 < 1$$

Clearly the closed loop system is stable by the Small Gain Theorem for this choice of parameters, leading to stability independent of delay. However, note that the master parameters needed to be chosen much larger than the slave parameters. This could lead to a system that, while stable independent of delays, may not perform as desirably as possible. A simulation showing the validity of the delay-dependent stability results with the entire teleoperation algorithm is given in Section 4.1.3.

3.6 Closed Loop Stability with the Operator Modeled as Nonlinear Dynamics

Just as the slave environment has been modeled as a dynamical system, it can be instructive to perform an analysis in which the human operator is also modeled as a dynamical system. It is often the case in bilateral teleoperation to assume that both the human operator and the environment are passive [5, 8, 89]. It is the view of the author that this is an unfair assumption. Humans have the ability to add energy to a system. As a result, assuming that a human will not add energy to a system is simplistic and does not consider the reality of the use of teleoperators. The same may be said for the slave side environment. It is possible that the environment may not be passive, especially if it is a human that is on the slave side as well. In light of this, the analysis carried out here makes no assumptions on the passivity of the human or environment.

As in the previous sections, the Small Gain Theorem will be used to show stability. While this is a conservative stability result, it is not restrictive in terms of the assumptions placed on the human or the environment. The only requirement on these two subsystems is that they be finite-gain stable with a known upper bound on the gain.

Consider the subsystem shown in Figure 3.6, where $y_{m_1} = [X_{m_1}^T, X_{m_2}^T]^T$ and $y_{m_2} = \dot{X}_{m_2}$. That is, the human receives as input the master manipulator state. The output from the human is the force $F_h \in R^n$. The signal $F_e^{d_2} \in R^n$ is the delayed environmental force coming from the slave side. This section will show that, as long as the human is a finite-gain stable operator, closed loop impedance parameters for the master manipulator may always be found in order to stabilize the master-human loop.

Define the human operator as a relation,

$$F_h = H_m y_{m_1} \tag{3.94}$$

where $H_m : L_{2e}^{2n} \rightarrow L_{2e}^n$ is the mapping relating the master state to the human force. This subsystem is finite-gain L_2 stable with gain γ_h. The master closed loop dynamics are

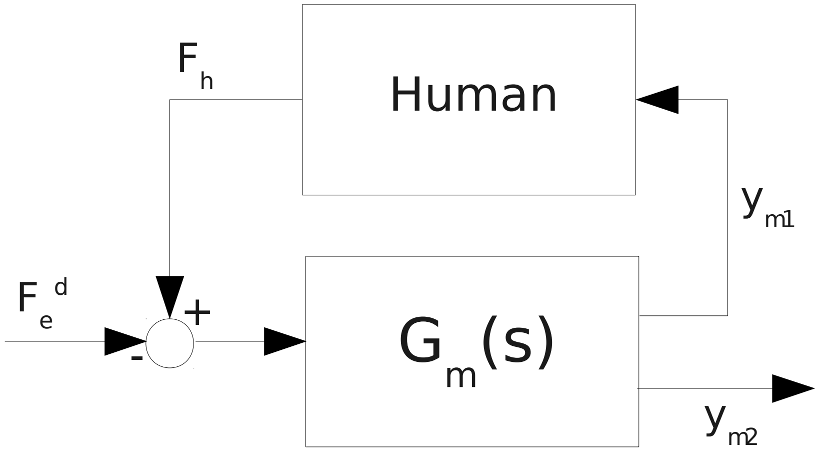


Figure 3.6: Block diagram of the interconnection between the human operator and the master plant.

expressed as in (3.60) and (3.61). This system has a transfer function representation given by (3.65).

It was shown in Section 3.4 that the H_∞ norm of the master system is expressed as,

$$\begin{aligned} ||G_m(s)||_\infty &= \sup_\omega \sqrt{\lambda_{\max}(G_m^T(-j\omega)G_m(j\omega))} \\ &= \sup_\omega \sqrt{\max_i \left(\frac{\omega^4+\omega^2+1}{m_{m_i}^2\omega^4+(b_{m_i}^2-2m_{m_i}k_{m_i})\omega^2+k_{m_i}^2} \right)} \end{aligned} \tag{3.95}$$

Without loss of generality, assume that the j-th diagonal of $G_m^T(-j\omega)G_m(j\omega)$ is its largest eigenvalue. Then, the H_∞ norm of $G_m(s)$ may be expressed as,

$$||G_m(s)||_\infty = \sup_\omega \sqrt{\frac{\omega^4+\omega^2+1}{m_{m_j}^2\omega^4+(b_{m_j}^2-2m_{m_j}k_{m_j})\omega^2+k_{m_j}^2}} \tag{3.96}$$

With this result, the following theorem can be stated.

Theorem 3.6.1. *Consider the finite-gain stable human operator (3.94), with known gain γ_h, connected in feedback with the master subsystem (3.60) and (3.61). One can always* *ensure that this feedback connection is finite-gain stable by ensuring that the master L_2 gain $\gamma_m < 1/\gamma_h$.*

Proof. Consider (3.96). It is clear that by increasing any one of m_{m_j}, b_{m_j}, or k_{m_j} individually, it is possible to make $||G_m(s)||_\infty = \gamma_m$ as small as desired. As a result, since γ_h is known and bounded, one may always be able to choose the master closed loop gain such that $\gamma_m < 1/\gamma_h$. Rearranging this expression leads to,

$$\gamma_m \gamma_h < 1$$

Then, when this condition is satisfied, the closed loop system consisting of the master manipulator and the human operator is finite-gain L_2 stable by the Small Gain Theorem. □

The master+human closed loop gain is defined as γ_{m+h}. The sufficient stability result of this section has shown that it is always possible to ensure that $\gamma_{m+h} < 1$. Results from Section 3.4.2 have shown that the gain of the slave+environment system, γ_{s+e}, can be made less than one, under suitable modifications to the slave closed loop plant. As a result, for a human operator and slave side environment modeled as nonlinear relations, it is always possible to ensure that the entire closed loop is finite-gain L_2 stable by the Small Gain Theorem.

3.7 Closed Loop Transparency Analysis

Since the closed loop teleoperation system presented here results in LTI systems for each of the master and slave manipulators, it is possible to derive the hybrid matrix for this system. Recall from Section 2.1 that the hybrid matrix is defined as,

$$\begin{bmatrix} F_h(s) \\ V_s(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} V_m(s) \\ -F_e(s) \end{bmatrix}$$

Since each DOF of the closed loop system is linear and decoupled, it suffices to derive the hybrid matrix for the i-th degree of freedom. The equation of dynamics for the i-th degree of freedom of the master manipulator in the closed loop is given as,

$$\tilde{M}_{m_i}\ddot{X}_{m_{1i}} + \tilde{B}_{m_i}\dot{X}_{m_{1i}} + \tilde{K}_{m_i}X_{m_{1i}} = F_{h_i} - F_{e_i}^{d_2} \tag{3.97}$$

where the subscript i denotes the i-th diagonal of the diagonal matrices or the i-th component of the vector X_{m_1}. Similarly, the equation of dynamics for the i-th degree of freedom of the slave manipulator tracking error dynamics is expressed as,

$$\tilde{M}_{s_i}\ddot{e}_{r_{1i}} + \tilde{B}_{s_i}\dot{e}_{r_{1i}} + \tilde{K}_{s_i}e_{r_{1i}} = -F_{e_i} \tag{3.98}$$

The hybrid matrix relies on a frequency domain model of the dynamics. Taking the Laplace transform of (3.97) and (3.98) yields,

$$V_{m_i}(s)\left(\tilde{M}_{m_i}s + \tilde{B}_{m_i} + \frac{\tilde{K}_{m_i}}{s}\right) = F_{h_i}(s) - F_{e_i}(s)e^{-T_2 s} \tag{3.99}$$

$$V_{e_i}(s)\left(\tilde{M}_{s_i}s + \tilde{B}_{s_i} + \frac{\tilde{K}_{s_i}}{s}\right) = -F_{e_i}(s) \tag{3.100}$$

where $V_{m_i}(s) = sX_{m_{1i}}(s)$ and $V_{e_i}(s) = se_{r_{1i}}(s)$. Defining $V_{s_i}(s) = sX_{s_{1i}}$ and noting that $e_{r_{1i}} = X_{s_{1i}} - X_{m_{1i}}^{d_1}$, one arrives at $V_{e_i}(s) = V_{s_i}(s) - e^{-T_1 s}V_{m_i}(s)$. Substituting this last expression into (3.100) and rearranging for $V_{s_i}(s)$ yields,

$$V_{s_i}(s) = e^{-T_1 s}V_{m_i}(s) - \frac{s}{\tilde{M}_{s_i}s^2 + \tilde{B}_{s_i}s + \tilde{K}_{s_i}}F_{e_i}(s) \tag{3.101}$$

Rearranging (3.99) for $F_{h_i}(s)$ gives,

$$F_{h_i}(s) = \left(\tilde{M}_{m_i}s + \tilde{B}_{m_i} + \frac{\tilde{K}_{m_i}}{s}\right)V_{m_i}(s) + e^{-T_2 s}F_{e_i}(s) \tag{3.102}$$

Combining (3.101) and (3.102) into a matrix equation, one arrives at the following hybrid matrix equation,

$$\begin{bmatrix} F_{h_i}(s) \\ V_{s_i}(s) \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{M}_{m_i}s + \tilde{B}_{m_i} + \frac{\tilde{K}_{m_i}}{s} & -e^{-T_2 s} \\ e^{-T_1 s} & \frac{s}{\tilde{M}_{s_i}s^2+\tilde{B}_{s_i}s+\tilde{K}_{s_i}} \end{bmatrix}}_{H(s)} \begin{bmatrix} V_{m_i}(s) \\ -F_{e_i}(s) \end{bmatrix} \qquad (3.103)$$

Recall that the ideal hybrid matrix for transparent teleoperation requires that $H_{11}(s) = 0$, $H_{12}(s) = -e^{-T_2 s}$, $H_{21}(s) = e^{-T_1 s}$, and $H_{22}(s) = 0$. From (3.103), entries $H_{12}(s)$ and $H_{21}(s)$ are exactly the ideal values. The physical interpretation of this is that perfect force tracking and perfect free motion tracking of the delayed master trajectory by the slave are achieved with this algorithm. The non-zero entry for $H_{11}(s)$ shows that there is some input impedance at the master side.[1] This effectively means that the operator will feel the dynamics of the master manipulator. This result is expected, as the master controller is designed to give the master manipulator a particular closed loop impedance. In fact, the closed loop stability results rely on having a non-zero impedance for the master manipulator. Anderson and Spong [5] show, using scattering theory, that the norm of the scattering matrix corresponding to the hybrid matrix for the ideal time-delayed system is unbounded. This result means that the ideal time-delayed teleoperator with hybrid matrix corresponding to (2.3) cannot be stabilized for all passive operators and environments. Therefore, the ideal time-delayed teleoperator is not robustly stable [5].

Niemeyer and Slotine address the issue of sensing the dynamics of the teleoperator in [9]. They mention that, in the presence of time delays, it is not possible to completely remove the dynamics of the teleoperator from the operator's perception. Instead, they propose designing the teleoperator system, including the dynamics present, as a virtual tool. This perspective fits well within the context of time delayed systems with a non-ideal hybrid matrix. By designing the teleoperator as a tool with dynamics that the operator

[1] Note that this term is expressed in the standard impedance model form [90]. In this form, the dynamics are expressed as the ratio of force to velocity, leading to a system that approaches infinity as the frequency of the input velocity tends to zero when there is a spring in the dynamics.

will learn to use, the limitations imposed by the time delay are directly addressed and not necessarily seen as negative.

The non-zero entry for $H_{22}(s)$ in (3.103) is also an expected result. This term is an admittance between slave environment force and slave motion when the master manipulator is not in motion. The value for the ideal hybrid matrix is zero, which suggests that when the master is not in motion, any environment forces acting on the slave should have no effect on its motion. However, the design decision was made when designing the slave sliding surface to allow the environmental force to drive the slave tracking error dynamics. Allowing the slave manipulator to have some compliance in this way was done in order to prevent large reaction forces at the slave when contacting a stiff environment. As well, from a practical point of view, it would be possible for a large environment force to saturate the slave controller if part of the control objective were to have the slave remain immobile when the master is immobile, regardless of the environmental forces.

Of particular note in this transparency analysis is the achievement of the goal of perfect force tracking. Up until now, only algorithms with force sensors were able to achieve this [13]. While the proposed algorithm does fall under the position-force type of architectures in terms of its behaviour, it requires only the physical setup of a position-position architecture, but offers the benefit of a position-force architecture.

3.7.1 Performance Comparison with Other Teleoperation Architectures

Since transparency is a very common measure of performance in teleoperation, it is a suitable metric for comparing the performance of various algorithms. The task of performing a comprehensive numerical comparison of several algorithms would be quite involved, due to the number of parameters that may be varied from one architecture to the next.

The hybrid matrix derived for the algorithm in this work has the same form as that developed by Cho *et al.* [89]. This is due to the fact that the closed loop dynamics for each of the master and slave have the same form in both cases. As a result, one could expect similar performance between the two approaches. (Note that the algorithm in [89] is only presented for linear 1-DOF systems.) As well, the algorithm of Cho *et al.* requires force sensors at both the master and slave sides, as well as velocity sensors, in order to achieve this transparency result. Whereas, the approach presented here requires only position sensors.

In [13], Tavakoli *et al.* derive the hybrid matrix representations for several teleoperation architectures. Unfortunately, time delays are not considered in their work. However, the similarities between the structure of the hybrid matrix derived here and those of other architectures can still be discussed. For the standard non-delayed Position Error Based (PEB) architecture, the hybrid matrix is given by,

$$H(s) = \begin{bmatrix} Z_m + C_m \frac{Z_s}{Z_{ts}} & -\frac{C_m}{Z_{tm}} \\ \frac{C_s}{Z_{ts}} & \frac{1}{Z_{ts}} \end{bmatrix} \tag{3.104}$$

where $Z_m(s) = M_m s$ and $Z_s(s) = M_s s$ represent the dynamics of the master and slave robots respectively, $C_m = (k_{v_m} s + k_{p_m})/s$ and $C_s = (k_{v_s} s + k_{p_s})/s$ are the PD controllers used at each of the master and slave sides, and $Z_{tm} = Z_m + C_m$ and $Z_{ts} = Z_s + C_s$. It is clear from the entry $H_{12}(s)$ that this method does not give ideal force tracking. This has typically been the case for methods that do not make force measurements. Although, ideal delayed force tracking is achieved by the new algorithm presented in this work with unknown input sliding mode observers. The entry $H_{21}(s)$ is also not the ideal value. This shows that for this method, in general, ideal free motion tracking is not achieved either. As in the case with the method proposed in this work, the operator feels some dynamics when using the teleoperator system even in free motion. With a proper choice of master controller, the sensation of these dynamics could be adjusted. The PEB approach also has some admittance between the slave and environment, as seen in the term $H_{22}(s)$.

Tavakoli *et al.* also present the hybrid matrix for the Direct Force Reflection (DFR) architecture [13]. This method is a position-force architecture. A force sensor is required at the slave side, and the forces measured by it are transmitted directly back to the master. In this case, there is no controller at the master side. A local PD controller at the slave side is used. The hybrid matrix for the DFR architecture is given as,

$$H(s) = \left[\begin{array}{cc} Z_m & -1 \\ \frac{C_s}{Z_{ts}} & \frac{1}{Z_{ts}} \end{array} \right] \tag{3.105}$$

where the parameters of the hybrid matrix are defined as above. The use of the slave side force sensor provides ideal force tracking. The input impedance at the master side and admittance term between the slave and environment remain present in this architecture. Once again, ideal free motion tracking of the master trajectory by the slave is not achieved with this architecture. This is due to the $H_{21}(s)$ term being non-ideal.

Hashtrudi-Zaad and Salcudean [93] show that, using a four channel teleoperation architecture, significant time delays lead to the loss of stability and transparency. While in the non-delayed case, a four channel architecture can provide ideal transparency, this is not the case for delayed systems. However, the authors of [93] present two three channel architectures that, under suitable conditions, can lead to ideally transparent systems in the presence of time delays. For brevity, only the second approach will be discussed. The interested reader may refer to [93] for further details. The second of these two control architectures is known as the Environment-Force-Compensated (EFC) architecture. This approach requires unity local force feedback at the slave side, thus the need to use a force sensor. The slave velocity and environment force signals are transmitted to the master, but only the human force is transmitted back to the slave. In this architecture, it can be shown [93] that the hybrid matrix has the form,

$$H(s) = \left[\begin{array}{cc} \frac{Z_{tm}}{C_2}\left(1 - e^{-2T_s}\right) & -e^{-T_s} \\ e^{-T_s} & 0 \end{array} \right] \tag{3.106}$$

where the term $C_2(s)$ is a feedforward control parameter that the slave environment force passes through on the way to the master side. With this control architecture, choosing $C_2(s) \gg 1$ ensures that the term $H_{11}(s)$ approaches the ideal value. In [93], a constant time delay of T_s seconds in each direction is assumed.

It is clear that the EFC architecture provides very close to ideal transparency for a time delayed teleoperator. However, following Llewellyn's absolute stability criterion, it can be shown [93] that one cannot guarantee stability of this architecture for all passive operators and environments. This is a fundamental issue in ideally transparent time delayed teleoperators. Hashtrudi-Zaad and Salcudean develop certain stability conditions in terms of the closed loop impedance parameters and human and environment impedances such that a necessary and sufficient stability condition set is available. That is, depending on the human operator and environment impedances, one may be able to choose closed loop parameters to stabilize the system. However, one may need to choose C_2 in such a way that ideal transparency is lost.

This comparison has shown that trade offs must always be made in designing a teleoperator. One may choose to use a simpler hardware setup, as in the case of the PEB method, but this is often at the cost of performance. The PEB architecture is straightforward to implement, and is based on PD controllers, but the performance from a transparency point of view is poorer than other approaches. The DFR architecture provides increased performance over the PEB method, but requires force sensing. As well, there remain several non-ideal terms in the hybrid matrix. The three channel EFC architecture provides an approach that is much closer to the ideal delayed transparency matrix. This approach also requires more complicated hardware, in that force sensors are required. As well, the cost associated with the increased transparency comes in terms of having a system that is not necessarily stable for some combinations of even passive humans and environments. Choosing stable closed loop parameters for particular combinations of environments may result in a system that is further from the ideal hybrid matrix than may be desired.

The approach taken in the algorithm in this work has been to ensure ideal delayed force tracking of the slave force by the master, and ideal delayed position tracking of the master by the slave. These goals correspond to entries $H_{12}(s)$ and $H_{21}(s)$ in the hybrid matrix, which are ideal for this algorithm. Entries $H_{11}(s)$ and $H_{22}(s)$ are not ideal in this work, however this is not necessarily a negative aspect of the approach. The notion of a virtual tool, as put forward by Niemeyer and Slotine [9], used by the operator to interact with the environment clearly has merit. In many non-teleoperated tasks, an operator makes use of a real tool that has its own dynamics to accomplish the task. Extending this concept to virtual teleoperated tasks has merit. As well, Cho *et al.* [89] found that without some local compliance between the slave and environment, slave contact forces were quite high. An earlier paper by Park and Cho [66] used a slave sliding mode controller without any local compliance. The results of that work prompted further research into architectures that allow compliance. So again, a non-ideal value in the term $H_{22}(s)$ can be useful from an implementation perspective.

Based on this comparison, it is clear that the algorithm presented in this thesis provides good force and position tracking, while making other decisions that deviate from ideal transparency. The benefits of of these deviations can be seen in terms of stability properties. This work has shown that this algorithm may be stabilized for any finite-gain stable operator and environment. This represents a less restrictive assumption on the properties of the human and environment in terms of passivity. As well, in practice, the admittance at the slave-environment interface allows for contact that does not generate large reaction forces, potentially destabilizing the system in the contact transition.

3.8 Robustness to Unmodeled Dynamics

A well known issue with feedback linearizing and computed torque method controllers is the issue of robustness to unmodeled dynamics. Since these controllers rely on an exact mathematical cancellation of the nonlinearities in the plant, a perfect cancellation of the nonlinearities would require a perfect model. In practice, this is never feasible.

Consider the equations for the master manipulator dynamics (3.11), (3.12) expressed in task space. In practice, the terms $\bar{M}_m(X_m)$ and $\bar{h}_m(X_m, \dot{X}_m)$ are not known exactly. These terms can be expressed as,

$$\bar{M}_m(X_m) = \hat{\bar{M}}_m(X_m) - \Delta\bar{M}_m \tag{3.107}$$

$$\bar{h}_m(X_m) = \hat{\bar{h}}_m(X_m, \dot{X}_m) - \Delta\bar{h}_m \tag{3.108}$$

where $\hat{\bar{M}}_m(X_m)$ and $\hat{\bar{h}}_m(X_m, \dot{X}_m)$ are the approximate models of $\bar{M}_m(X_m)$ and $\bar{h}_m(X_m)$, and $\Delta\bar{M}_m$ and $\Delta\bar{h}_m$ represent the modeling error. In this case, the master observer is actually implemented as,

$$
\begin{aligned}
\dot{\hat{X}}_{m_1} &= \hat{X}_{m_2} + z_{m_1} && (3.109)\\
\dot{\hat{X}}_{m_2} &= \hat{\bar{M}}_m^{-1}(\hat{X}_{m_1})\left(-\hat{\bar{h}}_m(\hat{X}_{m_1}, \hat{X}_{m_2}) + F_m\right) + z_{m_2} && (3.110)
\end{aligned}
$$

Substituting the right hand sides of (3.107) and (3.108) into the master manipulator dynamics in task space and simplifying, one arrives at,

$$
\begin{aligned}
\dot{X}_{m_1} &= X_{m_2} && (3.111)\\
\dot{X}_{m_2} &= \hat{\bar{M}}_m^{-1}(X_{m_1})\left(-\hat{\bar{h}}_m(X_{m_1}, X_{m_2}) + F_m\right)\\
&\quad +\hat{\bar{M}}_m^{-1}(X_{m_1})\left(\Delta\bar{M}_m\dot{X}_{m_2} + \Delta\bar{h}_m + F_h\right) && (3.112)
\end{aligned}
$$

Provided that the unmodeled dynamics are bounded over all time, the observer states will still converge to the true states in finite time for a suitable choice of the observer gains. At that point, the observer velocity error dynamics may be expressed as,

$$
\dot{e}_{m_2} = \hat{\bar{M}}_m^{-1}(X_{m_1})\left(\Delta\bar{M}_m\dot{X}_{m_2} + \Delta\bar{h}_m + F_h\right) - z_{m_2} = 0 \qquad (3.113)
$$

where $e_{m_2} = X_{m_2} - \hat{X}_{m_2}$. Then, the unknown input is recovered from the observer as,

$$
\underbrace{F_h + \Delta\bar{M}_m\dot{X}_{m_2} + \Delta\bar{h}_m}_{\hat{F}_h} = \hat{\bar{M}}_m(X_{m_1})z_{m_{2eq}} \qquad (3.114)
$$

Therefore, the observer estimates the sum of the external force and some modeling error terms in the case where modeling error is present in the plant. It is not possible to separate out these terms. The master outer loop controller, expressed in terms of the approximate dynamic model of the plant, is given as,

$$F_m = \hat{\bar{M}}_m(\hat{X}_{m_1})v_m + \hat{\bar{h}}_m(\hat{X}_{m_1}, \hat{X}_{m_2}) - \hat{F}_h \tag{3.115}$$

Applying this controller to the master plant (3.111), (3.112) and simplifying yields,

$$\dot{X}_{m_1} = X_{m_2} \tag{3.116}$$

$$\dot{X}_{m_2} = v_m \tag{3.117}$$

This is a noteworthy result since, despite the presence of unmodeled dynamics in the plant, the unknown input estimate from the observer is able to estimate and cancel these terms. (Provided that the unmodeled dynamics are bounded). Therefore, in a situation where there is plant uncertainty, one could make use of an unknown input observer to estimate the unmodeled dynamics and use the estimate to cancel the unmodeled terms. This would ensure that the closed loop system does become a chain of integrators despite modeling error. However, in the current situation, the inner control law v_m is an impedance controller and reintroduces the unmodeled terms. Applying the inner loop control law (3.24) to the above double integrator system, making use of the unknown input estimate (3.114), yields,

$$\dot{X}_{m_1} = X_{m_2} \tag{3.118}$$

$$\dot{X}_{m_2} = -\tilde{M}_m^{-1}\tilde{B}_m X_{m_2} - \tilde{M}_m^{-1}\tilde{K}_m X_{m_1} + \tilde{M}_m^{-1}F_h - \tilde{M}_m^{-1}\hat{F}_e^{d_2} + \tilde{M}_m^{-1}\left(\Delta\bar{M}_m\dot{X}_{m_2} + \Delta\bar{h}_m\right) \tag{3.119}$$

The last term in (3.119) contains the unmodeled dynamics. In general, the form of these terms is unknown. Therefore, it is not possible to conclude anything regarding the stability of this system in the presence of modeling error.

Another complication comes from the fact that the estimate of the unmodeled dynamics generated by the observer is transmitted to the slave side. It is not possible to conclude anything about the stability of the slave system when driven with these unmodeled master dynamics. This analysis has only considered unmodeled dynamics on the master side, but similar results can be found when the analysis is done on the slave side. In this case, the slave unmodeled dynamics are transmitted back to the master manipulator, and then again back to the slave side through the term $\hat{F}_e^{dd}$. This additional feedback would further complicate a stability analysis. While it is not possible to form conclusions about the stability of the entire system in the presence of unmodeled dynamics, in practice it appears that with a well-modeled plant, these terms will be quite small and do not appear to cause problems in terms of the stability of the system.

Chapter 4

Simulation and Experimental Results

While the previous chapter presented detailed theoretical results about the stability of the proposed algorithm under a variety of situations, it remains important to evaluate the algorithm in practice. This chapter will examine the stability and performance of the algorithm through both numerical simulations and an experimental implementation. Simulation results for 2-DOF nonlinear manipulators are examined for situations involving both free and constrained motion in the presence of time delays. The experimental implementation is performed on the University of Waterloo Teleoperation Platform. The experimental results show the effectiveness of the algorithm in a bilateral teleoperation setup for various time delays in the forward and backward paths. In practice, these time delays can be largely attributed to the communications. See Figure 3.1. Other real world factors are addressed in the experimental implementation. The sliding mode controller is implemented with a boundary layer in order to mitigate the effect of chattering. As well, the performance of the experiments with the the inevitable presence of modeling error in the robot dynamic models serves to show robustness of the approach to unmodeled dynamics.

4.1 Numerical Simulations

The performance of the algorithm presented in this work is now examined through a simulation study. In this case, the master and slave manipulators are chosen as 2-DOF nonlinear serial link manipulators. The master and slave dynamics are given by (3.1) and (3.2) respectively.

The dynamics of each manipulator have the same structure. To simplify notation, define the inertial parameters as $\theta_{i_1} = m_{i_1} l_i^2$ and $\theta_{i_2} = m_{i_2} l_i^2$, $i \in \{m, s\}$. Then m_{i_1} and m_{i_2} represent the masses of the first and second link, and l_i represents the length of each link. The mass and Coriolis matrices may be expressed as,

$$M_i(q_i) = \begin{bmatrix} \theta_{i_1} + 2\theta_{i_2} + 2\theta_{i_2}\cos q_{i_2} & \theta_{i_2} + \theta_{i_2}\cos q_{i_2} \\ \theta_{i_2} + \theta_{i_2}\cos q_{i_2} & \theta_{i_2} \end{bmatrix} \tag{4.1}$$

$$C_i(q_i, \dot{q}_i) = \begin{bmatrix} -2\theta_{i_2}\dot{q}_{i_2}\sin q_{i_2} & -\theta_{i_2}\dot{q}_{i_2}\sin q_{i_2} \\ \theta_{i_2}\dot{q}_{i_1}\sin q_{i_2} & 0 \end{bmatrix} \tag{4.2}$$

The forward kinematics for each manipulator are as follows,

$$X_{i_1} = f_i(q_i) = \begin{bmatrix} l_i\cos(q_{i_1}) + l_i\cos(q_{i_1} + q_{i_2}) \\ l_i\sin(q_{i_1}) + l_i\sin(q_{i_1} + q_{i_2}) \end{bmatrix} \tag{4.3}$$

The manipulator Jacobian is given as,

$$J_i(q_i) = \begin{bmatrix} -l_i\sin q_{i_1} - l_i\sin(q_{i_1} + q_{i_2}) & -l_i\sin(q_{i_1} + q_{i_2}) \\ l_i\cos q_{i_1} + l_i\cos(q_{i_1} + q_{i_2}) & l_i\cos(q_{i_1} + q_{i_2}) \end{bmatrix} \tag{4.4}$$

Note that the task space expressions of the manipulator dynamics are computed online in the simulation. The master controller (3.23), (3.24) and master observer (3.15), (3.16) are applied to the master plant. Similarly, the slave controller (3.32) and slave observer

(3.20), (3.21) are applied to the slave plant. Both the master and slave have the same dynamic parameters of $m_{m_1} = m_{m_2} = m_{s_1} = m_{s_2} = 1$ kg for the mass and $l_m = l_s = 1$ m for the length of the links.

The sliding mode observer gains are set to $\lambda_{m_1} = \lambda_{m_2} = 7.5$, $\alpha_{m_1} = \alpha_{m_2} = 11$, $\lambda_{s_1} = \lambda_{s_2} = 3$, and $\alpha_{s_1} = \alpha_{s_2} = 16.5$ in all simulations. These values were determined through simulation to give good results. The slave sliding mode controller gain is chosen as $K_g = 20I$, which is large enough to ensure that a sliding mode occurs. Note that, in the slave controller, the pure switching function is replaced with a saturation function to introduce a boundary layer where switching does not occur. The width of the boundary layer is chosen as 10^{-2}. This value was determined numerically to yield a control signal that does not cause high frequency switching. This minimizes the effects of chattering in the system. Note, however, that pure switching terms are used in the observers. The master and slave force estimates are obtained with first order low pass filters that each have a bandwidth of 30 Hz. A filter of this bandwidth will attenuate the high frequency switching signals while allowing the lower frequency equivalent signals to pass through. The human force applied to each master end-effector degree of freedom is a square wave of amplitude 5 N passed through a first order pre-filter with a pole at $s = -5$.

Three simulations are performed. The first shows the teleoperation system when the slave is in free motion. The second simulation shows the situation with the slave in contact with an environment. The third simulation shows the case where the delay-dependent stability results of Section 3.5 are used to determine a delay margin for a particular choice of impedance parameters. In all simulations, the master and slave initial conditions in the joint space are set as $q_m = [-0.5, 2.5]^T$, $\dot{q}_m = [0, 0]^T$, and $q_s = [-0.55, 2.53]^T$, $\dot{q}_s = [0, 0]^T$. The observers are initialized as $\hat{X}_m = [0.4614, 0.4299, 0, 0]^T$ and $\hat{X}_s = [0.4546, 0.3948, 0, 0]^T$. Note that the initial conditions on the observers are the task space positions corresponding to the robot joint angles, and velocities of zero. This is a very reasonable assumption, as it is always possible to obtain initial positions from the position sensors. The system was simulated in Matlab using a fixed step 5th order Runge-Kutta algorithm with a sample period of $T_s = 10^{-5}$ seconds.

Note that the end effector state estimates are a function of the joint angle estimates, but the observers are designed in the end effector space. In order to implement the observers, the inverse kinematics of each manipulator must be known. The manipulator dynamics in task space are actually functions of the manipulator joint angles and velocities. So in order to correctly build up the dynamic expressions in task space, estimates of the joint space positions and velocities are required. The inverse kinematics are used here to compute the joint angle estimates from the end effector state estimates. The resulting joint angle estimates are then used to build up the task space dynamics. However, note that the observers estimate the task space states directly.

4.1.1 Simulation Results with Slave in Free Motion

Initially, a simulation where the slave is not in contact with an environment is performed. A time delay of 0.25 seconds in each direction is used, giving a round trip time delay of 0.5 seconds. In this case, one expects the slave to asymptotically track the delayed master trajectory. In this simulation, the desired impedance characteristics for each manipulator are chosen as, $\tilde{M}_m = 22I$, $\tilde{B}_m = 32I$, $\tilde{K}_m = 22I$, $\tilde{M}_s = I$, $\tilde{B}_s = 4I$, and $\tilde{K}_s = 4I$. Figure 4.1 shows the master and slave end effector trajectories. The time delay is evident in this plot. It is also clear that the slave manipulator is tracking the delayed version of the master trajectory, as desired. Figure 4.2 shows the slave tracking error. There is a jump in the error at 0.25 seconds due to the time delay. For the first 0.25 seconds the slave receives a zero valued signal as the master "trajectory" since the data from the master side has not yet reached the slave side. Once the first bit of master data arrives at the slave, there is a jump in the error since there has effectively been a jump in the delayed master trajectory. After this jump, asymptotic tracking occurs.

Both the master and slave observers perform properly, as expected. Figure 4.3 shows a plot of the slave state values and estimates during the simulation. Since the observer is initialized with the same values as the actual states, observer convergence has already occurred at the beginning of the simulation. Note that the observer must be initialized such that the position estimates are exactly the position measurements, and the velocity estimate is zero. Since the plants start from rest, the initial velocity is also zero. The

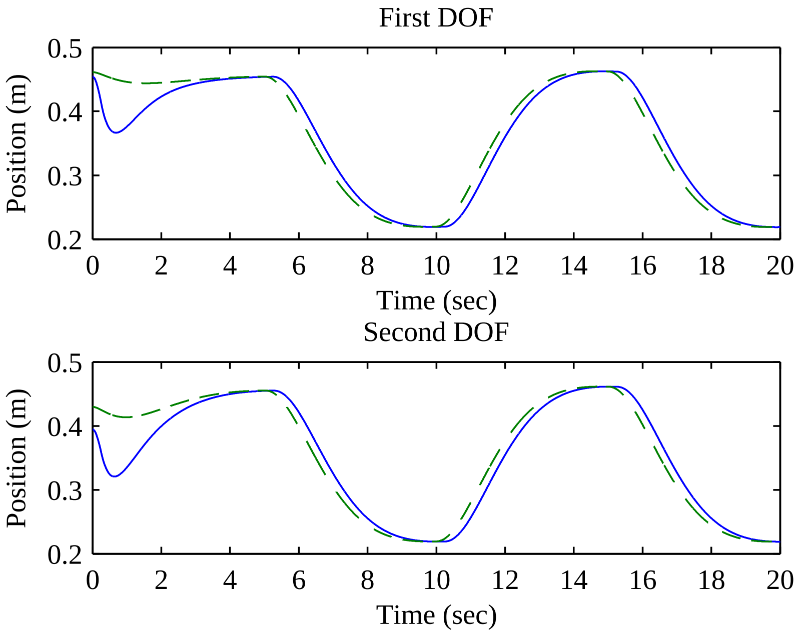


Figure 4.1: Position of the slave (solid) and master (dashed) end effectors with the slave in free motion.

master external force, along with its estimate, is shown in Figure 4.4. It is clear from this plot that the external force is estimated quite well. There are some chattering type effects from the switching, but this tends to be no worse than the noise that would be produced from a strain gauge for force sensing.

From these results, it is apparent that the teleoperation algorithm works well for the 2-DOF nonlinear manipulators when the slave is not in contact with an environment. The next case will show the performance of the algorithm when the slave is held in contact with an environment.

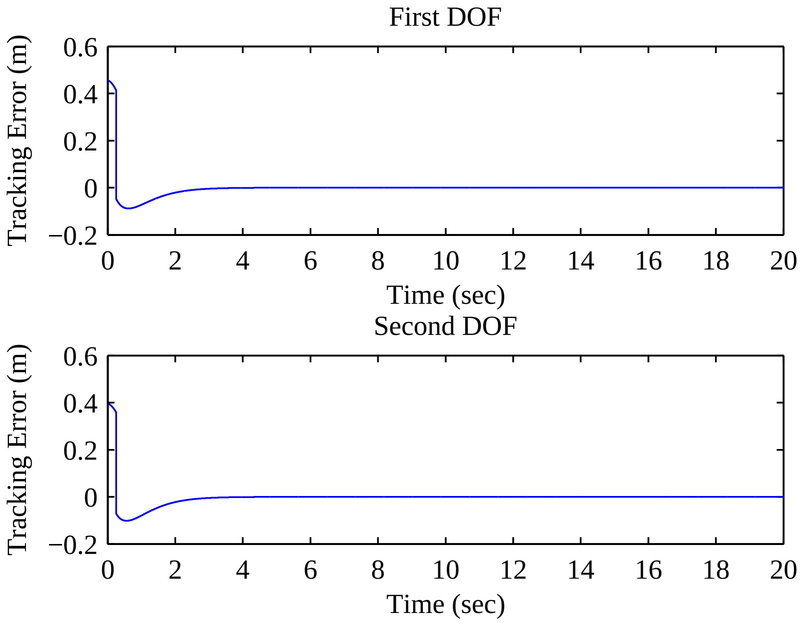


Figure 4.2: Slave tracking error with slave in free motion.

4.1.2 Delay-Independent Simulation Results with Slave in Contact

In this case, the slave side environment is modeled as a set of two linear springs, one connected to each slave end effector degree of freedom. That is,

$$F_e = K_e \left(X_{s_1} - X_c \right)$$

where $K_e = 150I$ and X_c is the location of the environment. In this simulation, $X_c = [0.43, 0.41]^T$. A time delay of 0.25 seconds in each direction is used, giving a round trip time delay of 0.5 seconds. Due to the definition of the slave desired impedance model, one does not see asymptotic tracking when the slave is in contact with an environment. If the slave were to come in contact with a stiff environment such as a wall, without some local compliance in the controller there could be large reaction forces which could lead to instability [89]. As a result, having some compliance in the slave control law allows for

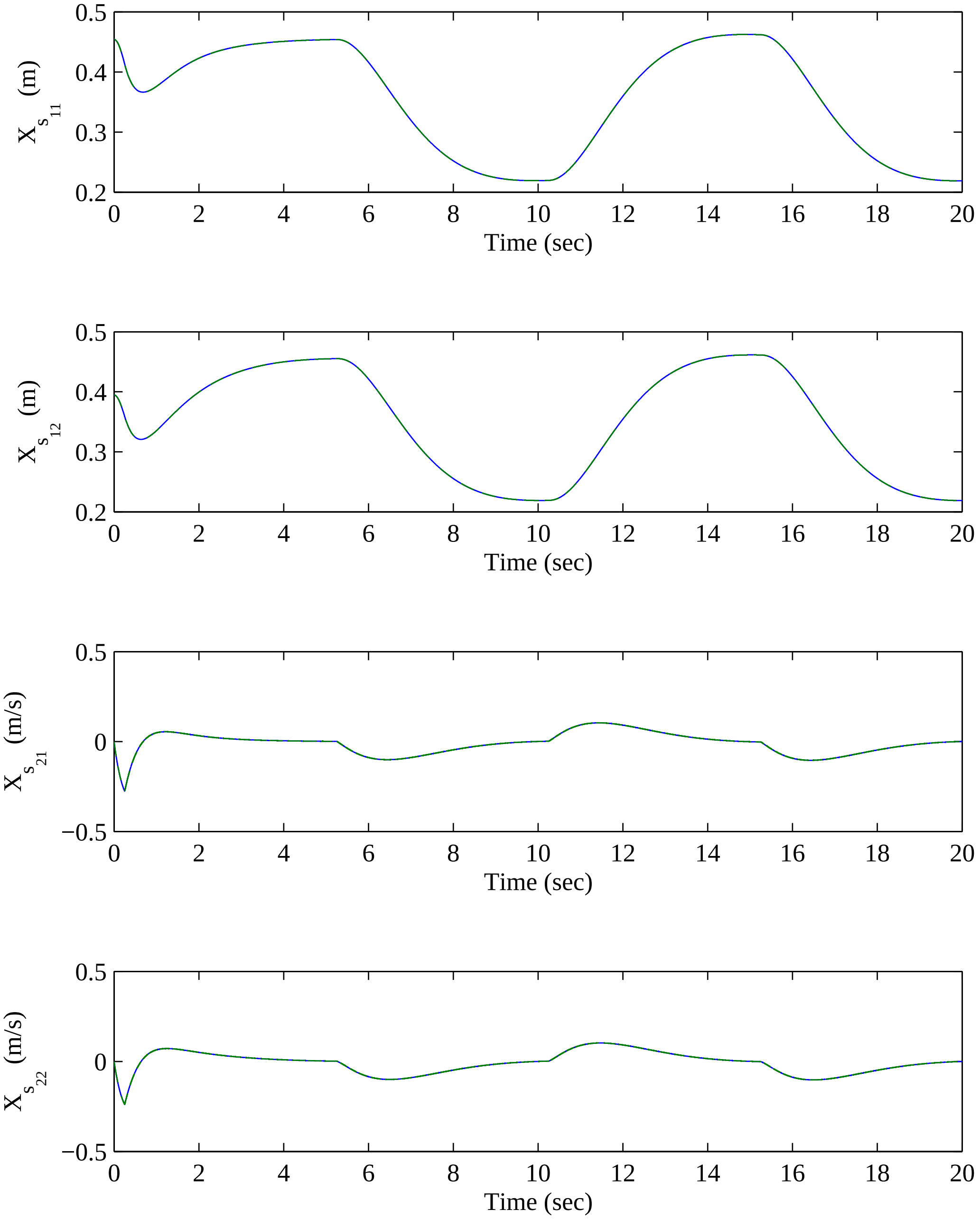


Figure 4.3: Slave actual (solid) and estimated (dashed) states when the slave is in free motion.

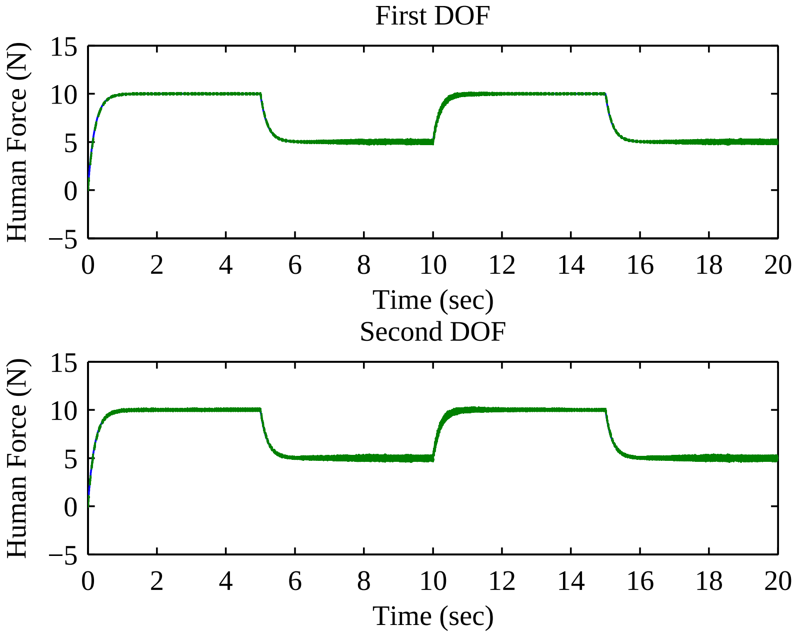


Figure 4.4: Actual (solid) and estimated (dashed) human force applied to the master manipulator.

more desirable behaviour in contact.

In this simulation, the desired impedance characteristics for each manipulator are chosen as, $\tilde{M}_m = 22I$, $\tilde{B}_m = 32I$, $\tilde{K}_m = 22I$, $\tilde{M}_s = I$, $\tilde{B}_s = 4I$, and $\tilde{K}_s = 4I$. These values are chosen to ensure that the closed loop system is stable independent of delay. See Section 3.4 for the selection of these parameters to ensure stability independent of delay. Figure 4.5 shows the master and slave end effector positions over time. Clearly, the slave is not tracking the master trajectory. Since the slave tracking error dynamics are driven by the environmental force, one expects the separation of master and slave trajectories that is seen here. Note also that the slave, when coupled with the environment, has a much smaller gain than the slave system in free motion. As a result, the same inputs to the slave system that are applied in the case of free motion have a much smaller impact on the slave+environment system.

The master manipulator states and estimates are shown in Figure 4.6. It is apparent again in this simulation that the observers are functioning correctly. Figure 4.7 shows the external environmental force acting on the slave and its estimate. The force estimates are quite accurate, though the effect of chattering is more pronounced in the force acting on the second degree of freedom. The effect of the relatively stiff spring environment is seen as well. It is the cause of significant oscillation for the first two to three seconds of the simulation. Coupling the slave to this spring environment shifts the equilibrium point of the slave away from the origin to the location of the spring.

From the results of this simulation, it is clear that the algorithm works as expected for ideal nonlinear plants in contact with an environment under some time delay. The next section will present further simulation results for less conservative impedance parameters and smaller delays.

4.1.3 Delay-Dependent Simulation Results with Slave in Contact

In order to demonstrate the usefulness of the results of Section 3.5 and the application of Theorem 3.5.1 to the teleoperation system, a second simulation with the slave in contact

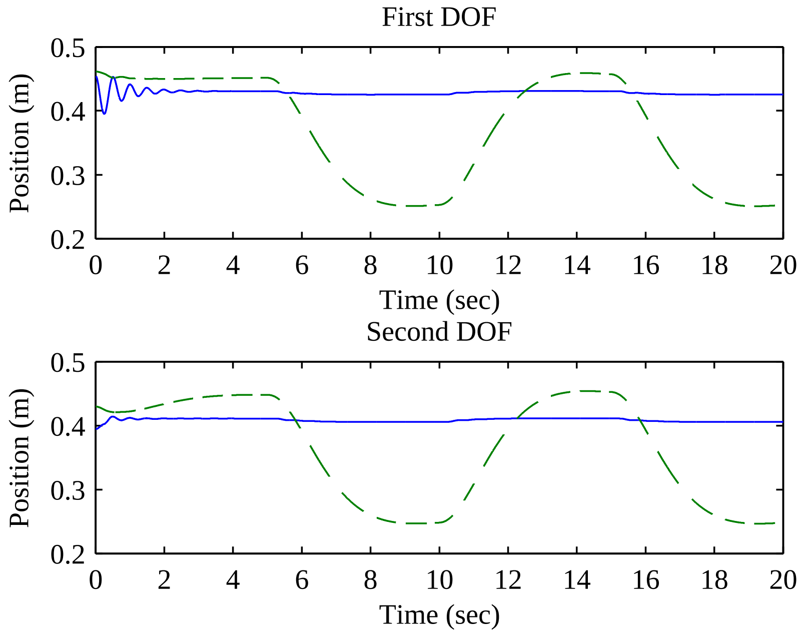


Figure 4.5: Position of the slave (solid) and master (dashed) end effectors with the slave in contact.

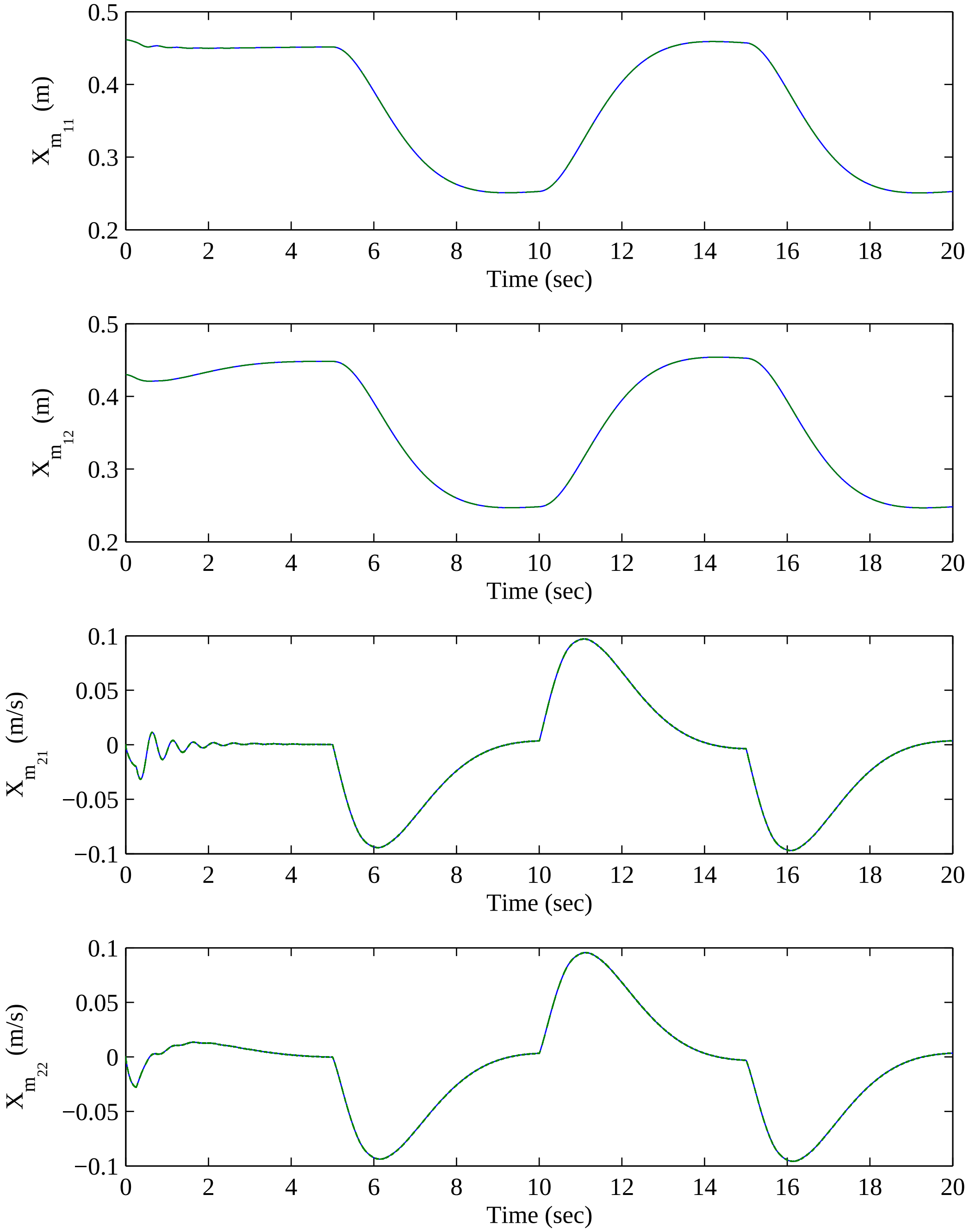


Figure 4.6: Master actual (solid) and estimated (dashed) states when the slave is in contact.

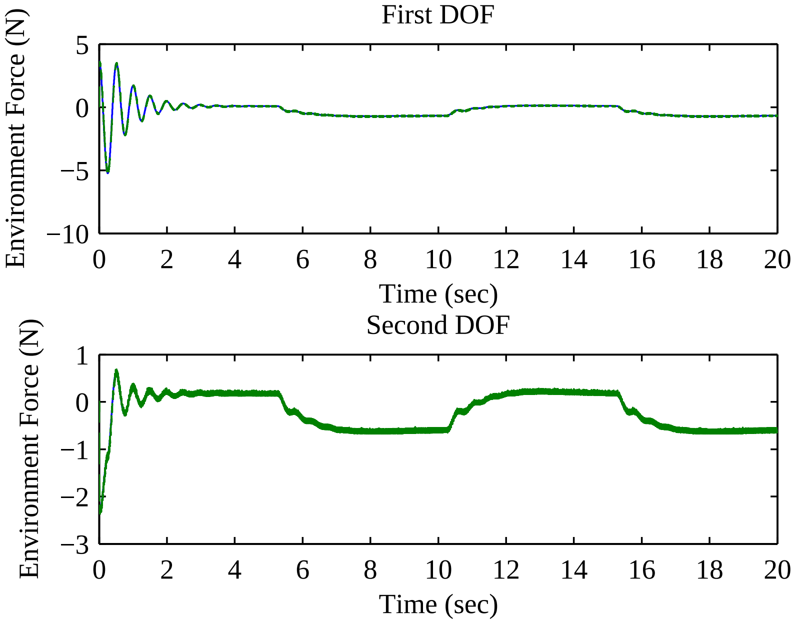


Figure 4.7: Actual (solid) and estimated (dashed) environment force applied to the slave manipulator.

with the environment is performed. In this case, desired master and slave impedance parameters are chosen as $\tilde{M}_m = \tilde{M}_s = I$, $\tilde{B}_m = \tilde{B}_s = 4I$, and $\tilde{K}_m = \tilde{K}_s = 4I$. The environment is modeled as a set of spring-dampers, one connected to each end effector degree of freedom. That is,

$$F_e = K_e \left(X_{s_1} - X_c \right) + B_e X_{s_2}$$

where $K_e = 150I$, $B_e = 10I$, and X_c is the location of the environment. In this simulation, $X_c = [0.43, 0.41]^T$.

Putting the system in the RFDE form (3.93) and computing the delay margin using Theorem 3.5.1, one arrives at a delay margin of $T_d = 0.070205$ seconds. That is, for delays up to and including T_d the system will be stable, but for values greater than T_d the system will be unstable. The following simulation was run for a time delay of 0.06 seconds in each direction. Figure 4.8 shows the master and slave end effector positions for this simulation. While the closed loop system is stable, the effect of the time delay on the system is apparent. Oscillations in the end effector positions show that the closed loop system is approaching instability. Figure 4.9 shows the same simulation, this time with the time delay increased to 0.065 seconds in each direction. In this case, the oscillations on the position states do not decay. In fact, it appears that the position of the slave first degree of freedom is experiencing oscillations that increase in amplitude. While the delay margin that was computed is greater than 0.065 seconds, this computation is numerical and thus contains error. It is possible that the actual delay margin may be closer to 0.065 seconds. Another cause of this discrepancy could be the nonlinearities in the actual teleoperator dynamics. The delay margin computation assumes a perfectly linear model, but the system consists actually of nonlinear models as well as controllers and observers to linearize the system. Any nonlinearities present could lead to some error on the computed delay margin. Regardless, the delay margin that is computed gives a good indication of the size of time delays that may be tolerated before the system becomes unstable. Compare the end effector positions with those of Figure 4.10. This third plot shows a simulation with the same parameters, except for a time delay of 0.001 seconds. It is clear that the

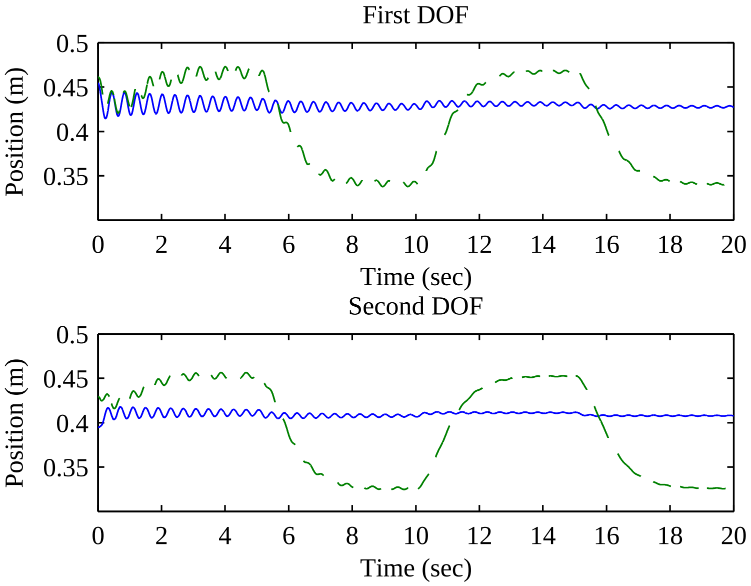


Figure 4.8: Position of the slave (solid) and master (dashed) end effectors with the slave in contact for the second in-contact simulation with time delay of 0.06 seconds.

increase in time delay takes the system closer to instability.

The slave tracking error is shown in Figure 4.11. As expected, since the slave is in contact with the environment, the tracking error does not approach zero asymptotically. Instead, it is driven by the environmental force. The presence of oscillations in the tracking error are due to the fact that the time delay is close to the threshold for stability.

4.2 Experiments

So far, a theoretical proof of stability for the novel bilateral teleoperation algorithm has been presented and verified through numerical simulation. It is also of great importance to perform the additional step of experimental verification on a real platform. In this case, many of the idealities present in a theoretical development and a numerical simulation are absent. In particular, issues such as unmodeled dynamics, friction, sensor noise, and a limited sample period appear in an experiment. The presence of these additional factors

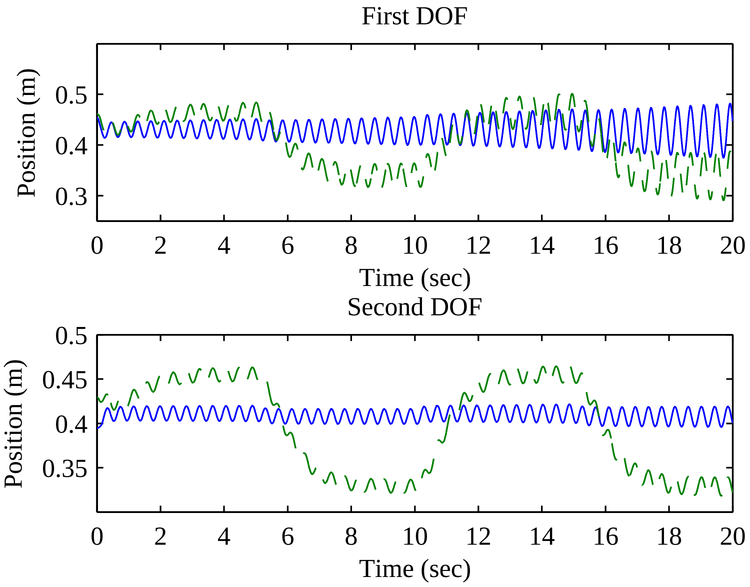


Figure 4.9: Position of the slave (solid) and master (dashed) end effectors with the slave in contact for the second in-contact simulation with time delay of 0.065 seconds.

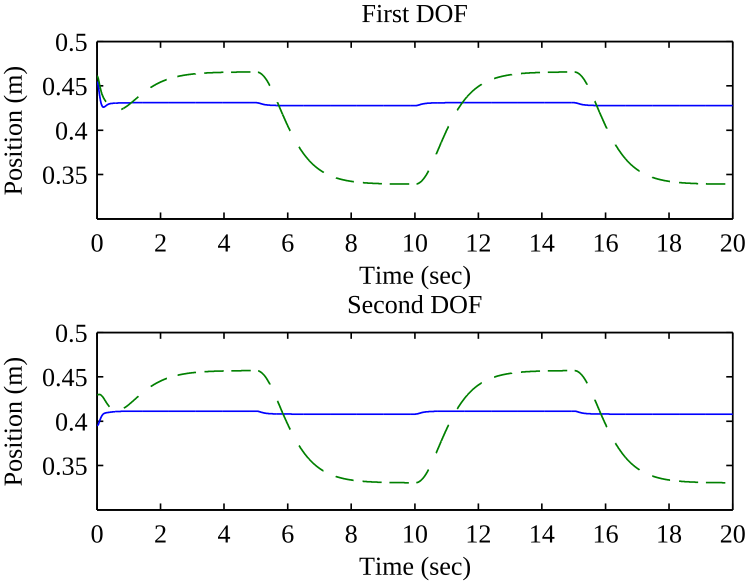


Figure 4.10: Position of the slave (solid) and master (dashed) end effectors with the slave in contact for the second in-contact simulation with time delay of 0.001 seconds.

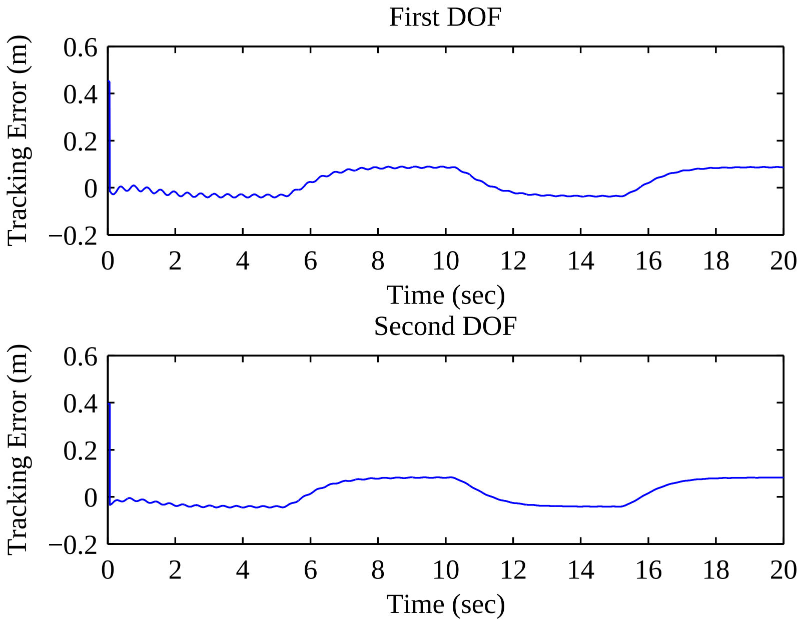


Figure 4.11: Slave tracking error for the second in-contact simulation.

makes it important to ensure that the proposed algorithm will in fact work in practice. This section will cover in detail the experimental implementation of this algorithm.

4.2.1 Robot Dynamics

Dynamic models of the base degree of freedom for each manipulator were developed prior to running the experiments. For each robot, the base degree of freedom was modeled as a mass-damper with Coulomb friction. The master plant dynamics are given as,

$$\dot{x}_{m_1} = x_{m_2} \tag{4.5}$$

$$\dot{x}_{m_2} = -\frac{B_m}{J_m}x_{m_2} - \frac{1}{J_m}f_{cm}(x_{m_2}) + \frac{1}{J_m}F_h + \frac{1}{J_m}F_m \tag{4.6}$$

where J_m represents the robot inertia, B_m represents the robot damping, and the term f_{cm} is defined as,

$$f_{cm}(x_{m_2}) = \begin{cases} f_{cm1} & \text{if } x_{m_2} \geq 0 \\ f_{cm2} & \text{if } x_{m_2} < 0 \end{cases}$$

Similarly, the slave dynamic model is given as,

$$\dot{x}_{s_1} = x_{s_2} \tag{4.7}$$

$$\dot{x}_{s_2} = -\frac{B_s}{J_s} x_{s_2} - \frac{1}{J_s} f_{cs}(x_{s_2}) - \frac{1}{J_s} F_e + \frac{1}{J_s} F_s \tag{4.8}$$

where,

$$f_{cs}(x_{s_2}) = \begin{cases} f_{cs1} & \text{if } x_{s_2} \geq 0 \\ f_{cs2} & \text{if } x_{s_2} < 0 \end{cases}$$

A system identification of these robots was performed using the approach proposed by Madill [94]. The plants are modeled as linear systems with the addition of a Coulomb friction term. The approach for system identification used by Madill is based on the observation when a robot modeled by the above dynamics is placed under proportional control, the velocity of the system remains positive until the first peak [94]. Using this observation, the robot under proportional control may be modeled as an equivalent linear system up until the time of the first peak in the position signal. The friction term, which becomes a part of the equivalent linear dynamics, is determined by plotting the magnitude of the first peak in position versus the magnitude of the input steps for a variety of step inputs. Finding the intercept of this curve with the input step magnitude axis, one can determine the coefficient of Coulomb friction. A similar approach, making use of the equivalent linear system, is used to identify the inertia and damping parameters. See [94] for complete details.

After performing the system identification of the base joint on the master manipulator, the following values for the parameters were obtained: $J_m = 0.8084$ kg·m^2, $B_m = 0.1150$ N·m·s·/rad, $f_{cm1} = 0.1090$ N·m, and $f_{cm2} = -0.0746$ N·m. Identifying the slave parameters yields the following values: $J_s = 0.8042$ kg·m^2, $B_s = 0.1768$ N·m·s·/rad, $f_{cs1} = 0.1462$ N·m, and $f_{cs2} = -0.0237$ N·m. Note that different values for Coulomb friction were obtained for different directions that the robots travelled. This shows that the friction is not symmetric. Additional sources of friction such as cable drag could contribute to this.

A detailed description of the hardware and software that make up the experimental platform is given in Appendix B.

4.2.2 Experimental Results

Due to limitations of the hardware used, the sample time in the experiments is limited to $T_s = 5 \times 10^{-4}$ seconds, giving a sample frequency of 2 kHz. While this sample frequency is relatively high for the robot manipulators, it is much lower than the sample frequency used in simulation. This lower sample frequency contributes to more chattering in the sliding mode observer estimates. As well, the position encoders produce signals with some noise. This can also be a complication with sliding mode observers in practice. As a result, the pure switching components in the observers were replaced by saturation functions, which allow the use of a boundary layer. This boundary layer reduces the effect of chattering and provides more usable state estimates. In all experiments, it was determined experimentally that boundary layer widths of $\epsilon_m = 10^{-4}$ and $\epsilon_s = 10^{-2}$, for the master and slave observers respectively, yielded the best state estimates in terms of reducing chattering. The observer gain parameters were chosen experimentally to ensure that the second order sliding mode gain conditions are met and that good state estimates are produced. In all experiments, the observer gains are set to $\lambda_m = 10.5$, $\alpha_m = 15.4$ for the master observer, and $\lambda_s = 10.5$, $\alpha_s = 15.4$ for the slave as well. In order to obtain the estimated force signals from the observers, the switching signals were passed through 5 Hz second order low pass filters. This filtering yielded very usable force estimates.

It was also found in practice that the use of a pure switching component in the slave control signal produced significant chattering in these manipulators. As a result, a boundary layer was used in the slave sliding mode controller. Various values for the width of the boundary layer were tried for different experimental runs. As well, the slave control signal was passed through a second order low pass filter before being applied to the plant. Without the use of the filter, the chattering became too great, causing too much power draw through the motor amplifier power supply. Various bandwidths for the second order filter were used in different experimental runs.

From a usability point of view, it was found that the closed loop dynamics for the master manipulator should not contain a spring term. Viewing the master manipulator as a tool being used by an operator, a spring term makes the tool more difficult to use as it moves away from the origin. As a result, for many experimental runs no spring term was used at the master side. On the slave side, however, a spring term is necessary in order to ensure asymptotic tracking of the delayed master trajectory. Without a spring term, the tracking error dynamics are not asymptotically stable, and the slave no longer tracks the delayed master trajectory either in or out of contact.

In the experiments where the slave manipulator comes into contact with an environment, a stiff metal structure was used. However, the last link on the manipulator (the link that contacts the environment) is a link with some flexibility, giving some compliance to the manipulator-environment interface. This allows for a softer contact with lower reaction forces than if the manipulator were very rigid. When the manipulator is in contact with the environment, the flexible link could be considered a part of the slave environment. The environment dynamics are not specifically modeled, but closed loop impedance parameters were chosen experimentally to ensure stable behaviour in contact.

The implementation of this teleoperation algorithm involves choosing the closed loop impedance parameters. A number of different closed loop stability cases were examined in Chapter 3. In each of these cases, methods for choosing the closed loop parameters for a variety of environments were discussed. In practice, however, one may not know precisely the environment dynamics or the time delays. The delay-independent stability results effectively rely on making the master and slave systems sufficiently sluggish so that they remain stable in the presence of time delays. This knowledge may be used as the basis for selecting closed loop impedance parameters in practice. One may begin by choosing the masses of the closed loop master and slave manipulators to be fairly large, leading to a sluggish system. With a stable closed loop system in practice, one can adjust the impedance parameters in order to yield more desirable performance. Another possible approach for choosing the impedance parameters in practice is through simulation. With good dynamic models, one may simulate the teleoperation system and choose parameters in simulation that will work well in the experimental implementation.

Experiment with No Time Delay

In the first experiment, the closed loop impedance parameters for the master manipulator are chosen as $\tilde{M}_m = 7$, $\tilde{B}_m = 14$, and $\tilde{K}_m = 0$. The slave closed loop impedance parameters were chosen as $\tilde{M}_s = 7$, $\tilde{B}_s = 14$, and $\tilde{K}_s = 7$. The slave sliding mode controller gain was chosen to be $K_g = 3$. The boundary layer on the slave sliding mode controller was set to $\epsilon_c = 0.15$. The slave control signal was passed through a second order low pass filter with a bandwidth of 100 Hz before being applied to the slave manipulator. A stiff metal environment was placed on the slave side at an angular position of roughly -0.27 rad.

Figure 4.12 shows the master and slave positions during the run. When the slave comes into contact with the environment, it is no longer able to travel in that direction. However, as expected from the desired slave impedance model, the master may continue to travel in that direction. When the slave comes out of contact with the environment it is able to track the master position reasonably well. There is some error in the master position tracking. Several factors may contribute to this. The first is modeling error. It is apparent that there is some static friction in the system that has not been modeled, causing a non-zero position error in steady state since not enough control effort is being applied to overcome the static friction. As well, there is certainly some error on the inertial and damping parameter values in the identified robot models. An additional source of error is due to the fairly large boundary layer that is used in the sliding mode controller. This contributes to tracking error, as the sliding surface $s = 0$ is never actually reached. This tracking error is on the order of 0.06 rad, or roughly 3.4 degrees. A final source of error is error on the velocity estimates in the sliding mode observers. Due to the use of saturation functions in the observers, instead of pure switching terms, some error arises in the state estimates. This effect is shown further in Section 4.2.3, where the same dynamic models

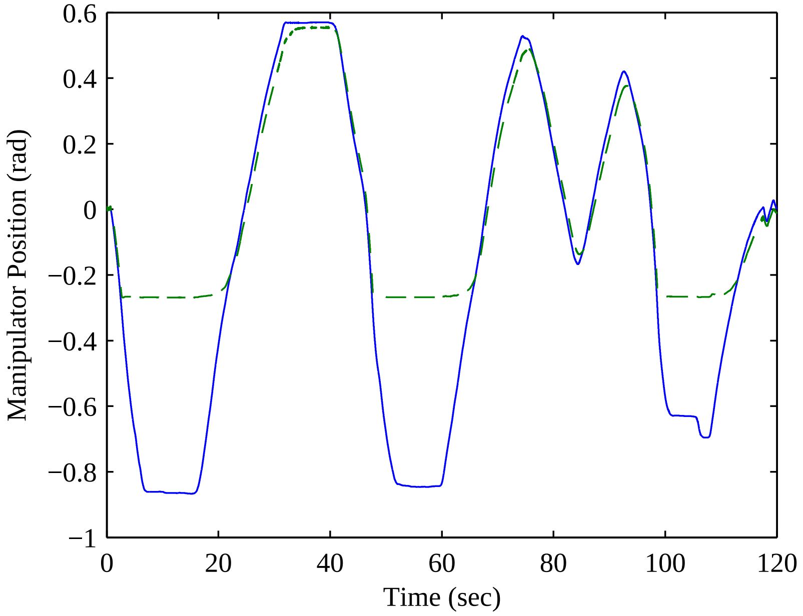


Figure 4.12: Position of the master (solid) and slave (dashed) manipulators in the first experiment.

used in the experiments are simulated.

The estimate of the external torque acting on the slave is given in Figure 4.13. It is evident from this plot when the slave comes into contact with the environment. There is an initial peak in the torque when the slave first comes into contact, but it then reaches a steady state value. This torque estimate is transmitted to the master manipulator, where it can be felt by the operator. When the slave is not in contact with the environment, there is some non-zero torque estimate. This is as a result of running the observers at a relatively large time step. If it were possible with the hardware being used to run the observers at a smaller time step, more favourable results would be seen, as was the case in simulation. The slave state estimate are shown in Figure 4.14. While there is some noise due to chattering on the velocity estimate, it remains quite usable.

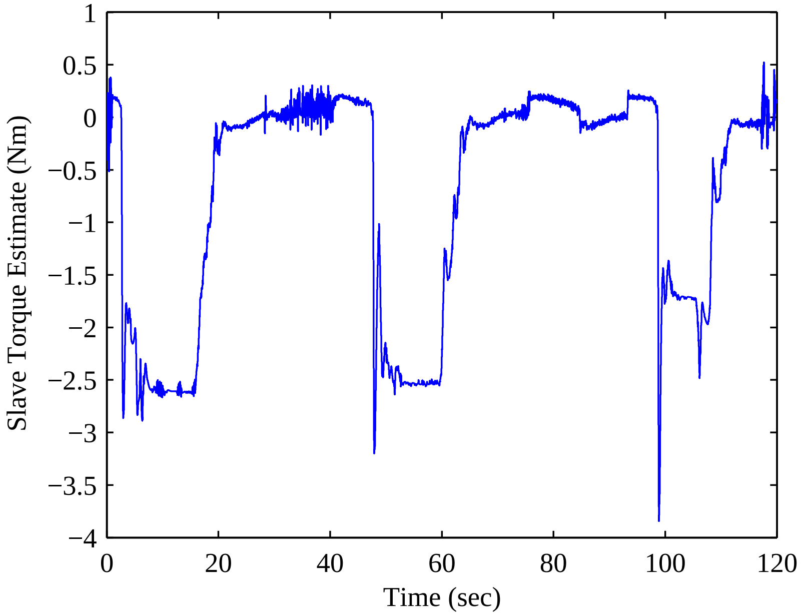


Figure 4.13: Estimate of the environmental torque acting on the slave in the first experiment.

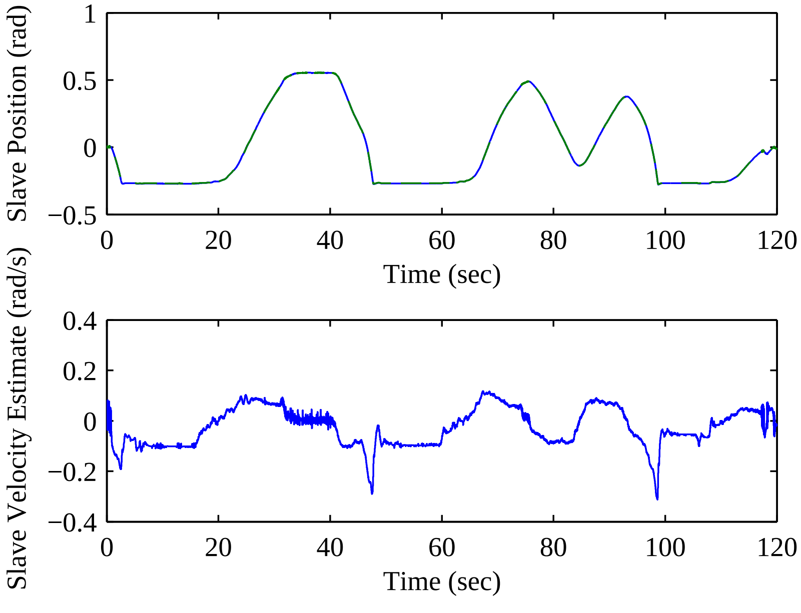


Figure 4.14: Slave state estimates and position measurement (top plot, solid line) in the first experiment.

Experiment with Small Time Delay

A second experiment was performed, this time introducing some time delay. The same closed loop impedance parameters were chosen for both the master and slave in this case. The slave control signal was filtered with a second order low pass filter with a bandwidth of 25 Hz. Time delays of 30 milliseconds were introduced from the master to the slave side, and from the slave back to the master.

Figure 4.15 shows the master and the slave trajectories for this experiment. In the presence of small time delays, the system remains stable both in and out of contact with the environment. Again there is some small error in tracking when the slave is in free motion. This is likely for the same reasons as the previous experiment.

The estimate of the torque applied to the master manipulator is given in Figure 4.16. A fairly clean estimate is supplied by the observer. During the periods of time that the slave is in contact with the manipulator, the human operator applies a constant non-zero torque to the master side, however when the slave is out of contact the torque applied by the human is roughly zero when a constant position has been reached. This is due to the fact that the master closed loop dynamics have a pole at the origin. When the slave is out of contact with the environment, and therefore not reflecting any torque back to the master side, a constant non-zero torque on the master would cause it to continue to move. However, when the slave is reflecting a torque back to the master, a constant non-zero torque on the master is required to balance this torque.

This second experimental run has shown that the teleoperation algorithm works effectively even in the presence of forward and backward time delays in the communications.

Experiment with Larger Time Delay

A third experiment was performed, this time introducing a larger time delay. For this experiment, the master closed loop impedance parameters were chosen as $\tilde{M}_m = 22$, $\tilde{B}_m = 32$, and $\tilde{K}_m = 22$. The slave closed loop impedance parameters were chosen as $\tilde{M}_s = 7$, $\tilde{B}_s = 42$, $\tilde{K}_s = 63$. The slave values were chosen to give the slave manipulator, and

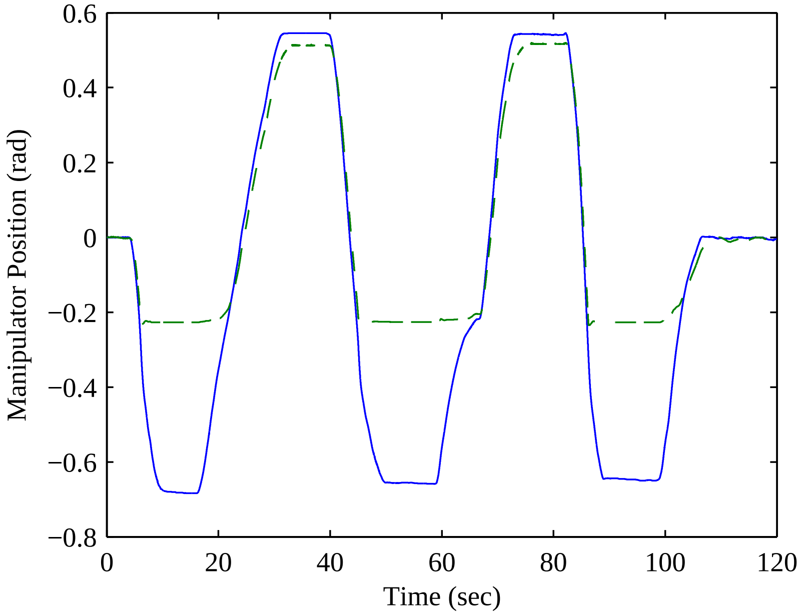


Figure 4.15: Position of the master (solid) and slave (dashed) manipulators in the second experiment.

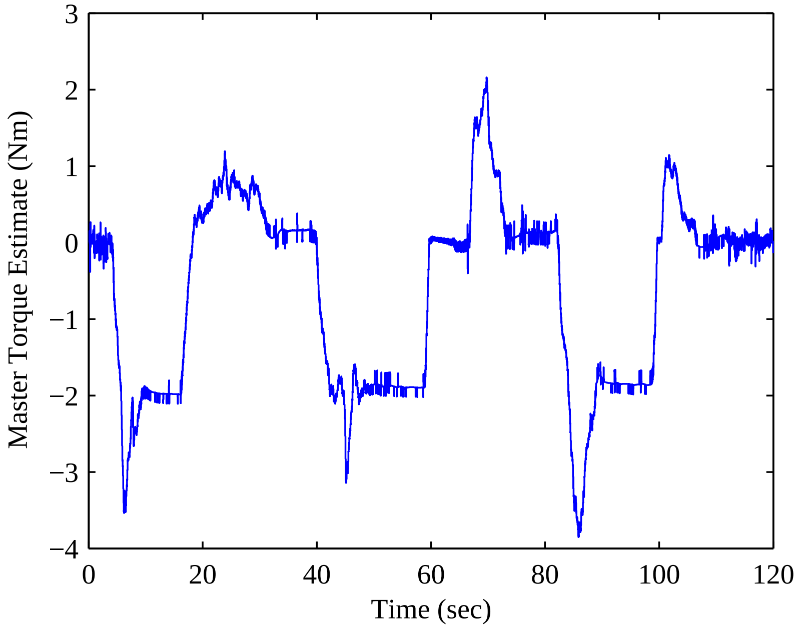


Figure 4.16: Estimate of the human torque applied to the master in the second experiment.

by extension the tracking error dynamics, a pair of critically damped poles at $s = -3$. The master impedance parameters were chosen such that the closed loop system would be stable independent of delay by the Small Gain Theorem, assuming that the environment is modeled as a linear spring with spring constant 19 N/m. The slave control signal was filtered with a second order low pass filter with a bandwidth of 25 Hz. Time delays of 0.5 seconds were introduced from the master to the slave side, and from the slave back to the master, for a round trip delay of 1 second. The same environment as the previous experiments was placed on the slave side.

Figure 4.17 shows the master and the slave trajectories for this experiment. In the presence of fairly large time delays, the system remains stable both in and out of contact with the environment. Not surprisingly, there is again some small tracking error when the slave is in free motion. This is likely caused by the same factors as in the previous experiments. That is, modeling error, unmodeled effects, and large sliding mode boundary layers could all be contributing factors to the tracking error. There are some small periods of time where the slave manipulator experiences some chattering. This is likely caused by the noise on the state and force estimates due to running the observers at a larger than ideal sample period. However, it is not significant in this experiment.

The estimate of the torque applied to the master manipulator is given in Figure 4.18. Due to the spring term on the master manipulator, which is actually a requirement for the closed loop stability analysis, the human operator must always apply a non-zero force to the manipulator when it is away from the origin. In practice, it is desirable to set the spring term to zero, or close to zero, on the master side as this gives the operator the sensation of using a tool that can be arbitrarily placed in the space and left there. It is also apparent in Figure 4.18 that during periods of time when the slave manipulator is in contact with the environment, the operator must apply more torque to the master manipulator to compensate for the environment torque that is fed back to the master from the slave. This further shows the effectiveness of the force feedback portion of the algorithm.

This third experimental run has shown that the teleoperation algorithm works effectively even in the presence of relatively large forward and backward time delays in the

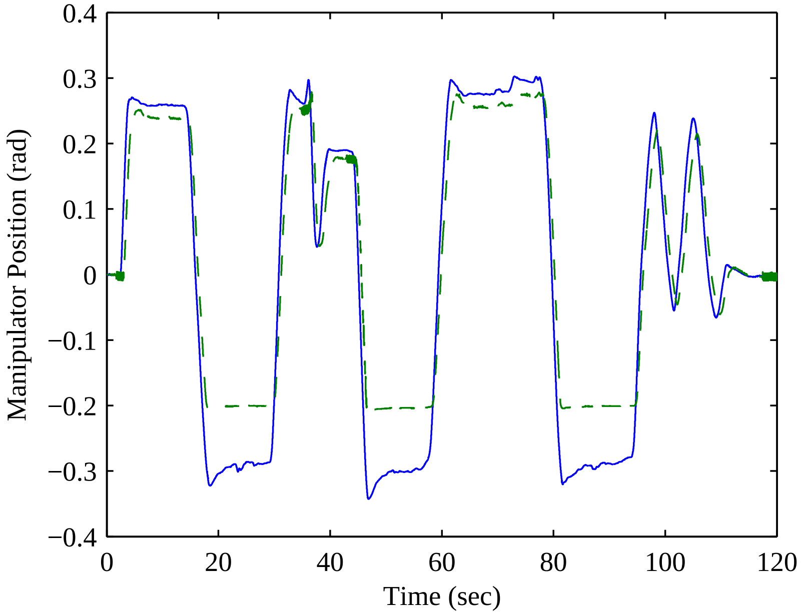


Figure 4.17: Position of the master (solid) and slave (dashed) manipulators in the third experiment.

communications, both when the slave is in free motion and in contact with an environment.

4.2.3 Simulation of Experimental Dynamics

In order to verify the experimental results, a simulation of the experimentally identified robot dynamics in the teleoperation algorithm was performed. This simulation makes it possible to compare the experimental results, which are affected by modeling error and sensor noise, to the results that would be expected for this setup, assuming there were no modeling error or sensor noise. The robot dynamics used here in simulation are exactly the identified dynamics of the base joint of each manipulator used in the experiments.

For the simulations in this section, the closed loop impedance parameters for the master and slave manipulators are chosen as $\tilde{M}_m = \tilde{M}_s = 7$, $\tilde{B}_m = \tilde{B}_s = 14$, and $\tilde{K}_m = \tilde{K}_s = 7$. The slave sliding mode controller gain was set to $K_g = 3$. The boundary layer on the slave

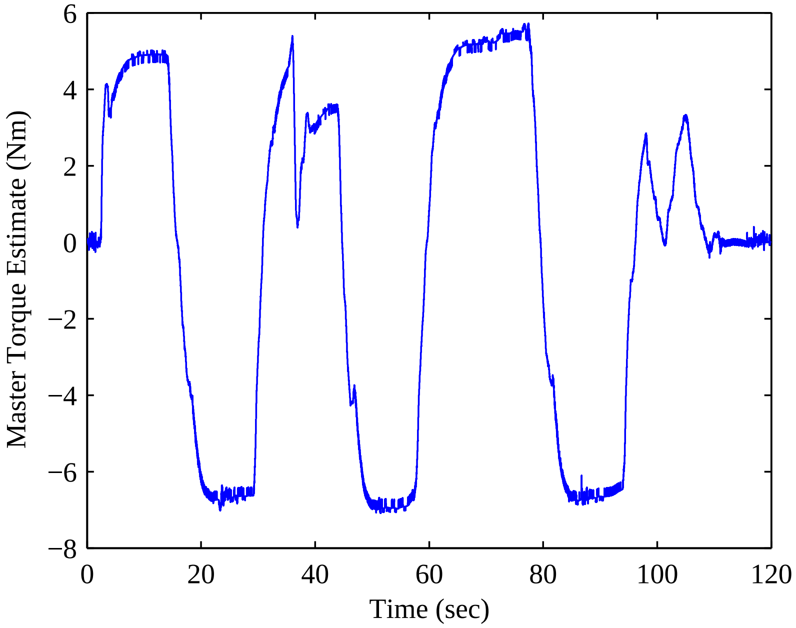


Figure 4.18: Estimate of the human torque applied to the master in the third experiment.

sliding mode controller was set to $\epsilon_c = 0.15$. The slave control signal was passed through a linear second order low pass filter with a bandwidth of 100 Hz before being applied to the manipulator. A spring-damper environment is used at the slave side at an angular position of -0.27 rad. The environment has spring and damping constants of $K_e = 40$ and $B_e = 5$. A time delay of 30 milliseconds in each direction is used in the simulation. All of the other parameters in the system were set the same as in the experiments. The choice of parameters in this simulation is the same as in the experiment with a small time delay, with the exception of the non-zero spring term in the master dynamics here. This non-zero spring was used so that, in simulation, the application of a step signal to the master dynamics would result in a constant output at steady state. The human input to the master manipulator is a step function passed through a first order low pass filter with a pole at $s = -5$.

The master and slave position signals for the simulation are shown in Figure 4.19. It is apparent from this plot that there is some steady state error in the tracking. Comparing this to Figure 4.15, there is steady state tracking error in both plots. Figure 4.20 shows

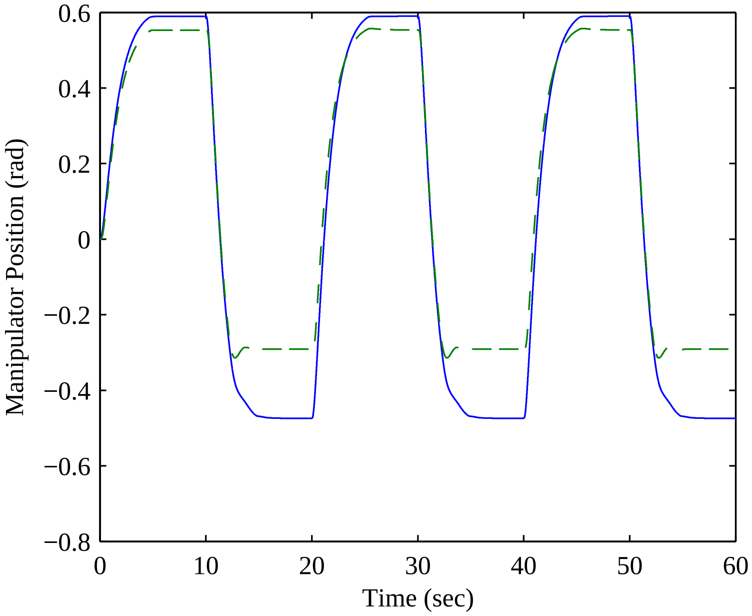


Figure 4.19: Position of the master (solid) and slave (dashed) manipulators in the simulation with experimental dynamics.

the master velocity signal and its estimate for the simulation with experimental dynamics. The master velocity state estimate has an error at steady state as well. This error is likely the cause of the tracking error in Figure 4.19. While only the master velocity signals are shown here, the same behaviour was observed in the slave velocity signals. In both the experiments and this simulation, the discontinuous switching terms in the observers were replaced with boundary layers. This was done in the experiments to reduce the high frequency content in the state estimates and to make the observers less sensitive to measurement error.

A second simulation with experimental dynamics was performed, this time reducing the width of the boundary layers in the observers. Boundary layer widths of $\epsilon_m = \epsilon_s = 10^{-6}$ were used for the master and slave observers. Figure 4.21 shows the master and slave positions for this simulation run. It is clear that a reduction in the boundary layers of the observers directly results in better steady-state tracking performance. Examining the master velocity estimate in Figure 4.22, it is clear that the reduction in the width of the

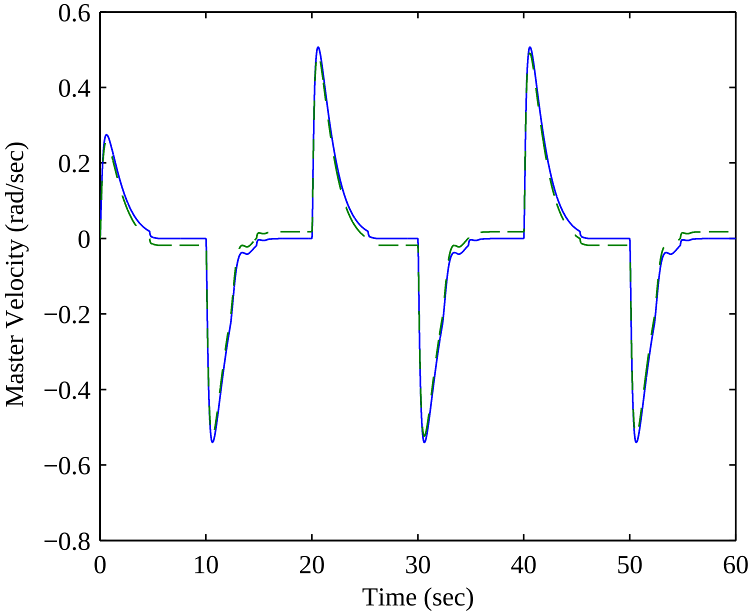


Figure 4.20: Master velocity (solid) and velocity estimate (dashed) in the simulation with experimental dynamics.

observer boundary layer has reduced the steady-state velocity error. Due to the nature of the experimental setup, the observer boundary layers were made as small as possible while still yielding useful state estimates. However, it is clear that in this case the observer boundary layer widths have an impact on tracking performance.

An examination, through simulation, of the experimental dynamics using the same parameters as in the experiments themselves has provided validation of the experimental results. In particular, it was seen that the presence of steady-state tracking error in the experiments can be attributed to, at least in part, a steady-state error on the velocity estimates.

4.3 Summary

This chapter has examined the implementation of the novel bilateral teleoperation algorithm that has been proposed in this work. Initially, numerical simulation results were

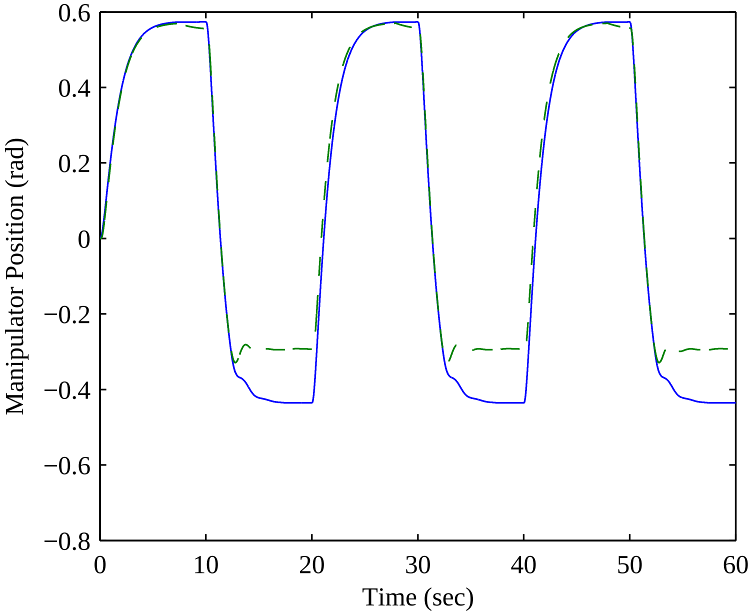


Figure 4.21: Position of the master (solid) and slave (dashed) manipulators in the simulation with experimental dynamics with smaller boundary layers.

presented that show the effectiveness of the algorithm in a situation that is close to ideal. That is, the simulated robots were perfectly modeled, very small sample periods were used, and noise was not a factor.

In all of the simulations, a 2-DOF robot model was used for both the master and slave manipulators. The first simulation examined the performance of the algorithm in free motion in the presence of time delays. The algorithm performed well and good tracking results of the delayed master trajectory by the slave were obtained. The second simulation examined results with the slave in contact with a linear spring environment. In this simulation, closed loop impedance parameters were chosen to ensure that the whole system is stable independent of delay. This decision resulted in master impedance parameters that were quite large, making the master side gain very small. This effectively resulted in a very "heavy" master manipulator. The final simulation showed the delay-dependent stability results for the case where the slave is in contact with a known linear environment. In this case, master and slave impedance parameters were chosen, and a delay margin was found

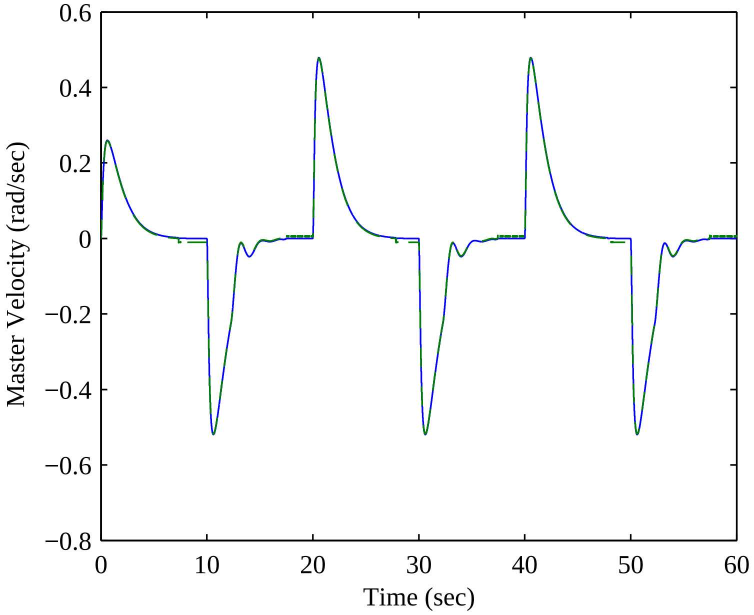


Figure 4.22: Master velocity (solid) and velocity estimate (dashed) in the simulation with experimental dynamics with smaller boundary layers.

for these parameters. It was found that simulating the system with a time delay close to this delay margin did result in a stable system, but oscillations in the states were present. As the time delay is increased toward the delay margin, the oscillations increase. Past that margin, the closed loop system is unstable.

Following the simulation results, experiments were performed on the University of Waterloo Teleoperation Platform. The experiments verified that the algorithm does in fact work in practice, in the presence of non-idealities such as noise, modeling error, finite sample times, and unmodeled dynamics. All of the experiments involved the transition from the slave in free motion to the slave in contact with an environment. The first experiment considered the case of no time delays, and showed effective tracking and force feedback. In the second experiment, a small time delay was inserted in both the forward and backward paths while keeping the same closed loop impedance parameters. The system remained stable both in and out of contact. A third experiment was performed, this time with a much larger time delay. New impedance parameters were computed to ensure that the system would be stable independent of delay. In all cases, the experiments worked as expected. Tracking of the delayed master trajectory by the slave was seen, and the estimated forces were properly reflected back to the operator through the master manipulator. Finally, simulations based on the experimental dynamics were run in order to compare the experimental results to a situation free from modeling error and sensor noise.

Chapter 5

Conclusions and Future Research

Having presented the entirety of the new theory and experimental results for a new force estimating bilateral teleoperation algorithm, this chapter will focus on summarizing the results and discussing potential directions for future research in this area.

5.1 Conclusions

The focus of this work has been the presentation of a novel bilateral teleoperation algorithm for n-DOF nonlinear robot manipulators connected through time delayed communications. Typically, in the field of bilateral teleoperation, a teleoperator either makes use of force sensors for force feedback (i.e. the position-force architecture) or makes use of some function of the position as a substitute for force feedback (i.e. position-position and position-error-based architectures). It is possible to show that a teleoperation system that makes use of force sensors can provide a higher level of transparency, which effectively means greater performance, than a position-position type system. However, the cost of this decision is that a more complicated hardware setup must be used. In particular, force sensors must be mounted at the slave site, and in many situations the master site as well. Not only is this more costly in terms of physical equipment, but force sensors may produce noisy measurements and may be difficult to calibrate [87].

The approach to bilateral teleoperation presented in this work has combined advantages of both popular types of teleoperator architectures. Specifically, a teleoperation algorithm with the performance of a position-force architecture has been developed, but this approach requires the more simple hardware setup found in position-position type architectures. This has been accomplished by integrating unknown input second order sliding mode observers into the teleoperator system. These unknown input observers, when designed with an accurate dynamic model of the plant being observed, provide an estimate of the full state of the plant as well as an estimate of the unknown input acting on it. When the external force applied to a robot manipulator is viewed as an unknown input, an estimate of this force is obtained from the unknown input observer.

When considering the prior work in bilateral teleoperation, this work is most similar to the algorithm presented by Cho and Park [67]. The new approach presented in this thesis has advanced the state of knowledge in bilateral teleoperation, over and above the contributions made in [67]. The algorithm presented by Cho and Park is designed for linear 1-DOF teleoperation systems, and requires full state and force measurements. As previously mentioned, the new algorithm is developed for nonlinear n-DOF non-kinematically redundant manipulators, and requires only position measurements. Table 5.1 describes the advances in the state of the art provided by this algorithm, as compared to the algorithm of Cho and Park [67]. Note that, were the algorithm presented here implemented with a linear 1-DOF teleoperation system, one would expect to see the same performance as the algorithm in [67], but this algorithm would require fewer sensors and as a result a more simple hardware configuration.

In particular, the major new contributions to knowledge presented in this thesis are as follows,

- The development of a novel bilateral teleoperation algorithm that yields the performance of a position-force architecture in terms of transparency, while only relying on position measurements. In particular, the stability result involved a novel combination of a sliding mode controller and sliding mode observer where upper bounds on convergence times were determined in order to guarantee the stability of the

Table 5.1: Summary of differences between the new algorithm proposed in this work and that presented by Cho and Park.

	Cho and Park [67]	*New Algorithm*
Degrees of Freedom	1	$n \leq 6$
Type of Plant	Linear	Nonlinear
Required Sensors	Position, Velocity, Force	Position
Operator/Environment Assumptions	Passive human and environment	General nonlinear operator for both human and environment
Stability Results	Absolute stability for passive human and environment	Small gain stability results for linear and nonlinear environments (no passivity requirement), computation of delay margin for linear environment with commensurate delays

plant/controller/observer combination. This result was first published for the case of first order sliding mode observers by the author and his supervisor in [25].

- Closed loop stability of the teleoperator algorithm was proved for a variety of environmental conditions. As previously discussed, many closed loop stability proofs for teleoperators rely on the assumption that both the human and the slave environment are passive systems. This can be a particularly restrictive assumption. The closed loop stability results presented in this work do not require passivity of either the human operator or the slave environment.

The bilateral teleoperation algorithm presented in this thesis assumes that the dynamic model of each manipulator is well known. A computed torque method controller was used at the master side for the master manipulator, while a sliding mode controller was used at the slave side to ensure that the slave tracks the delayed master trajectories. Unknown input second order sliding mode observers were placed at both the master and slave sides to perform state and force estimation. Control design was performed in the robot task space for each of the manipulators. This was done so that each degree of freedom in task space could be assigned a characteristic impedance, and so that analysis of the system in terms of the forces acting on the end effectors could be simplified. The process of expressing the robot dynamics in task space was outlined in detail.

The specific controllers and observers used in the algorithm were next presented. Theorem 3.3.1 presented the stability proofs for each of the master and slave plants. This theorem showed that, provided certain assumptions are satisfied, both the master and slave will have the desired impedance models, and estimates of all states and forces will be available.

Following the proof of stability for each of the master and slave manipulators, the stability of the entire closed loop was considered. Theorem 3.4.2 showed that, given the closed loop teleoperator where the slave is in contact with a linear spring-damper environment and the H_∞ norm of the slave+environment system is known, master side closed loop impedance parameters may always be chosen such that the closed loop teleoperator is stable independent of the delays in the communications. In the case of a nonlinear finite-gain stable environment with a known gain, Theorem 3.4.3 showed that the feedback connection between the slave and this environment can always be made finite-gain stable with a proper choice of slave closed loop impedance parameters. It is then straightforward to show that the entire closed loop may be stabilized independent of delay, following the results of Theorem 3.4.2. A final closed loop stability result was presented for the case where the slave side environment may be modeled as a set of linear spring-dampers and where the delays in each direction are commensurate. In that case, the closed loop dynamics are expressed in retarded functional differential equation form and Theorem 3.5.1 from [92] is used to determine the delay margin for the system. That is, for a particular choice of closed loop impedance parameters that ensure stability with no delay, an upper bound on the delay before which the system goes unstable may be found.

A transparency analysis of the closed loop teleoperator was given in order to determine the performance of this approach. This analysis shows that, while the approach is not ideally transparent, some useful properties in terms of performance are obtained. In particular, perfect force tracking and perfect free motion tracking of the delayed master trajectory by the slave are achieved. The hybrid matrix also shows that there is some input impedance at the master side. Physically, this means that the operator will feel the dynamics of the master manipulator. One can consider the master dynamics as part of a virtual tool that allows the operator to perform the teleoperation task. The second nonideality in the transparency analysis is the occurrence of an admittance between the slave environment force and the slave motion when the master manipulator is not in motion. This result is expected, as this admittance was incorporated in the slave control design. Some compliance between the slave and the environment is important in practice in order to prevent large reaction forces.

In practice, there is always some error associated with a dynamic model of a real plant. A robustness analysis of the teleoperator system was performed in order to examine the impact of modeling error on this system. This analysis showed where the unmodeled terms would appear in the closed loop expressions of the dynamics. However, without specific knowledge of the unmodeled dynamics, it was not possible to conclude anything theoretically in terms of stability in the presence of significant modeling error.

Both numerical simulation and experimental results have been presented in this work. The numerical simulations used 2-DOF nonlinear robot manipulators as plants at the master and slave sides. Three sets of simulations were performed. In all simulations, there was some time delay in both the forward and backward directions. The first simulation showed the performance of the algorithm when the slave is in free motion. The slave clearly tracked the delayed master trajectory, as desired. The second simulation showed the performance of the algorithm when the slave was in contact with an environment. Closed loop impedance parameters were chosen to stabilize the teleoperator loop independent of delay. The results of the simulation showed that the system remained stable in environmental contact and that force estimates were reflected back from the slave to the master, as expected. A final simulation with the slave in environmental contact was performed. In this case, a particular delay margin was computed for a certain choice of closed loop impedance parameters. The system was simulated with the time delay set to just below this margin. From these results, it was clear that the system remained stable, but was approaching instability. This simulation showed the delay-dependent stability results of

Chapter 3 in practice.

Finally, the results of a set of experiments carried out on the University of Waterloo Teleoperation Platform were presented. Three experiments were performed: an experiment with no time delay, one with a small time delay, and one with a larger time delay. All experiments involved transitions of the slave from free motion to environmental contact. The experiments served to validate the theory, and further showed experimentally that the proposed algorithm has some robustness properties. Despite the presence of modeling error, unmodeled terms, sensor noise, and finite switching of the discontinuous sliding terms, the algorithm performed as expected in practice.

5.2 Future Work

There are several avenues for pursuing future research related to the work presented in this thesis. Initially, modifying the algorithm so that the controller at the slave side is a second order sliding mode controller and developing an accompanying proof of stability would be a useful endeavour. This would minimize the effect of chattering and the discontinuous switching in the slave sliding mode control signal. Such a change would further facilitate experimental implementation and hardware requirements.

The stability analyses performed in this thesis have assumed that the slave manipulator is in contact with the slave side environment. Stability has been shown in this case, and a proof of closed loop stability for the case when the slave is in free motion is trivial. However, the issue of contact transition stability has not yet been examined from a theoretical standpoint. Experimentally, it is apparent that the system remains stable through contact transitions. It would be beneficial, though, to perform a theoretical analysis of this issue. Though it is typical in the literature to examine stability for the situation where the slave remains in contact, contract transition stability results do exist for other algorithms [95]. If a contact transition proof of stability were developed, the conditions on the stability result could also be verified experimentally.

Another extension on the experimental side would be the implementation of this algorithm using robots with more than one degree of freedom. Due to the hardware available, implementation in practice was limited to one degree of freedom in this work. While the simulation results confirmed the effectiveness of the algorithm for more than one degree of freedom, an experimental implementation would further strengthen the result.

In the experimental validation of this algorithm, the sample period was limited to $T_s = 5 \times 10^{-4}$ seconds. While this sample period is sufficiently small for the time constants of the robots, the sliding mode observers show significant improvements in performance with smaller sample periods. If a hardware setup were available to allow an implementation of the algorithm in real-time with a smaller sample period for the sliding mode observers, it would be possible to refine the experimental results so that they more closely match the simulation results. As well, if manipulators with less friction were available, this would allow for a more accurate dynamic model. This, too, would aid in further verifying the experimental implementation of the algorithm.

Due to the incorporation of time delays in the bilateral teleoperation system, the hybrid matrix (3.103) contains terms of the form $e^{-\tau s}$ for some time delay τ. In an ideal teleoperator without time delay, these terms would be unity. Some researchers have incorporated predictive approaches into teleoperation algorithms in order to improve performance. Typically, approaches such as Kalman filters [46, 48, 81], time forward observers [84], and Smith predictors [47, 50, 61, 71] are used as predictors. While this is a challenging task, especially for situations where the slave comes into and out of contact with the environment, these approaches have been shown to improve performance. Future work related to the algorithm presented in this thesis could involve the addition of a predictive strategy to compensate for time delays in the communications. This work would present theoretical challenges in terms of proving the stability of a predictive strategy, but may allow for improved performance. On the usability side, user testing could be performed to compare the current non-predictive algorithm with a predictive version in order to determine if improvements in task completion times can be seen when prediction is employed.

While the sliding mode observers used in this work effectively estimate the states and unknown inputs of the robot manipulators, it is well known that they do not function as well in the presence of measurement noise. There are other forms of estimators that are much better equipped to handle this noise. The Kalman filter and extended Kalman filter work very well in the presence of Gaussian measurement noise. The particle filter can provide accurate estimates in more general cases where the noise is not Gaussian. However, to the best of the author's knowledge, these estimators do not provide estimation of the unknown inputs to a system. As an extension to the algorithm presented in this work, the development of estimators that provide unknown input estimates and work well in the presence of measurement noise would be useful. This would be challenging theoretically, in terms of developing such estimators, but also in re-deriving proofs of stability for the bilateral teleoperation algorithm with a different unknown input observer. The unknown input sliding mode observers used in this work provide estimates that converge to the true states in finite time. Other observers, such as the Kalman filter, yield asymptotically stable error dynamics. The change from finite time convergent estimators to asymptotically convergent ones would add a layer of difficulty in the closed loop stability proof for the teleoperator. Experimental implementations of the teleoperation algorithm comparing the effectiveness of an estimator that is more tolerant of noisy measurements to the current algorithm would be worthwhile.

Section 3.5 presented delay-dependent stability results assuming a linear slave side environment and commensurate time delays. It was shown that, using this result, one could be less conservative with the choices of the closed loop impedance parameters for the master and slave manipulators. However, the method proposed in that section does not yield a constructive result in terms of selecting closed loop parameters to yield a particular time delay. Instead, the designer may choose a set of closed loop impedance parameters, and use the result of the section to determine an upper bound on the time delay before which the closed loop system will go unstable. It would be interesting to develop an approach that would operate in the other direction – given an upper bound on the time delay, what is a suitable choice of closed loop impedance parameters? In practice, one may have some idea of the time delay, so selecting impedance parameters based on a maximum time delay makes sense. Given that the existing approach to compute the upper bound on the time delay is numerical, it may be possible to develop an efficient numerical algorithm to instead allow selection of the closed loop parameters based on a desired time delay margin.

In practice, the dynamics of a human operator are quite complicated and not always well understood. Some of the teleoperation researchers have chosen to model the human in the system as a passive system. In the work presented here, the human is modeled with a finite-gain stable nonlinear operator. Further work to better understand the behaviour of the human operator within the teleoperation loop would be of value. It would be useful to examine the assumptions made on human operator dynamics in the teleoperation literature and attempt to develop more accurate human models, both for simulation and analysis of the teleoperation algorithms. The effect of the human in the closed loop on the stability of the algorithm would also be a useful topic for further study.

Teleoperation is a particularly applications-oriented area within control and robotics. As a result, the development of algorithms that work particularly well in certain applications is a useful task. The algorithm presented here is designed to apply generally to teleoperation tasks. There are certain cases where it may perform particularly well over other algorithms. This algorithm provides ideal force tracking without the need for force sensing. As a result, applications where force feedback is desired, but where it would be difficult to include a force sensor, seem particularly well fitted to the algorithm. For instance, micro-manipulation, where the manipulator could be quite small, may benefit from an algorithm such as this. It may be difficult to mount a force sensor on a very small manipulator, but force signals may still be required. The investigation of the application of this algorithm in such areas would be useful. Another applications oriented task for future work could involve tailoring the choice of closed loop impedance parameters for certain applications, and determining the stability of the resulting closed loop system. This would allow for the application of the algorithm to a variety of scenarios. The practical performance of the algorithm for completing tasks on a variety of scales (for example, very small and very large manipulators with a variety of environments) with impedance parameters that have been selected *a priori* could then be evaluated. In particular, specific application areas such as hazardous environments and biomedical use including remote medical surgeries could be investigated, and the usefulness of this algorithm in those areas could be evaluated.

Appendices

Appendix A

Proofs

A.1 Proof of Theorem 2.2.1

Proof. The proof of this theorem follows that of [24] very closely, with the exceptions that a slight modification of the observer design is made in this work and upper bounds on convergence time are computed here. Initially, when $\hat{x}(t_0) \neq x(t_0)$, the observer error dynamics take the form of (2.32) where all the $E_i = 0$.

As a result of Assumptions 2.2.1, 2.2.3, and 2.2.4 the estimation error does not have a finite escape time. This effectively guarantees that the estimation error is in the space L_{pe}. Consider the following Lyapunov function candidate for the state e_1,

$$V_1 = \frac{1}{2}e_1^2 \tag{A.1}$$

Then,

$$\dot{V}_1 = e_1\dot{e_1} = -\lambda_1|e_1| + e_1e_2 \leq -\lambda_1|e_1| + |e_1||e_2| = -|e_1|(\lambda_1 - |e_2|) \tag{A.2}$$

One could choose $\lambda_1 > |e_2| + \varepsilon_1$, where $\varepsilon_1 > 0$. Then, $\dot{V}_1 \leq -|e_1|\varepsilon_1$. This would ensure that $\dot{V}_1 < 0$ for all $e_1 \neq 0$, and also ensure finite time convergence of e_1 to the sliding surface

$e_1 = 0$. Finite time convergence is assured through the use of the Comparison Lemma. Alternatively, choosing $\lambda_1 > \max_{t\in[t_0,T]} |e_2|$ where T is greater than or equal to the time it takes for e_1 to converge to the sliding surface, allows a fixed value for λ_1 and also ensures that $\dot{V}_1 < 0$ for all $e_1 \neq 0$. Therefore, the state e_1 will converge to zero. In order to show that the convergence happens in finite time, define the function $W_1 = \sqrt{2V_1} = |e_1|$. Here, the choice of gain $\lambda_1 > |e_2| + \varepsilon_1$ is considered. Then,

$$D^+W_1 = \dot{W}_1 = \frac{1}{\sqrt{2V_1}}\dot{V}_1 = \frac{1}{|e_1|}\dot{V}_1 \leq \frac{-|e_1|}{|e_1|}\varepsilon_1 = -\varepsilon_1 \tag{A.3}$$

So $D^+W_1 \leq -\varepsilon_1$. Define $f_1(t, W_1) = -\varepsilon_1$. Then $D^+W_1 \leq f_1(t, W_1)$. Now consider $\dot{u} = f_1(t, u) = -\varepsilon_1$, $u(t_0) = W_1(e_1(t_0))$. Then,

$$\begin{aligned}
\int_{t_0}^{t} \frac{du}{d\tau} d\tau &= -\varepsilon_1 \int_{t_0}^{t} d\tau && \text{(A.4)}\\
u(t) - u(t_0) &= -\varepsilon_1(t - t_0) && \text{(A.5)}\\
u(t) &= u(t_0) - \varepsilon_1(t - t_0) && \text{(A.6)}
\end{aligned}$$

Making use of Lemma 2.2.1, the result $W_1(t) \leq u(t)$ is obtained. Then,

$$W_1(t) \leq W_1(e_1(t_0)) - \varepsilon_1(t - t_0) \tag{A.7}$$

Since $W_1(t) = |e_1(t)|$, it is clear from the above that there will be some finite time when $|e_1|$ reaches zero. In order to solve for the upper bound on this time, set $W_1(e_1(t_0)) - \varepsilon_1(t - t_0) = 0$. Rearranging, one arrives at,

$$T_1 \triangleq \frac{|e_1(t_0)|}{\varepsilon_1} + t_0 \tag{A.8}$$

Therefore by the time $t = T_1$ the state e_1 will have converged to the sliding surface $e_1 = 0$. On that surface $e_1 = \dot{e}_1 = 0$ and solving for the equivalent output injection yields $(\lambda_1 \text{sgn}(e_1))_{eq} = e_2$. This results in $\tilde{x}_2 = x_2$ on the sliding surface. Following [21], the equivalent control is obtained by a low pass filtering operation of $\lambda_1\text{sgn}(e_1)$.

After time T_1 the error dynamics have the form,

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \vdots \\ \dot{e}_{n-1} \\ \dot{e}_n \end{bmatrix} = \begin{bmatrix} e_2 - \lambda_1 \mathrm{sgn}(e_1) = 0 \\ e_3 - \lambda_2 \mathrm{sgn}(e_2) \\ e_4 \\ \vdots \\ e_n \\ \Delta h(x, \hat{x}) + \Delta g(x, \hat{x})u + g(x)d \end{bmatrix} \tag{A.9}$$

(Recall that on the surface $e_1 = 0$, $\tilde{x}_2 = x_2$ and so $\tilde{x}_2 - \hat{x}_2 = x_2 - \hat{x}_2 = e_2$. As well, the term E_1 is now set to one since $e_1 = 0$, but the terms E_i for $i > 1$ remain zero.) Showing convergence of the second observer error state may now proceed. From above, since the observer error is in L_{pe} and a finite time T_1 has elapsed, the state e_2 remains finite at this point by Assumption 2.2.1. Choosing as a Lyapunov function candidate for this system,

$$V_2 = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 = \frac{1}{2}e_2^2 \tag{A.10}$$

The e_1 term drops out since after $t = T_1$, $e_1 = 0$ and the Lyapunov function simplifies, as above. Taking the derivative along the trajectories of the system,

$$\dot{V}_2 = e_2\dot{e_2} = -\lambda_2|e_2| + e_2e_3 \leq -\lambda_2|e_2| + |e_2||e_3| = -|e_2|(\lambda_2 - |e_3|) \tag{A.11}$$

Choosing $\lambda_2 > |e_3| + \varepsilon_2$, $\varepsilon_2 > 0$, yields $\dot{V}_2 \leq -|e_2|\varepsilon_2$, for all $e_2 \neq 0$. Alternatively, choosing a fixed gain $\lambda_2 > \max_{t\in(T_1,T]} |e_3|$, for some T greater than or equal to the time it takes for e_2 to converge to the siding surface $e_2 = 0$, would also ensure that $\dot{V}_2 < 0$ for all $e_2 \neq 0$. With either choice of gain, the state e_2 will converge to zero in finite time. It can be verified that, when the gain is chosen as $\lambda_2 > |e_3| + \varepsilon_2$, the bound is,

$$T_2 \triangleq \frac{|e_2(T_1)|}{\varepsilon_2} + T_1 \tag{A.12}$$

Once the trajectory has converged to the surface $e_2 = 0$, one has $\dot{e}_2 = 0$ and the equivalent output injection term $(\lambda_2 \mathrm{sgn}(e_2))_{eq} = e_3$. At this point, E_2 is set to one, since

$e_2 = 0$. This equivalent value is obtained in practice through a low pass filtering operation on $\lambda_2 \text{sgn}(e_2)$. At this point, $\tilde{x}_3 = \hat{x}_3 + e_3 = x_3$. This procedure is carried out for each observer error state up to and including e_{n-1} in order to show that a gain λ_i exists at each step and has a finite upper bound on the time for the state to converge to the particular sliding surface. In general, the gain λ_i can be chosen as $\lambda_i > |e_{i+1}| + \varepsilon_i$, $\varepsilon_i > 0$, for all $i = 1, \ldots, n-1$. Once the sliding surfaces $e_1 = e_2 = \ldots = e_{n-1} = 0$ have been reached and the corresponding anti-peaking terms have been set appropriately, an analysis of the stability of the final state e_n may be performed. The upper bound on the time required for the first $n-1$ states to converge to their sliding surfaces is T_{n-1}. At this point, the error dynamics have the form,

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_{n-1} \\ \dot{e}_n \end{bmatrix} = \begin{bmatrix} e_2 - \lambda_1 \text{sgn}(e_1) = 0 \\ e_3 - \lambda_2 \text{sgn}(e_2) = 0 \\ \vdots \\ e_n - \lambda_{n-1} \text{sgn}(e_{n-1}) = 0 \\ \Delta h(x, \hat{x}) + \Delta g(x, \hat{x}) u + g(x)\xi - \lambda_n \text{sgn}(e_n) \end{bmatrix} \tag{A.13}$$

In order to show stability for the final observer state consider the Lyapunov function candidate,

$$V_n = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \ldots + \frac{1}{2}e_n^2 = \frac{1}{2}e_n^2 \tag{A.14}$$

since $e_i = 0$ for all $i < n$ at time T_{n-1}. Taking the derivative of V_n along the trajectories of the system,

$$\begin{aligned} \dot{V}_n &= e_n \dot{e}_n & \text{(A.15)} \\ &= e_n(\Delta h(x, \hat{x}) + \Delta g(x, \hat{x}) u + g(x)\xi - \lambda_n \text{sgn}(e_n)) & \text{(A.16)} \\ &= -\lambda_n |e_n| + e_n(\Delta h(x, \hat{x}) + \Delta g(x, \hat{x}) u + g(x)\xi) & \text{(A.17)} \\ &\leq -\lambda_n |e_n| + |e_n|(|\Delta h(x, \hat{x}) + \Delta g(x, \hat{x}) u + g(x)\xi|) & \text{(A.18)} \\ &= -|e_n|(\lambda_n - |\Delta h(x, \hat{x}) + \Delta g(x, \hat{x}) u + g(x)\xi|) & \text{(A.19)} \end{aligned}$$

Choosing $\lambda_n > |\Delta h(x,\hat{x}) + \Delta g(x,\hat{x})u + g(x)\xi| + \varepsilon_n$, $\varepsilon_n > 0$ yields $\dot{V}_n \leq -|e_n|\varepsilon_n < 0$, for all $e_n \neq 0$ This ensures finite time convergence to the sliding surface $e_n = 0$. One could alternatively choose a fixed value for λ_n as $\lambda_n > \max_{t\in[T_{n-1},T)} |\Delta h(x,\hat{x}) + \Delta g(x,\hat{x})u + g(x)d_0|$, for some T greater than or equal to the time required for the state e_n to reach the sliding surface. It can be verified that with the choice of the gain as $\lambda_n > |\Delta h(x,\hat{x}) + \Delta g(x,\hat{x})u + g(x)d| + \varepsilon_n$ an upper bound is,

$$T_n \triangleq \frac{|e_n(T_{n-1})|}{\varepsilon_n} + T_{n-1} \tag{A.20}$$

Finally, after time T_n has passed, the observer error trajectories lie on the surface $e_1 = \ldots = e_n = 0$, and an estimate of the unknown input d can be obtained from the equivalent output injection. To compute the equivalent output injection, solve the above $\dot{e}_n$ equation for $\lambda_n \text{sgn}(e_n)$ when $e_n = \dot{e}_n = 0$. This results in $\Delta h(x,\hat{x}) = \Delta g(x,\hat{x}) = 0$, since $\Delta h(x,\hat{x}) = h(x) - h(\hat{x})$, $\Delta g(x,\hat{x}) = g(x) - g(\hat{x})$. As well,

$$(\lambda_n \text{sgn}(e_n))_{eq} = g(x)\xi \tag{A.21}$$

So the unknown input ξ is computed as $\xi = g(x)^{-1}(\lambda_n \text{sgn}(e_n))_{eq}$. (Recall that, on the surface $e_1 = \ldots = e_n = 0$ the state vector x is available since the observer error is zero, so $g(x)$ may be computed.) The equivalent output injection is obtained by a low pass filtering operation on $\lambda_n \text{sgn}(e_n)$ [21]. This concludes the proof, showing finite time convergence of the state estimates to the actual states and providing an estimate of the unknown disturbance as well. □

A.2 Proof of Theorem 2.2.2

Proof. In order to prove convergence of the state estimates to the real states, it is necessary to first prove the convergence of $\tilde{x}_1$ and $\dot{\tilde{x}}_1$ to zero. As follows from the observer error dynamics and (2.45), the estimation error $\tilde{x}_1$ and $\tilde{x}_2$ satisfy the differential inclusion,

$$\begin{aligned} \dot{\tilde{x}}_1 &= \tilde{x}_2 - \lambda |\tilde{x}_1|^{1/2} \text{sign}(\tilde{x}_1) && \text{(A.22)} \\ \dot{\tilde{x}}_2 &\in [-f^+, +f^+] - \alpha \text{sign}(\tilde{x}_1) && \text{(A.23)} \end{aligned}$$

Note that the solutions of (A.22), (A.23) exist in the sense of Fillipov for any initial condition. Next, compute $\ddot{\tilde{x}}_1$ when $\tilde{x}_1 \neq 0$.

$$\ddot{\tilde{x}}_1 = \dot{\tilde{x}}_2 - \frac{1}{2} \lambda |\tilde{x}_1|^{-1/2} \dot{\tilde{x}}_1$$

where the identity $d|x|/dt = \dot{x}\text{sign}(x)$ is used.

From (A.23) it is possible to express $\ddot{\tilde{x}}_1$ as the following inclusion,

$$\ddot{\tilde{x}}_1 \in [-f^+, +f^+] - \left(\frac{1}{2} \lambda \frac{\dot{\tilde{x}}_1}{|\tilde{x}_1|^{1/2}} + \alpha \text{sign}(\tilde{x}_1) \right) \tag{A.24}$$

The approach taken in what follows is to show the worst case error trajectory in each quadrant of the $\tilde{x}_1$-$\dot{\tilde{x}}_1$ plane. This trajectory of the worst case system is called the majorant trajectory. See Figure A.1.

In the first quadrant, $\tilde{x}_1 > 0$ and $\dot{\tilde{x}}_1 > 0$. The trajectory starts at $\tilde{x}_{1_0} = 0$ and $\dot{\tilde{x}}_{1_0} = x_2$. From (A.24) the worst case acceleration in this quadrant, that is the largest it can be, may be found. Note that,

$$\ddot{\tilde{x}}_1 \in [-f^+, +f^+] - \alpha - \underbrace{\left(\frac{1}{2} \lambda \frac{\dot{\tilde{x}}_1}{|\tilde{x}_1|^{1/2}} \right)}_{\text{positive}}$$

The last term in brackets is always positive in this quadrant, as a result it will only make the acceleration more negative. So the largest that the acceleration can be is given by,

$$\ddot{\tilde{x}}_1 = \underbrace{f^+ - \alpha}_{\text{negative}} \tag{A.25}$$

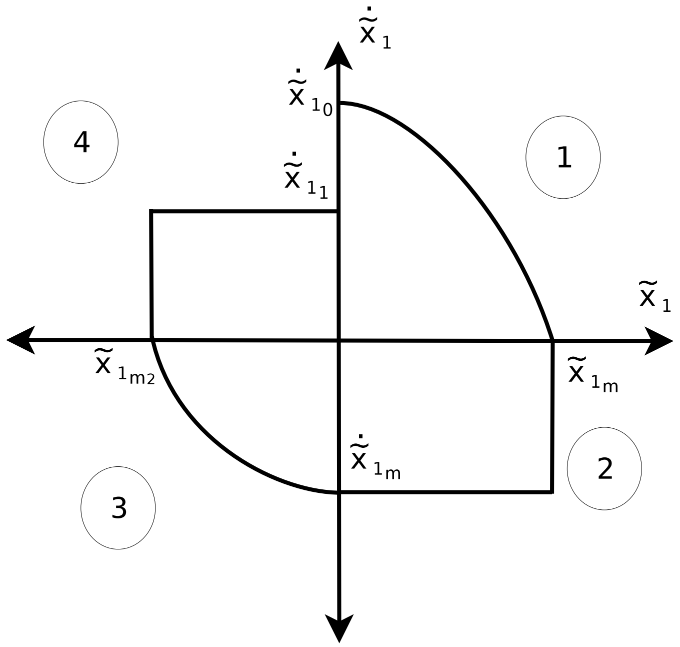


Figure A.1: Worst case trajectory for observer error dynamics.

So the trajectory is confined by the axis $\tilde{x}_1 = 0$, $\dot{\tilde{x}}_1 = 0$ and the equation (A.25). Since $\alpha > f^+$, the acceleration $\ddot{\tilde{x}}_1 < 0$, and so $\dot{\tilde{x}}_1$ decreases toward 0. The trajectory of the states in this quadrant is found by solving (A.25). Integrating both sides of (A.25) one gets,

$$\begin{aligned}
\int_{\dot{\tilde{x}}_{1_0}}^{\dot{\tilde{x}}_1} \ddot{\tilde{x}}_1 dt &= -\int_0^t (\alpha - f^+) d\tau \\
\dot{\tilde{x}}_1 - \dot{\tilde{x}}_{1_0} &= -(\alpha - f^+)t \\
\dot{\tilde{x}}_1 &= -(\alpha - f^+)t + \dot{\tilde{x}}_{1_0} \qquad \text{(A.26)}
\end{aligned}$$

Now integrating (A.26) to get the equation of motion for $\tilde{x}_1$, one arrives at,

$$\int_{\tilde{x}_{1_0}}^{\tilde{x}_1} \dot{\tilde{x}}_1 dt = -\int_0^t (\alpha - f^+)\tau d\tau + \int_0^t \dot{\tilde{x}}_{1_0} d\tau \tag{A.27}$$

$$\tilde{x}_1 - \tilde{x}_{1_0} = -(\alpha - f^+)\frac{t^2}{2} + \dot{\tilde{x}}_{1_0} t \tag{A.28}$$

$$\tilde{x}_1 = -(\alpha - f^+)\frac{t^2}{2} + \dot{\tilde{x}}_{1_0} t + \tilde{x}_{1_0} \tag{A.29}$$

Re-arranging (A.26) gives,

$$t = \frac{\dot{\tilde{x}}_{1_0} - \dot{\tilde{x}}_1}{\alpha - f^+} \tag{A.30}$$

Substituting (A.30) into (A.29) and simplifying yields,

$$\dot{\tilde{x}}_1^2 - \dot{\tilde{x}}_{1_0}^2 = 2(\alpha - f^+)(\tilde{x}_{1_0} - \tilde{x}_1) \tag{A.31}$$

This curve is shown in quadrant 1 in Figure A.1.

Defining the point where the trajectory reaches the axis $\dot{\tilde{x}}_1 = 0$ as $\tilde{x}_{1_m}$, the relationship between the initial point $(0, \dot{\tilde{x}}_{1_0})$ and $(\tilde{x}_{1_m}, 0)$ may be obtained from (A.31) as,

$$\dot{\tilde{x}}_{1_0}^2 = 2(\alpha - f^+)\tilde{x}_{1_m} \tag{A.32}$$

Since $\ddot{\tilde{x}}_1 < 0$ for all of quadrant 1, the curve will hit the $\dot{\tilde{x}}_1 = 0$ axis in finite time. This moves the trajectory into quadrant 2.

In quadrant 2, $\dot{\tilde{x}}_1 \leq 0$ and $\tilde{x}_1 > 0$. Here the worst case trajectory of the estimation error is given by the equation,

$$\ddot{\tilde{x}}_1 = \underbrace{-f^+ - \alpha \text{sign}(\tilde{x}_1)}_{\text{negative}} - \underbrace{\frac{1}{2}\lambda|\tilde{x}_1|^{-1/2}\dot{\tilde{x}}_1}_{\text{negative}} \tag{A.33}$$

Note that in this quadrant the second term in (A.33) must be included in the worst case trajectory. With the first term being negative, subtracting the second negative term from it will make the acceleration larger. So the worst case trajectory is made up of both terms on the right hand side of (A.33). This equation is negative for a proper choice of α. As a result, the largest value of $\tilde{x}_1$ in quadrant 2 is $\tilde{x}_{1_m}$. As well, since $\ddot{\tilde{x}}_1 < 0$, one arrives at,

$$-\frac{1}{2}\lambda|\tilde{x}_1|^{-1/2}\dot{\tilde{x}}_1 < f^+ + \alpha \tag{A.34}$$

Re-arranging for $\dot{\tilde{x}}_1$ gives,

$$\dot{\tilde{x}}_1 > -\frac{2}{\lambda}|\tilde{x}_1|^{1/2}(f^+ + \alpha) \tag{A.35}$$

Since $\tilde{x}_1 \leq \tilde{x}_{1_m}$, the velocity $\dot{\tilde{x}}_1$ has the following lower bound defined as $\dot{\tilde{x}}_{1_m}$,

$$\dot{\tilde{x}}_{1_m} = -\frac{2}{\lambda}|\tilde{x}_{1_m}|^{1/2}(f^+ + \alpha) \tag{A.36}$$

So the system trajectory will cross the axis $\tilde{x}_1 = 0$ at $\dot{\tilde{x}}_1 \geq \dot{\tilde{x}}_{1_m}$, as in Figure A.1. The trajectory of the error dynamics will continue to cross that axis as the trajectory continues in time. Due to symmetry between the first and third quadrants, and symmetry between the second and fourth quadrants, a similar result is obtained when $\tilde{x}_1 < 0$.

In order to show convergence of the trajectory, one must show that $|\dot{\tilde{x}}_{1_m}|/|\dot{\tilde{x}}_{1_0}| < 1$. Each of the numerator and denominator are defined as,

$$\begin{aligned} |\dot{\tilde{x}}_{1_0}| &= \sqrt{2(\alpha - f^+)\tilde{x}_{1_m}} && \text{(A.37)} \\ |\dot{\tilde{x}}_{1_m}| &= \frac{2}{\lambda}\tilde{x}_{1_m}^{1/2}(f^+ + \alpha) && \text{(A.38)} \end{aligned}$$

Then,

$$\frac{|\dot{\tilde{x}}_{1_m}|}{|\dot{\tilde{x}}_{1_0}|} = \frac{2(f^+ + \alpha)}{\lambda\sqrt{2(\alpha - f^+)}} \tag{A.39}$$

Substituting the definition of λ from (2.47) into the above and simplifying yields,

$$\frac{|\dot{\tilde{x}}_{1_m}|}{|\dot{\tilde{x}}_{1_0}|} < \frac{1-p}{1+p} < 1 \tag{A.40}$$

Denoting the consequent crossing points of the system trajectory starting at $(0, \dot{\tilde{x}}_{1_0})$ with the $\tilde{x}_1 = 0$ axis as $\dot{\tilde{x}}_{1_0}, \dot{\tilde{x}}_{1_1}, \dot{\tilde{x}}_{1_2}, \ldots, \dot{\tilde{x}}_{1_i}, \ldots$, the following result, due to (A.40), will hold,

$$\frac{|\dot{\tilde{x}}_{1_{i+1}}|}{|\dot{\tilde{x}}_{1_i}|} < \frac{1-p}{1+p} < 1 \tag{A.41}$$

This inequality ensures the convergence of the state to $\tilde{x}_1 = \dot{\tilde{x}}_1 = 0$ and ensures the convergence of the geometric series $\sum_0^\infty |\dot{\tilde{x}}_i|$.

In order to show finite time convergence and determine an upper bound on the time, consider the dynamics of $\tilde{x}_2$. From (A.23),

$$0 < \alpha - f^+ \leq |\dot{\tilde{x}}_2| \leq \alpha + f^+$$

which leads to,

$$|\dot{\tilde{x}}_2| \geq \alpha - f^+ \tag{A.42}$$

Let t_i represent the time of the i-th crossing of $\dot{\tilde{x}}_1$ with the axis $\tilde{x}_1 = 0$. At these crossing points, that is when $\tilde{x}_1 = 0$, $\tilde{x}_2 = \dot{\tilde{x}}_1$. So $\tilde{x}_2$ crosses the axis at time t_i as well. Integrating both sides of (A.42) from the time of the $(i-1)$-th axis crossing to the time of the i-th axis crossing,

$$\begin{aligned}
\int_{t_{i-1}}^{t_i} |\dot{\tilde{x}}_2| d\tau &\geq \int_{t_{i-1}}^{t_i} (\alpha - f^+) d\tau && \text{(A.43)}\\
|\tilde{x}_2(t_i)| - |\tilde{x}_2(t_{i-1})| &\geq (\alpha - f^+) \underbrace{(t_i - t_{i-1})}_{\Delta t_i} && \text{(A.44)}\\
|\tilde{x}_2(t_i)| &\geq (\alpha - f^+)\Delta t_i \\
&\quad + |\tilde{x}_2(t_{i-1})| && \text{(A.45)}\\
|\tilde{x}_2(t_i)| &\geq (\alpha - f^+)\Delta t_i && \text{(A.46)}
\end{aligned}$$

So the upper bound on the time between the i-th and $(i-1)$-th axis crossing is given as,

$$\Delta t_i \leq \frac{|\tilde{x}_2(t_i)|}{(\alpha - f^+)} \tag{A.47}$$

An upper bound for the total convergence time is given as,

$$\begin{aligned} T &\leq \sum_{0}^{\infty} \Delta t_i && \text{(A.48)} \\ T &\leq \sum_{0}^{\infty} \frac{|\tilde{x}_2(t_i)|}{(\alpha - f^+)} && \text{(A.49)} \end{aligned}$$

Note that, due to convergence of the geometric series, the right hand side of (A.49) exists and is finite.

The only requirement for the implementation of this observer is the boundedness of $F(t, x_1, x_2, \hat{x}_2)$ in the domain of interest. One way to ensure that this is always true is to design the observer to converge in a fast enough time that the system has not left some pre-specified area, in which stabilization can be ensured [31]. This approach will ensure boundedness of the system within the desired operating range.

Finally, to show that once the trajectory has reached $(0, \dot{\tilde{x}}_{1_m})$, it will return to cross the axis $\tilde{x}_1 = 0$ when $\dot{\tilde{x}}_1 > 0$, consider the worst case trajectories in the final two quadrants. In quadrant 3, where $\tilde{x}_1 < 0$ and $\dot{\tilde{x}}_1 \leq 0$ the acceleration must be positive in order to drive the velocity, $\dot{\tilde{x}}_1$, back to zero. The boundary given by the differential inclusion is,

$$\ddot{\tilde{x}}_1 = \underbrace{-f^+ + \alpha}_{\text{positive}} - \underbrace{\frac{1}{2}\lambda \frac{\dot{\tilde{x}}_1}{|\tilde{x}_1|^{1/2}}}_{\text{negative}} \tag{A.50}$$

The last term on the right hand side of (A.50) will only make the acceleration more positive, so for the worst case acceleration (the smallest value that the acceleration will be) that term is disregarded. This gives the following worst case trajectory for quadrant 3,

$$\ddot{\tilde{x}}_1 = \underbrace{\alpha - f^+}_{\text{positive}} \tag{A.51}$$

This will always be positive and will drive the system trajectory back up to $\dot{\tilde{x}}_1 = 0$. Define the point where the system intersects with the axis $\dot{\tilde{x}}_1 = 0$ as $\tilde{x}_{1_{m2}}$.

At this point the trajectory will move into quadrant 4, where $\tilde{x}_1 < 0$, $\dot{\tilde{x}}_1 \geq 0$. Again it must be ensured that $\ddot{\tilde{x}}_1 > 0$. Here the boundary of the differential inclusion is given as,

$$\ddot{\tilde{x}}_1 = \underbrace{-f^+ + \alpha}_{\text{positive}} - \underbrace{\frac{1}{2}\lambda \frac{\dot{\tilde{x}}_1}{|\tilde{x}_1|^{1/2}}}_{\text{positive}} \tag{A.52}$$

The last term on the right hand side of (A.52) will make the acceleration smaller, so it must be considered in the worst case. Note that for a suitable choice of α, $\ddot{\tilde{x}}_1 > 0$ and $\tilde{x}_1$ will continue to increase until $\tilde{x}_1 = 0$. The largest absolute value of $\tilde{x}_1$ is $\tilde{x}_{1_{m2}}$ in this quadrant. An expression for the upper bound on the velocity may be be found from the inequality $\ddot{\tilde{x}}_1 > 0$. In this case,

$$-f^+ + \alpha - \frac{1}{2}\lambda \frac{\dot{\tilde{x}}_1}{|\tilde{x}_1|^{1/2}} > 0 \tag{A.53}$$

Rearranging for $\dot{\tilde{x}}_1$,

$$\dot{\tilde{x}}_1 < \frac{2}{\lambda}(\alpha - f^+)|\tilde{x}_1|^{1/2} \tag{A.54}$$

Since $|\tilde{x}_1| \leq |\tilde{x}_{1_{m2}}|$, the upper bound on the velocity, defined as $\dot{\tilde{x}}_{1_1}$, is given as,

$$\dot{\tilde{x}}_{1_1} = \frac{2}{\lambda}(\alpha - f^+)|\tilde{x}_{1_{m2}}|^{1/2} \tag{A.55}$$

At this point the trajectory returns to $(0, \dot{\tilde{x}}_{1_1})$ where $\dot{\tilde{x}}_{1_1} > 0$. It has already been shown that, for suitably chosen λ, the trajectory will continue this pattern of twisting around the origin, eventually converging in finite time.

In order to show recovery of the unknown term $\xi(t, x_1, x_2, u)$ in the dynamics, consider the error dynamics (2.43) and (2.44). After T seconds, it is guaranteed that $\tilde{x}_1 = \dot{\tilde{x}}_1 = \tilde{x}_2 = 0$. At this point,

$$F(t, x_1, x_2, \hat{x}_2, u) = \xi(t, x_1, x_2, u)$$

so (2.44) reduces to,

$$\xi(t, x_1, x_2, u) = \alpha \text{sign}(\tilde{x}_1) \tag{A.56}$$

The estimate of the unknown input to the dynamics is recovered from the equivalent value of the discontinuous term on the right hand side of (A.56). The equivalent signal may be obtained by passing this term through a low pass filter.

□

Appendix B

Experimental Platform

The bilateral teleoperation algorithm presented in this work has been implemented experimentally on the University of Waterloo Teleoperation Platform. This platform consists of two 3-DOF direct-drive robot manipulators. Each manipulator is a five bar linkage that operates in the vertical plane with an additional revolute joint at the base. These robots were designed and manufactured at the University of Waterloo. The use of direct-drive motors helps to eliminate issues such as stiction and backlash, which can be difficult to account for in the dynamic models. Figure B.1 shows the master manipulator, while the slave is shown in Figure B.2.

The experimental implementation makes use of only one degree of freedom on each robot. For each manipulator, the two joints that actuate the five bar linkage were rigidly locked. The base joint of each manipulator was actuated and used as the only degree of freedom in the experiments. As a result of this, the dynamics of each manipulator can be modeled as an inertia rotating in the horizontal plane with damping and friction. Due to the hardware setup and available equipment, is was not possible to implement a system where more than a total of three degrees of freedom between both robots could be actuated at the same time. These limitations were caused by the available hardware. There was only a power supply for three motor amplifiers, and the computer that was used could only accommodate one data acquisition card. The use of only one degree

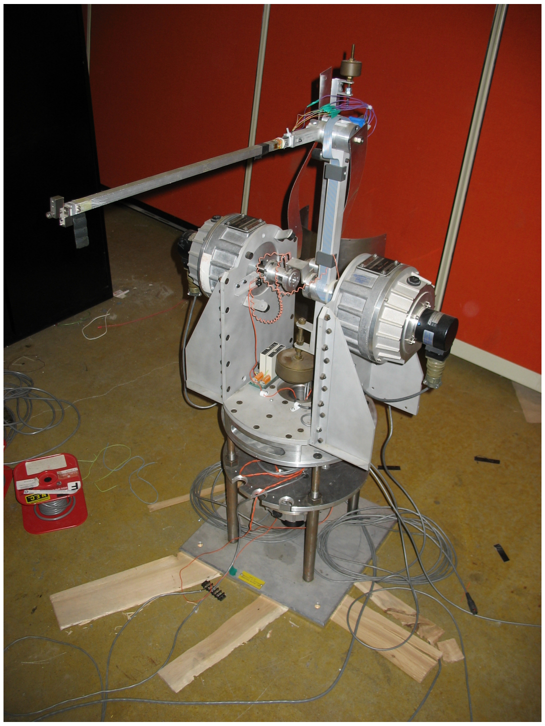


Figure B.1: Master manipulator.

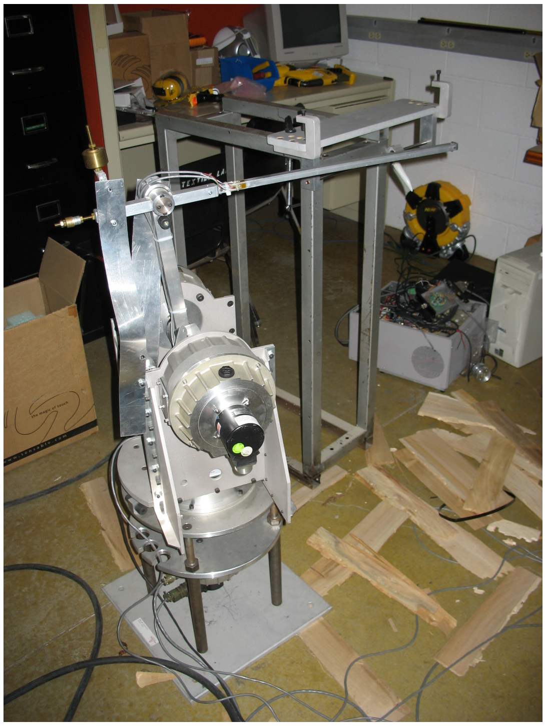


Figure B.2: Slave manipulator.

of freedom per robot simplifies controller design in that it is not necessary to have a Jacobian or to convert the robot dynamics into task space. This 1-DOF experimental setup, along with 2-DOF simulation results, instills confidence that the algorithm will function as expected with multiple degree of freedom setups. It is possible in practice that other issues, such as unmodeled coupling between the degrees of freedom, will come out. However, an experimental study of this situation is left for future work.

Data Acquisition Interface

The robots are connected to a desktop computer through PCI based data acquisition hardware. This hardware is a Sensoray Model 626 Analog and Digital I/O interface card. The card supports up to six encoders with a resolution of 24 bits each, four 14-bit D/A outputs, sixteen 16-bit differential A/D inputs, and 48 digital I/O channels. For the experimental setup used for this work, the digital I/O lines are not used.

In order to interface the robots to the data acquisition card, a Quanser Consulting PCI MultiQ VI terminal board was used. Ribbon cables from the Sensoray card are connected to the terminal board to provide easy access to the I/O lines of the card.

Motors and Amplifiers

Each degree of freedom on the robots is driven by a Kollmorgen ServoDisc DC motor. These motors are rated to produce a maximum of 1.41 horsepower. Position encoders are fixed to each motor in order to give angular position measurements. Advanced Motion Controls Series 50A servo amplifiers are used to power each of the motors. These amplifiers generate PWM signals to drive the motors. Limit switches are placed on the robots at the edge of the safe work envelope and connected to the fault lines of the amplifiers in order to stop the motors should the robot move outside of the safe region.

Control Software

In order to implement the bilateral control algorithm, Simulink was used. Matlab has built-in support for the Sensorary 626 card, making it quite straightforward to interface with. The decision to use Simulink was made because of the availability of Real-Time Workshop and the Real-Time Windows Target. This software makes it possible to run Simulink models in real-time on a Windows PC, greatly simplifying the process of experimentally verifying the algorithm. The support available through Matlab allows the implementation to be done directly in Simulink. Compiled code is automatically produced and interfaces with the hardware. An environment of this nature makes it possible to focus on the algorithm and the theory, while being able to implement it in a straightforward way.

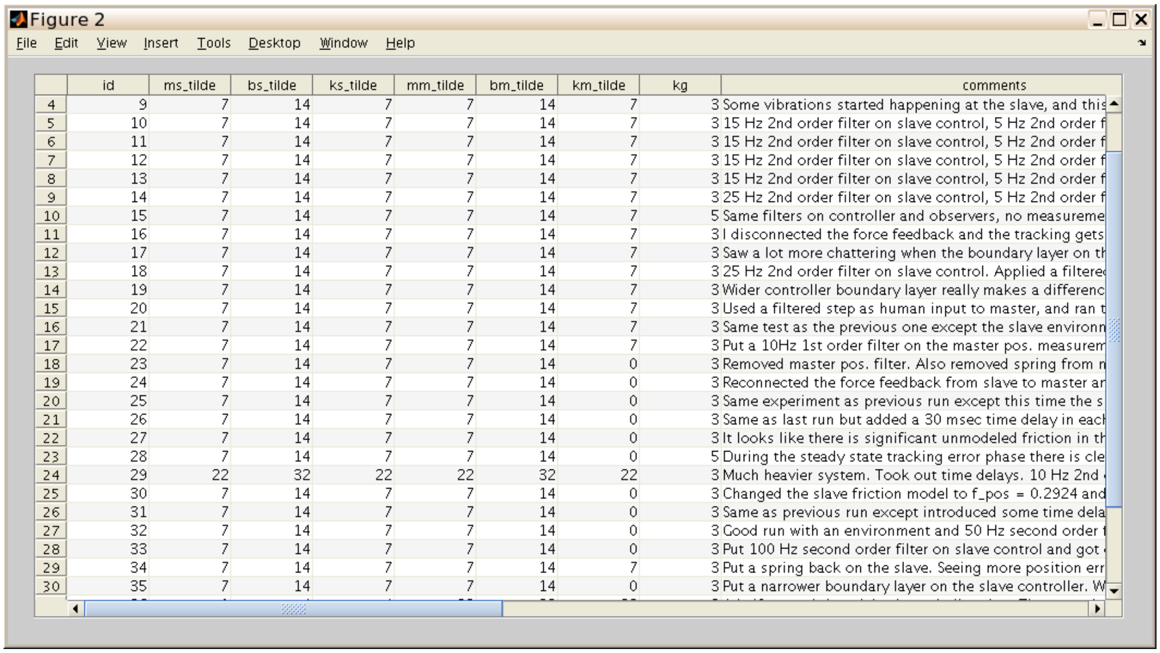
Within the Simulink model, the controllers and observers were implemented with Embedded Matlab blocks. These blocks allow the developer to write in a subset of the Matlab language from within the Simulink environment. In many cases it can be easier to write Matlab code to implement an algorithm. The various pieces of embedded code may be easily interfaced with the rest of the Simulink model. The code in the embedded blocks is compiled at build time, and executed in real-time within the model. All of the continuous time dynamics in the algorithm were integrated using a fifth order Runge-Kutta algorithm.

Experimental Data Collection and Storage

When performing experiments of this nature, there are many possible parameters that may be varied. For instance, controller gains, observer gains, time delays, sliding mode boundary layers, and filter bandwidths are all parameters that may be modified from one experiment to the next. This large number of parameters, combined with the fact that many sets of experiments must be performed, creates the need for a suitable method to store and organize the particular parameters used in each experimental run. While not common in this field, a relational database can be an ideal tool for this task.

In order to store the parameters of each experimental run, a database was designed using MySQL. The database is very straightforward in design. It consists of only one table with a column for each of the parameters that may be varied, as well as columns for experimental observations, and a reference to the data file for each experiment. Using the Matlab Database Toolbox, Matlab code has been written to interface with this database. In order to make use of this system, after completion of an experiment with the robots the database interface code is run. This code stores experimental data to a file and the parameters of the current experiment to the database. Several advantages exist to using a system such as this over merely storing experimental data files. The database allows the user to keep the parameters together with the data, as well as the set of observations made during each particular experimental run. Also, once the data is in a relational database, it is straightforward to perform searches and sorts on the experimental results, potentially displaying only certain subsets of the experimental runs. This would be useful, for instance, if the researcher wanted to examine only the set of results where a certain parameter was held at a particular value.

Another piece of code has been written in order to allow the viewing of the experimental results in a straightforward manner. Making use of the Matlab GUI development tools,



	id	ms_tilde	bs_tilde	ks_tilde	mm_tilde	bm_tilde	km_tilde	kg	comments
4	9	7	14	7	7	14	7	3	Some vibrations started happening at the slave, and this
5	10	7	14	7	7	14	7	3	15 Hz 2nd order filter on slave control, 5 Hz 2nd order f
6	11	7	14	7	7	14	7	3	15 Hz 2nd order filter on slave control, 5 Hz 2nd order f
7	12	7	14	7	7	14	7	3	15 Hz 2nd order filter on slave control, 5 Hz 2nd order f
8	13	7	14	7	7	14	7	3	15 Hz 2nd order filter on slave control, 5 Hz 2nd order f
9	14	7	14	7	7	14	7	3	25 Hz 2nd order filter on slave control, 5 Hz 2nd order f
10	15	7	14	7	7	14	7	5	Same filters on controller and observers, no measureme
11	16	7	14	7	7	14	7	3	I disconnected the force feedback and the tracking gets
12	17	7	14	7	7	14	7	3	Saw a lot more chattering when the boundary layer on th
13	18	7	14	7	7	14	7	3	25 Hz 2nd order filter on slave control. Applied a filtere
14	19	7	14	7	7	14	7	3	Wider controller boundary layer really makes a differenc
15	20	7	14	7	7	14	7	3	Used a filtered step as human input to master, and ran t
16	21	7	14	7	7	14	7	3	Same test as the previous one except the slave environn
17	22	7	14	7	7	14	7	3	Put a 10Hz 1st order filter on the master pos. measurem
18	23	7	14	7	7	14	0	3	Removed master pos. filter. Also removed spring from n
19	24	7	14	7	7	14	0	3	Reconnected the force feedback from slave to master ar
20	25	7	14	7	7	14	0	3	Same experiment as previous run except this time the s
21	26	7	14	7	7	14	0	3	Same as last run but added a 30 msec time delay in each
22	27	7	14	7	7	14	0	3	It looks like there is significant unmodeled friction in th
23	28	7	14	7	7	14	0	5	During the steady state tracking error phase there is cle
24	29	22	32	22	22	32	22	3	Much heavier system. Took out time delays. 10 Hz 2nd
25	30	7	14	7	7	14	0	3	Changed the slave friction model to f_pos = 0.2924 and
26	31	7	14	7	7	14	0	3	Same as previous run except introduced some time dela
27	32	7	14	7	7	14	0	3	Good run with an environment and 50 Hz second order
28	33	7	14	7	7	14	0	3	Put 100 Hz second order filter on slave control and got
29	34	7	14	7	7	14	7	3	Put a spring back on the slave. Seeing more position err
30	35	7	14	7	7	14	0	3	Put a narrower boundary layer on the slave controller. W

Figure B.3: The Matlab interface used to display results of experimental runs.

combined with the Database Toolbox, this code connects to the experimental database, retrieves the parameters from each of the experiments, and displays them in a graphical table. The user may then select a particular experimental run and have the results plotted on the screen. Sample output from the results display interface software is shown in Figure B.3. This makes comparing results from various runs particularly straightforward.

References

[1] W. Hamel and P. Murray, "Observations concerning Internet-based teleoperations for hazardous environments," in *IEEE International Conference on Robotics and Automation*, vol. 1. IEEE; 1999, 2001, pp. 638–643. 1

[2] C. Preusche, T. Ortmaier, and G. Hirzinger, "Teleoperation concepts in minimal invasive surgery," *Control Engineering Practice*, vol. 10, no. 11, pp. 1245–1250, 2002. 1

[3] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 592–606, Oct. 1993. 1

[4] G. Konesky, "Deep-sea telepresence: a proposed exploration of the Hudson submarine canyon," in *Proceedings of SPIE*, vol. 4570, 2002, p. 8. 1

[5] R. J. 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. 2, 9, 32, 33, 34, 42, 95, 99

[6] G. Niemeyer and J.-J. E. Slotine, "Towards force-reflecting teleoperation over the internet," in *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, May 1998. 2, 34

[7] ——, "Telemanipulation with time delays," *The International Journal of Robotics Research*, vol. 23, pp. 873–890, Sept. 2004. 2, 34, 35

[8] ——, “Stable adaptive teleoperation,” *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 681–694, Jan. 1991. 2, 9, 34, 42, 95

[9] ——, “Designing force reflecting teleoperators with large time delays to appear as virtual tools,” in *Proceedings of the 1997 IEEE International Conference on Robotics & Automation*, Apr. 1997. 2, 9, 34, 35, 99, 103

[10] S. Ganjefar, H. Momeni, and F. Janabi-Sharifi, “Teleoperation systems design using augmented wave-variables and Smith predictor method for reducing time-delay effects,” in *Proceedings of the IEEE International Symposium on Intelligent Control*, vol. 338. Piscataway, NJ, USA: IEEE, 2002. 6

[11] D. A. Lawrence, “Stability and transparency in bilateral teleoperation,” *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, Oct. 1993. 7, 9, 10, 42, 45

[12] H.-K. Lee and M. J. Chung, “Adaptive controller of a master-slave system for transparent teleoperation,” *Journal of Robotic Systems*, vol. 15, no. 8, pp. 465–475, 1998. 7, 51

[13] M. Tavakoli, A. Aziminejad, R. V. Patel, and M. Moallem, “Enhanced transparency in haptics-based master-slave systems,” in *Proceedings of the 2007 American Control Conference*, July 2007. 8, 9, 10, 100, 101

[14] L. Ni and D. W. L. Wang, “A gain-switching control scheme for position-error-based bilateral teleoperation: Contact stability analysis and controller design,” *International Journal of Robotics Research*, vol. 23, no. 3, pp. 255–274, Mar. 2004. 9, 11, 44

[15] K. Hashtrudi-Zaad and S. Salcudean, “Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators,” *The International Journal of Robotics Research*, vol. 20, no. 6, p. 419, 2001. 10

[16] H. K. Khalil, *Nonlinear Systems*, 3rd ed. New Jersey: Prentice Hall Inc., 2002. 11, 14, 17, 18, 21, 23, 49, 50, 79, 82

[17] A. Filippov and F. Arscott, *Differential equations with discontinuous righthand sides.* Springer, 1988. 16

[18] V. C. Aitken, "Sliding mode state estimation for nonlinear discrete time systems: Applications in image sequence analysis," PhD Dissertation, Carleton University, Ottawa, Ontario, Apr. 1995. 16

[19] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electro-mechanical systems*, 2nd ed. CRC Press, 2009. 16, 17, 18

[20] S. Drakunov and V. Utkin, "Sliding mode observers. tutorial." in *Proceedings of the 34th Conference on Decision & Control*, Dec. 1995, pp. 3376–3378. 20, 66

[21] I. Haskara, U. Ozguner, and V. Utkin, "On sliding mode observers via equivalent control approach," *International Journal of Control*, vol. 71, no. 6, pp. 1051–1067, 1998. 20, 22, 151, 154

[22] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. New Jersey: Prentice Hall Inc., 1993. 21, 33

[23] T. Floquet and J. P. Barbot, "A sliding mode approach of unknown input observers for linear systems," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Dec. 2004. 21

[24] J. P. Barbot, T. Boukhobza, and M. Djemai, "Sliding mode observer for triangular input form," in *Proceedings of the 35th Conference on Decision & Control*, Dec. 1996, pp. 1489–1490. 22, 150

[25] J. M. Daly and D. W. L. Wang, "Output feedback sliding mode control in the presence of unknown disturbances," *Systems & Control Letters*, vol. 58, no. 3, pp. 188–193, 2009. 22, 142

[26] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International Journal of Control*, vol. 58, no. 6, pp. 1247–1263, 1993. 24, 25, 26

[27] ——, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, 1998. 24, 26

[28] ——, "Higher-order sliding modes, differentiation and output-feedback control," *International Journal of Control, 76*, vol. 9, no. 10, pp. 924–941, 2003. 24

[29] ——, "Universal SISO sliding-mode controllers with finite-time convergence," *IEEE transactions on Automatic Control*, vol. 46, no. 9, pp. 1447–1451, 2001. 24

[30] I. Boiko and L. Fridman, "Analysis of chattering in continuous sliding-mode controllers," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1442–1446, 2005. 25

[31] J. Davila, L. Fridman, and A. Levant, "Second-order sliding-mode observer for mechanical systems," *IEEE transactions on automatic control*, vol. 50, no. 11, pp. 1785–1789, 2005. 26, 27, 29, 31, 60, 160

[32] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-6, pp. 24–32, Sept. 1965. 32

[33] ——, "Delayed force feedback," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-8, pp. 449–455, Oct. 1966. 32

[34] G. Niemeyer and J.-J. E. Slotine, "Using wave variables for system analysis and robot control," in *Proceedings of the 1997 IEEE International Conference on Robotics & Automation*, Apr. 1997. 35

[35] N. Chopra, M. W. Spong, R. Ortega, and N. E. Barabanov, "On position tracking in bilateral teleoperation," in *Proceedings of the 2004 American Control Conference*, July 2004. 35

[36] ——, "On tracking performance in bilateral teleoperation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 861–866, Aug. 2006. 35, 36

[37] N. Chopra, P. Berestesky, and M. Spong, “Bilateral teleoperation over unreliable communication networks,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 304–313, 2008. 36, 37

[38] Y. Gu, C. Zhang, and K. Chong, “Adaptive passive control with varying time delay,” *Simulation Modelling Practice and Theory*, 2009. 37

[39] A. Aziminejad, M. Tavakoli, R. Patel, and M. Moallem, “Stability and performance in delayed bilateral teleoperation: Theory and experiments,” *Control Engineering Practice*, vol. 16, no. 11, pp. 1329–1343, 2008. 37

[40] A. Haddadi and K. Hashtrudi-Zaad, “A new robust stability analysis and design tool for bilateral teleoperation control systems,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 663–670. 38

[41] G. Niemeyer and J.-J. E. Slotine, “Towards force-reflecting teleoperation over the internet,” in *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, May 1998. 38

[42] Y. Yokokohji, T. Imaida, and T. Yoshikawa, “Bilateral control with energy balance monitoring under time-varying communication delay,” in *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, Apr. 2000. 39

[43] T. Mirfakhrai and S. Payandeh, “A delay prediction approach for teleoperation over the internet,” in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, May 2002. 39

[44] ——, “On using delay predictors in controlling force-reflecting teleoperation over the internet,” *Robotica*, vol. 23, pp. 809–813, 2005. 39

[45] T. Zhang and Y. Li, “A control scheme for bilateral teleoperation systems based on time-varying communication delay identification,” in *Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics*, Jan. 2006. 39

[46] X. Liu, W. J. Wilson, and X. Fan, “Pose reflecting teleoperation using wave variables with wave prediction,” in *Proceedings of the 2005 IEEE International Conference on Mechatronics & Automation*, July 2005. 39, 146

[47] S. Ganjefar, H. Momeni, and F. Janabi-Sharifi, “Teleoperation systems design using augmented wave-variables and smith predictor method for reducing time-delay effects,” in *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Oct. 2002. 40, 146

[48] S. Munir and W. J. Book, “Internet-based teleoperation using wave variables with prediction,” *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 124–133, June 2002. 40, 41, 146

[49] H. Chin and W. J. Book, “Internet-based bilateral teleoperation based on wave variable with adaptive predictor and direct drift control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, pp. 86–93, Mar. 2006. 41

[50] E. Kamrani, H. R. Momeni, and A. R. Sharafat, “A novel adaptive control system for stable teleoperation via internet,” in *Proceedings of the 2005 IEEE Conference on Control Applications*, Aug. 2005. 41, 42, 146

[51] D. Lee and M. W. Spong, “Passive bilateral teleoperation with constant time delay,” *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 269–281, Apr. 2006. 42, 43

[52] M. McIntyre, W. Dixon, D. Dawson, and E. Tatlicioglu, “Passive coordination of nonlinear bilateral teleoperated manipulators,” *Robotica*, vol. 24, pp. 463–476, 2006. 43

[53] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, “Stable teleoperation with time-domain passivity control,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 365–373, Apr. 2004. 44

[54] E. Naerum and B. Hannaford, “Global Transparency Analysis of the Lawrence Teleoperator Architecture,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA’09*, 2009, pp. 4344–4349. 44, 45

[55] C. Seo, J. Kim, J. Kim, J. Yoon, and J. Ryu, "Stable bilateral teleoperation using the energy-bounding algorithm: Basic idea and feasibility tests," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2008. AIM 2008*, 2008, pp. 335–340. 45

[56] C. A. Lawn and B. Hannaford, "Performance testing of passive communication and control in teleoperation with time delay," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1993. 46

[57] J. Kim, H. Kim, B. K. Tay, M. Muniyandi, M. A. Srinivasan, J. Jordan, J. Mortenses, M. Oliveira, and M. Slater, "Transatlantic touch: A study of haptic collaboration over long distance," *Presence*, vol. 13, no. 3, pp. 328–337, June 2004. 46

[58] I. G. Polushin, P. X. Liu, and C.-H. Lung, "A control scheme for stable force-reflecting teleoperation over ip networks," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 36, no. 4, pp. 930–939, Aug. 2006. 46, 47, 85

[59] ——, "A force reflection algorithm for improved transparency in bilateral teleoperation with communication delay," in *Proceedings of the 2006 IEEE International Conference on Robotics & Automation*, May 2006. 46, 85

[60] ——, "Position-error based schemes for bilateral teleoperation with time delay: Theory and experiments," in *Proceedings of the 2006 IEEE International Conference on Mechatronics & Automation*, June 2006. 46, 47

[61] K. Brady and M. Farrokhi, "Bilateral control to achieve transparent teleoperation with perturbation of static time delay," in *Proceedings of the 32nd Annual IEEE Conference on Industrial Electronics*, Nov. 2006. 47, 146

[62] V. Pilieci, "Feeling the pulse of virtual reality," *CanWest News Service*, Oct. 2003. 48

[63] E. Nuno, R. Ortega, N. Barabanov, and L. Basañez, "A new proportional controller for nonlinear bilateral teleoperators," in *Proc. of the IFAC World Congress, Seoul, Korea*, 2008. 48

[64] E. Nuno and L. Basanez, “Nonlinear Bilateral Teleoperation: Stability Analysis,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA'09*, 2009, pp. 3718–3723. 48, 49

[65] N. Chopra, M. Spong, R. Ortega, and N. Barbanov, “On tracking performance in bilateral teleoperation,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 844–847, 2009. 49

[66] J. H. Park and H. C. Cho, “Sliding-mode controller for bilateral teleoperation with varying time delay,” in *Proceedings of the 1999 IEEE/ASME Conference on Advanced Intelligent Mechatronics*, Sept. 1999. 49, 50, 104

[67] H. Cho and J. Park, “Stable bilateral teleoperation under a time delay using a robust impedance control,” *Mechatronics*, vol. 15, no. 5, pp. 611–625, 2005. 50, 141, 142

[68] L. G. Garcia-Valdovinos, V. Parra-Vega, and M. A. Arteaga, “Observer-based higher-order sliding mode impedance control of bilateral teleoperation under constant unknown time delay,” in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006. 50

[69] S. Khan, A. Nergiz, and A. Şabanoviç, “Scaled Bilateral Teleoperation Using Discrete-Time Sliding Mode Controller,” *IEEE Transactions on Industrial Electronics*, 2008. 50

[70] K. Hashtrudi-Zaad and S. E. Salcudean, “Adaptive transparent impedance reflecting teleoperation,” in *Proceedings of the 1996 IEEE International Conference on Robotics & Automation*, Apr. 1996. 51, 52

[71] K. B. Fite, M. Goldfarb, and A. Rubio, “Transparent telemanipulation in the presence of time delay,” in *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2003. 52, 78, 146

[72] I. G. Polushin, A. Tayebi, and H. J. Marquez, “Adaptive schemes for stable teleoperation with communication delay based on ios small gain theorem,” in *Proceedings of the 2005 American Control Conference*, June 2005. 52

[73] A. Shahdi and S. Sirouspour, "Adaptive/robust control for time-delay teleoperation," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 196–205, 2009. 52

[74] G. Hirzinger, J. Heindl, and K. Landzettel, "Predictive and knowledge-based telerobotic control concepts," in *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, May 1989. 53

[75] W. S. Kim and A. K. Bejczy, "Graphics displays for operator aid in telemanipulation," in *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 1991. 53

[76] K. Brady and T.-J. Tarn, "Internet-based remote teleoperation," in *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, May 1998. 53

[77] P. A. Prokopiou, S. G. Tzafestas, and W. S. Harwin, "Towards variable-time-delays-robust telemanipulation through master state prediction," in *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Sept. 1999. 54

[78] A. C. Smith and K. Hashtrudi-Zaad, "Neural network-based teleoperation using smith predictors," in *Proceedings of the 2005 IEEE International Conference on Mechatronics & Automation*, July 2005. 54, 78

[79] ——, "Smith predictor type control architectures for time delayed teleoperation," *International Journal of Robotics Research*, vol. 25, no. 8, pp. 797–818, Aug. 2006. 54

[80] J. Sheng and M. W. Spong, "Model predictive control for bilateral teleoperation systems with time delays," in *Proceedings of the 2004 Canadian Conference on Electrical and Computer Engineering*, May 2004. 55

[81] S. Sirouspour and A. Shahdi, "Discrete-time linear quadratic control for teleoperation under communication time delay," *International Journal of Robotics Research*, vol. 25, no. 2, pp. 187–202, Feb. 2006. 55, 56, 146

[82] ——, “Model predictive control for transparent teleoperation under communication time delay,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1131–1145, Dec. 2006. 55, 56

[83] L. Ni and D. W. L. Wang, “A human-to-human force-reflecting teleoperation system using fuzzy logic controller tuning,” *Journal of Intelligent and Robotic Systems*, vol. 48, no. 2, pp. 209–224, Feb. 2007. 56

[84] S. Munir, “Internet-based teleoperation,” PhD Dissertation, Georgia Institute of Technology, Atlanta, GA, Mar. 2001. 56, 57, 146

[85] Y.-J. Pan, C. C. de Wit, and O. Sename, “A new predictive approach for bilateral teleoperation with applications to drive-by-wire systems,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1146–1162, Dec. 2006. 57

[86] K. Yoshida, T. Namerikawa, and O. Sawodny, “A State Predictor for Bilateral Teleoperation with Communication Time Delay,” in *47th IEEE Conference on Decision and Control, 2008. CDC 2008*, 2008, pp. 4590–4595. 57, 58

[87] J. M. Daly and D. W. L. Wang, “Bilateral teleoperation using unknown input observers for force estimation,” in *Proceedings of the 2009 American Control Conference*, June 2009. 60, 140

[88] J. Rosen, B. Hannaford, M. MacFarlane, and M. Sinanan, “Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-experimental performance evaluation,” *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 10, pp. 1212–1221, 1999. 60

[89] H. C. Cho, J. H. Park, K. Kim, and J.-O. Park, “Sliding-mode-based impedance controller for bilateral teleoperation under varying time delay,” in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, May 2001. 61, 95, 100, 101, 103, 113

[90] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control.* United States of America: John Wiley & Sons, Inc., 1989. 63, 99

[91] T. Yamamoto, M. Bernhardt, A. Peer, M. Buss, and A. Okamura, “Techniques for environment parameter estimation during telemanipulation,” in *2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008. BioRob 2008*, 2008, pp. 217–223. 78

[92] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems.* New York: Birkhauser Boston, 2003. 90, 91, 143

[93] K. Hashtrudi-Zaad and S. Salcudean, “Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 108–114, 2002. 102, 103

[94] D. R. Madill, “Modelling and control of a haptic interface: A mechatronics approach,” PhD Dissertation, University of Waterloo, Waterloo, Ontario, 1998. 124

[95] L. Ni and D. Wang, “Contact transition stability analysis for a bilateral teleoperation system,” in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA’02*, vol. 3, 2002. 145