Stabilization Of Delayed Teleoperation Systems Using Time Domain Passivity Control

Vom Fachbereich Elektrotechnik und Informatik der
Universität Siegen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

genehmigte Dissertation

von

M.Sc. Asif Iqbal

1. Gutachter: Prof. Dr.-Ing. H. Roth
2. Gutachter: Prof. Dr.-Ing. habil. O. Loffeld
Vorsitzender: Prof. Dr.-Ing. R. Mayr

Tag der mündlichen Prüfung: 22.06.2007

gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

To my dear parents

Acknowledgments

I would like to express my profound gratitude and appreciation to my advisor Prof. Dr. Hubert Roth for his consistent help, guidance and attention that he devoted throughout the course of this work. Despite his very busy schedule, he always could find time to advise me on scientific matters and extended his valuable support even while away from his office.

I am indebted to Prof. Dr. Otmar Loffeld for all that I had the privilege to learn from him, the useful discussions that made my research period enjoyable, and his frequent encouragements that contributed to the accomplishment of this work.

I also acknowledge and am thankful for the research scholarship for doctoral studies in Germany provided by DAAD(German Academic Exchange Service). I am grateful for the support and facilities provided by ZESS(Center for Sensor Systems), University of Siegen. I am especially thankful to Dr. Stefan Knedlik and Mrs. Silvia Niet-Wunram who were always there to provide any help that I needed during my stay at ZESS.

Thanks are due to my friends and colleagues for their moral support, good wishes and the memorable days shared together.

I am deeply indebted to Mr. Rana Amanullah for the benevolent role that he played throughout my academic years.

And finally, special thanks are due to my family members especially my parents and my wife Samina for their patience, understanding and invaluable prayers.

Contents

List of Figures

List of Tables

DISSERTATION ABSTRACT

Name: ASIF IQBAL

Title: STABILIZATION OF DELAYED TELEOPERATION SYSTEMS USING TIME-DOMAIN PASSIVITY CONTROL

Degree: DOKTOR-INGENIEUR (Dr.-Ing.)

Major Field: ELECTRICAL ENGINEERING

It is known that time delay in bilateral teleoperation can drive a system to instability. Time Domain Passivity Control (TDPC) deals with the stabilization of haptic interfaces in teleoperation using the notion of passivity directly in the continuous time variables like force and velocity. In this work, first it is shown that TDPC can be extended to stabilize the time-delayed teleoperation by considering the communication channel as an active component and then, to design passivity controllers for it on master side using a Kalman filter based recursive prediction of slave side energy. However, such a scheme is prone to large corrective impulses generated by passivity controllers as the scheme only comes into effect when the net energy goes negative, while on other time instants it stays out of the control loop in order to provide maximally transparent teleoperation. These impulses degrade the performance of teleoperator. It is thus further proposed, that the derivative of net energy should also be computed in real-time, and as soon as this term becomes negative, indicating a decline in the net energy, the passivity controllers should immediately compensate this active behavior. This forces the system to always dissipate energy and thus stop the occasional accumulation of a large amount of negative energy. In addition to that, parabolic power integration is employed to provide non-linear estimation of net energy in the communication channel.

The above developed approach is then used to stabilize time-varying delays. In order to provide a time-base for the predictor, a first order one-step ahead prediction of RTT (Round Trip Time) is used. Beta distributed and TrueTime network simulator based delays are used to evaluate the system performance. Simulation results are given showing the efficacy of the proposed approach.

Keywords: *Time Domain Passivity Control, Stabilization, Teleoperation, Time Delay, Telerobotics*

Kurzfassung

Name: ASIF IQBAL

Titel: STABILIZATION OF DELAYED TELEOPERATION SYSTEMS USING TIME-DOMAIN PASSIVITY CONTROL

Abschluss: DOKTOR-INGENIEUR (Dr.-Ing.)

Schwerpunkt: ELEKTROTECHNIK

Es ist bekannt, dass eine Zeitverzögerung in bilateraler Teleoperation ein System bis zur Instabilität führen kann. Time Domain Passivity Control (TDPC) behandelt die Stabilisierung von haptischen Ein- und Ausgabegeräten mittels der Idee von Passivität (Passivity), bezogen auf die stetigen Zeitvariablen wie Kraft und Geschwindigkeit. In dieser Arbeit wurde zuerst demonstriert, dass die Erweiterung des Konzeptes von TDPC für die Stabilisierung der zeitverzögerten Teleoperation eingesetzt werden kann, falls der Kommunikations-Kanal als ein aktives Bauteil angesehen wird. Anschließend wird ein auf einem rekursiven Kalman-Filter basierender Passivitäts-Controller entwickelt, der die Gesamtenergie des Slavesystems vorhersagt. Durch diese Vorgehensweise werden jedoch große Impulse durch den Passivitäts-Controller erzeugt, weil dieser ausschließlich dann aktiv wird, falls die Energie im Netz negativ wird. Ist die Energie im Netz positiv, ist der Regelkreis nicht aktiv, damit die Übertragung der Signale über das Kommunikationsnetz nicht beeinflusst wird. Diese Impulse können die Funktion eines Teleoperators stark beeinträchtigen. Es wird deshalb vorgeschlagen, die Ableitung der Energie im Netz in Echtzeit zu berechnen, damit, sobald die Ableitung negativ wird, was einen Abfall der Energie andeutet, der Passivitäts-Controller aktiv eingreifen kann, um das Netz wieder zu stabilisieren. Dies erzwingt eine ständige Verminderung der Energie im System, und somit ist das System gegen die zeitweise Akkumulation von großen negativen Energiemengen abgesichert. Darüber hinaus wird eine parabolische Integration eingesetzt, um die nicht-lineare Schätzung der Gesamtenergie im Kommunikations-Kanal zu gewährleisten.

Der oben entwickelte Ansatz wird dazu benutzt, um stochastische Totzeiten zu stabilisieren. Um eine Zeitbasis für die Schätzung der Netzenergie zu erhalten, wird die Rundreisezeit (RTT: Round Trip Time) mit Hilfe eines Prädiktors erster Ordnung geschätzt. Basierend auf der Betaverteilung und einem Programm namens ,,TrueTime network simulator" werden Totzeiten benutzt, um die Systemstabilität zu überprüfen. Simulationsergebnisse zeigen die Wirksamkeit des vorgeschlagenen Verfahrens.

Schlagwörter: *Time Domain Passivity Control, Stabilisierung, Teleoperation, Totzeit, Telerobotic*

Chapter 1

Introduction

1.1 Introduction

Teleoperation can be defined as the extension of a person's sensing and manipulative capabilities to a location, remote from him[1]. It involves the control of a robot or some other machine from distance by, in most cases, a human operator. There are certain applications in which using such an approach is highly advantageous. In situations where human life is endangered like deep sea surveillance, bomb disposal, mine-field clearance, and nuclear waste handling, etc., teleoperation becomes the primary choice. On the other hand, teleoperation provides solutions also in cases where human operators simply can't manipulate given objects, for example surgery inside human body through micro-robots or deep-space missions. Tele-surgery is another example of teleoperation systems.

Teleoperation is finding applications in these areas because the technology can save lives and reduce costs by removing the human operators from the operation sites. However in most of these areas, we still need humans in the control loop because of their very high level of skills and because machine intelligence is insufficiently advanced to operate autonomously and intelligently in such complex, unstructured and often cluttered environments.

Teleoperation has become one of the most rapidly expanding areas in mechanical, electrical, computer and control systems engineering. Today many industries utilize robots because they offer advantage of being able to perform set routines more quickly, cheaply, and accurately than humans. Instead of using programmed routines to maneuver the robots, telerobotics allows to operate the robot from a distance and make decisions while telemanipulating the robot in real time. With the development of more powerful and efficient computers, the future for teleoperation seems extremely promising. However, it is the flexibility with which these teleoperated robots can be used that is of great concern to both users and researchers of teleoperation.

An active research area in teleoperation is to compensate for time delays in operator-telerobot interface. Continuous manual control of the remote manipulator is impeded if there is a time-delay between the control input by the operator and the consequent feedback of its control actions visible on the display. A con-

tinuous closed-loop control becomes unstable at a particular frequency when the time-delay in the control loop exceeds half the time period at that frequency. It has been shown in the literature that when the human operator is in the control loop, this instability can be avoided by using a "move and wait strategy", wherein the operator makes small incremental moves in an open loop fashion and then waits for a new update of the position of the telerobot.

A time-delay in communication between local and remote site can occur due to a large distance between them, due to a low speed of data transmission, or due to computer processing at different stages, or all of the above. For teleoperation in earth orbit from the ground for example, the radio transmission takes 0.4 seconds, but in reality a round trip time delay of up to 6 seconds is common, owing to multiple reflections of signals through the satellites [2]. For teleoperation underwater, sonar signals which have a speed of about 1700m/s in water, are used for data transmission, when the remote manipulator is not directly connected with cables to the controlling site. A round trip time-delay of 10 seconds is common for teleoperation in deep sea. Apart from the speed of communication and distance, considerable time can be taken by signal processing and data storage in buffers at various stages between the local site and the remote site. Also digital communication channels such as internet, cause additional problems due to added uncertainty in the actual magnitude of the time-delay. If a dedicated medium of communication is used between the master and the slave side, many of the problems like time delays and data losses that teleoperation faces today, no longer exist significantly. But a dedicated communication channel is not usually feasible because of economical concerns. Also in some situations where we have such a communication channel like in satellite deep space operations, the time-delay is inevitable because of the presence of the large galactic distances.

In a teleoperation framework, the feedback from the remote site is of prime importance for the proper functioning of the teleoperator. This feedback can be of different kinds like visual, force, acoustic, etc. It was stated as early as 1966[3] that especially the force feedback greatly enhances the teleoperation because it is useful:

- to govern applied forces either to avoid excessive stresses or to produce a desired movement of an object, or even to minimize energy expenditure.
- to estimate mass, frictional resistance, and other physical properties.
- to provide information about contact with objects or with certain features of the environment while visual information is not available.

Force feedback is helpful but there are several issues attached to its provision in teleoperation. First of all, it turns the unilateral control scheme to a bilateral one. Thus both ends are serving as both input as well as output and/or feedback. Addition of a time delay in such a configuration seriously raises the question of the stability. Ferrel states[3]:"with a delay, the feedback is not only a source of information, but may act as a disturbance input as well. Just as one would predict instability for any closed loop system having a long delay and a loop gain greater

than unity, it would be expected that, if the delay and the rate at which feedback force changes with the position of the remote hand are great enough, a manipulator can become uncontrollable." The problem of delay is essential to address because in certain situations, its presence is unavoidable like deep space, or in acoustic signals used in sonar, and internet. In order to render a system usable in these cases, it is required to provide sufficient evidence that it is stable because otherwise it could result in unwanted consequences on the remote site, or could even cause damage to the human operator locally. Stabilization of such systems involving communication channels, where the delay is random, requires rigorous analysis of the problem and is actively being pursued in the research circles. For related work, see [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1] and the references therein.

In this work, a solution to the above described problem using Time Domain Passivity Control (defined in chapter 5) is presented for constant as well as time-varying delays.

1.1.1 Definitions

In this section, some definitions regarding the work being presented are given.

Robotics is the science and art of performing, by means of an automatic apparatus or device, functions ordinarily ascribed to human beings, or operating with what appears to be almost human intelligence.

Telepresence is the ideal of sensing sufficient information about the teleoperator and task, and communicating this to the human operator in a sufficiently natural way that she feels herself to be physically present at the remote site. Telepresence attempts to completely hide all the teleoperator dynamics and is not achievable for delayed systems, in ideal sense[18].

Teleoperation is the extension of a person's sensing and manipulative capabilities to a location, remote from him/her[1].

Telerobotics is a form of teleoperation in which a human operator acts as a supervisor, communicating to a computer, some information about task goals, constraints, plans, contingencies, assumptions, suggestions and orders, getting back information about accomplishments, difficulties, concerns, and, as requested, raw sensory data–while the subordinate teleoperator executes the task based on information received from the human operator plus its own artificial sensing and intelligence.

Bilateral Teleoperation is realized when the operator commands a velocity forward, through the master, communication block, and slave, to the environment, and contact force information is reflected back to the operator. Ideally, the slave is controlled so that in the steady state, the slave velocity v_s is equal to the operator velocity v_m and the back-driven force f_{md} is equal to the contact force f_e.

Impedance As described in [19]: "Stable positioning requires, at minimum, a static relation between force and position; some spring-like element must be included in the equivalent physical network. The controller must specify a vector quantity such as the desired position, but it must also specify a quantity which is fundamentally different: a relationship, an *impedance*, which has properties similar to those of a second-rank, twice covariant tensor; it operates on a contravariant vector of deviations from the desired position to produce a covariant vector of interface forces."

Transparency of the teleoperation system describes the extent to which the feel of the environment and the operator commands are preserved. Ideal kinesthetic coupling, and thus, a realistic haptic impression is achieved, if positions and forces at the HSI(Human System Interface) and teleoperator equal each other. Applying the generalized formulation of impedance, where position replaces velocity in $f = Zv$, transparency requires that the felt impedance be equal(or as close as possible) to the environment impedance[20].

Resolved Force Sensing is what the human body's joint, muscle and tendon receptors do to determine the net force and torque acting on the hand, i.e., the resultant vector of all the component forces and torques operating on the environment.

Haptics is the science of applying tactile sensation to human interaction with computers. By using haptic devices, the user can not only feed tactile information to the computer but can also receive information from the computer in the form of a felt sensation on some part of the body.

Haptic Interface is such an HMI(Human Machine Interface) that can generate adequate sensory stimuli to render to the human operator, the sensation of physical interaction with a virtual environment (e.g. forces generated either by the weight or by the collision with simulated objects) or with a remote environment (e.g. in teleoperation systems) through force feedback devices that can exert a controlled force on the operator to let him feel as if he were in touch with a real object [21].

Supervisory Control in the strictest sense means, that one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial effectors to the controlled process or task environment.

It must be noted here that in all of the above definitions, the human operator is normally the entity generating the actions to be carried out by the remote manipulator but this definition can also be extended to include some artificially intelligent algorithms running locally that can substitute the human operator.

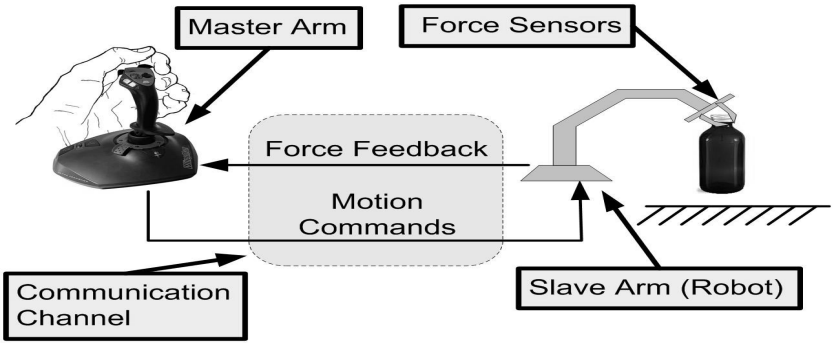

Figure 1.1: Major components of a teleoperation system

1.1.2 Components of a Teleoperation System

A teleoperator can be described as consisting of a minimum of (i) artificial sensors, (ii) actuators on local and remote sites, and (iii) a communication channel to and from the human operator or some intelligent control algorithm instead of a human being. Generally these components translate into following:

1. Master arm
2. Slave arm
3. Communication channel

The master arm serves as the HMI(Human Machine Interface) in teleoperation system. For force feedback enabled teleoperation, the master arm is a haptic device capable of displaying the reflected forces from the environment to the human operator. In some cases, a force feedback joystick is also appropriate for this purpose.

The slave arm is a robot that executes motion commands given by the human operator using the master arm. It can be a huge industrial robot or a micro surgical probe inside the human body.

The choice of communication channel depends on the given application as well as the resources. In deep space mission, the channel becomes a dedicated radio link which does have delay but they are measurable and are generally deterministic. On the other hand, in many current teleoperation applications, the ubiquitous internet serves as the communication link between master and slave robots. Use of internet, however, introduces several challenges to real-time robotic control applications. Apart from typical challenges inherent in teleoperation, the use of internet introduces time delays as well as packet loss into the control system. Several researchers have explored the control-theoretic issues of developing real-time dynamic controllers for communication channels experiencing time delay. In addition, the delays in a channel like internet are no more deterministic and thus greatly increase the complexity of the control and stabilization issues. These components can be easily identified in Fig. 1.1.

1.1.3 Abbreviations

The abbreviations used in this work are given below:

RHP	:	Right Half Plane
LHP	:	Left Half Plane
$BIBO$	:	Bounded Input Bounded Output
LMI	:	Linear Matrix Inequality
$MJLS$	:	Markov Jump Linear Systems
LTI	:	Linear Time Invariant
HSI	:	Human System Interface
$d.o.f.$	:	Degrees Of Freedom
$TDPC$	:	Time Domain Passivity Control
PPI	:	Parabolic Power Integration
RTT	:	Round Trip Time
FTT	:	Forward Trip Time
BTT	:	Backward Trip Time
TD	:	Time Delay
DT	:	Dead Time
NCS	:	Network Control System
GAS	:	Global Asymptotic Stability
PO	:	Passivity Observer
PC	:	Passivity Controller

1.1.4 Notations

This section provides the scientific notations that will be used in the following work.

$\mathbb{R}^n$	:	Set of real n-dimensional vectors
$\mathbb{R}^{n\times m}$	:	Set of real $n \times m$ matrices
$\mathcal{C}[-\tau, 0]$	:	Space of continuous functions defined on $[-\tau, 0]$
$\|.\|_2$	:	2-norm (vector, signal, or system)
$\|.\|_\infty$	:	∞-norm (vector, signal, or system)
$\mathbb{E}\{.\}$	:	Mathematical expectation
A	:	State matrix, $A \in \mathbb{R}^{n\times n}$
A_d	:	State matrix of the delayed state vector, $A_d \in \mathbb{R}^{n\times n}$
B	:	Input matrix, $B \in \mathbb{R}^{n\times m}$
v_h	:	Human velocity
v_{md}	:	Desired master velocity
v_m	:	Actual master velocity
v_{sd}	:	Desired slave velocity

v_s	:	Actual slave velocity
f_e	:	Environment force
f_s	:	Force felt by slave force sensor
f_{md}	:	Desired output force on master arm
f_m	:	Actual output force on master arm
b_m	:	Damping factor of master arm
m	:	Mass of master(and slave) arm beam
r	:	Length of master(and slave) arm beam
z_m	:	Impedance of master arm
z_e	:	Impedance of environment
v_{me}	:	Equivalent velocity to f_m
Z_m	:	Master impedance felt by human
T_f	:	Time delay in forward network
T_b	:	Backward time delay
T_s	:	Sample time
τ	:	Motor torque
α_m	:	Master side passivity controller
α_s	:	Slave side passivity controller

1.2 Organization of the Work

Chapter 2 describes the research problem. A brief literature survey pertinent to the dissertation objectives is given in chapter 3. In chapter 4, a fundamental coverage of stability and passivity concepts is provided. Time Domain Passivity Control, the fundamental approach used in this work, is covered in chapter 5. Main contributions of the dissertation are given in chapters 6 and 7. Discussion of results and conclusions are given in chapter 8.

Chapter 2

Problem Definition

It was stated as early as 1966 [3] that force feedback from remote site greatly enhances the teleoperation performance. However there are several issues raised by force-feedback in teleoperation and one of the most important among them is the question of stability.

The problem of stability in the bilateral teleoperation with time delay was first reported by Ferrell[3]. Since its identification, different solutions have been proposed to solve it. Anderson and Spong[8] attempted the stabilization of bilateral control of teleoperators with time delay by passivation of the system using scattering theory. In almost all teleoperation literature, the system is described as a network consisting of n-ports where master, communication block, and slave are represented by 2-ports and the operator and environment by 1-ports as shown in Fig. 2.1.

Niemeyer and Slotine[15] used wave variables based approach for the passivation of telerobotic systems and validated the results of [8].

It is reported that passivity features closure properties which implies that a combination of two passive systems connected in either feedback or in parallel configuration is again passive. A lot of work in recent years has focused on using wave variables for stabilization in the presence of constant or time-varying delays, see for example [22, 13, 23] and the references therein.

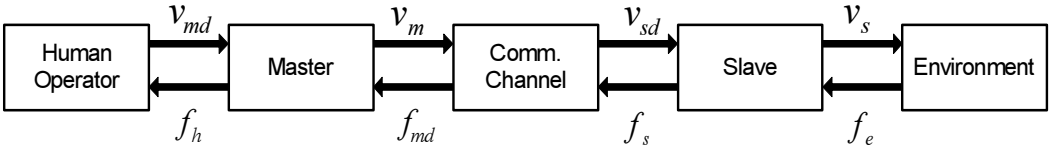

Figure 2.1: Port-based teleoperation

2.1 Instability due to Time Delay

Instability in a control loop with time delay can be described using *Nyquist frequency*. If it is required to control a process such that the difference between the reference and feedback signals becomes almost zero, then the loop gain is mostly greater than unity for almost all physical plants in the operational frequencies[2]. In such cases, if the time delay becomes greater than half the time constant of the system, the phase margin will diminish and the negative feedback will rather become positive. It means that controller will continue to induce energy into the system and the input signal will grow unboundedly thus eventually driving the system to instability. Frequencies of interest are highly dependent on transparency. If a high degree of transparency is required then upper bound on the frequencies of interest need to be increased which would essentially result in i) either smaller values of time delays, or ii) decreased stability margins. In other words, stability is inversely proportional to the transparency of a teleoperation system.

The problem can also be stated with the help of a simple 1^{st}-order system. Consider a linear time-invariant (LTI) plant with input delay:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t-\tau) \\ y(t) &= Cx(t) \end{aligned} \tag{2.1}$$

where $x \in R^n$, $u \in R$ and $y \in R$. The transfer function of this system will become:

$$\frac{y(s)}{u(s)} = P(s) = P_0(s)e^{(-\tau s)} \tag{2.2}$$

Here $P_0(s)$ is obtained using:

$$P_0(s) = C(sI - A)^{-1}B \tag{2.3}$$

Now considering the system in Fig. 2.2 with:

$$G(s) = K \tag{2.4}$$

$$P_0(s) = \frac{1}{Ts+1} \tag{2.5}$$

Using Eq. 2.2, the closed loop transfer function for this plant with an input delay τ, can be written as:

$$\frac{y(s)}{r(s)} = \frac{Ke^{-\tau s}}{1 + Ts + Ke^{-\tau s}} \tag{2.6}$$

It is clear that not only the characteristic equation now involves a transcendental function making it a distributed system with infinite poles, moreover it can also be seen that $e^{-\tau s} = e^{-jw\tau}$ readily results in a phase shift given as:

$$\phi(\omega) = -\omega\tau \tag{2.7}$$

which shows that for any given stability margins, the operating frequency and time delay are inversely proportional or in other words as τ/T grows, the stability

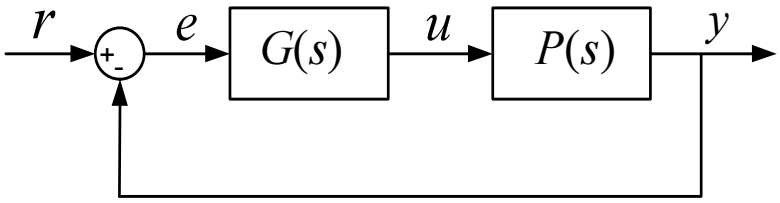

Figure 2.2: A closed loop SISO system

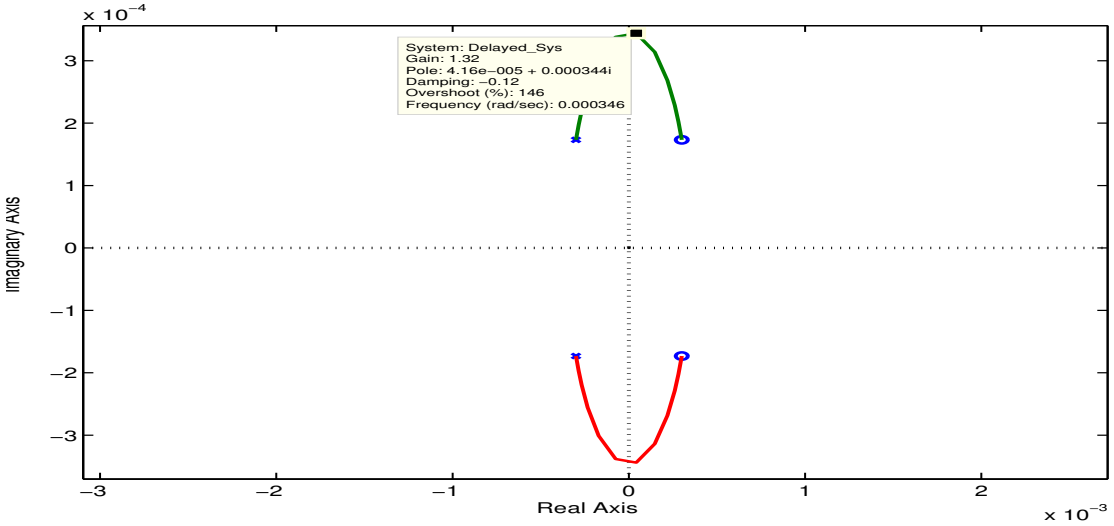

Figure 2.3: Root locus of system in (2.6) with $\tau/T = 1 \times 10^6$

decreases. For the delayed plant in Eq. 2.6, $K \to \infty$ as $\tau/T \to 0$ which also means that the system approaches to become delay free. However, as $\tau/T \to \infty$, $K \to 1$, as also predicted by small gain theorem. A root locus plot showing the second case is shown in Fig. 2.3.

If we just take the open loop system with transport delay as:

$$\frac{y(s)}{r(s)} = \frac{Ke^{-\tau s}}{Ts+1} \tag{2.8}$$

then with $\tau = 0$, the Nyquist diagram and bode plot given in Figs. 2.4 and 2.5 show that the system is stable with a very good phase margin.

However the inclusion of a delay equal to half the time constant makes the system with unity gain shift considerably toward instability as can been seen from Figs. 2.6 and 2.7.

Finally Figs. 2.8-2.9 confirm that with the same delay and a nominal gain of 5, the system is completely unstable as the Nyquist plot encircles $(-1, 0)$ point multiple times.

Preceding discussion clearly establishes the case for stability problem in tele-operation where not only the system is controlled bilaterally but also the delays are much larger in comparison to the desired command frequency.

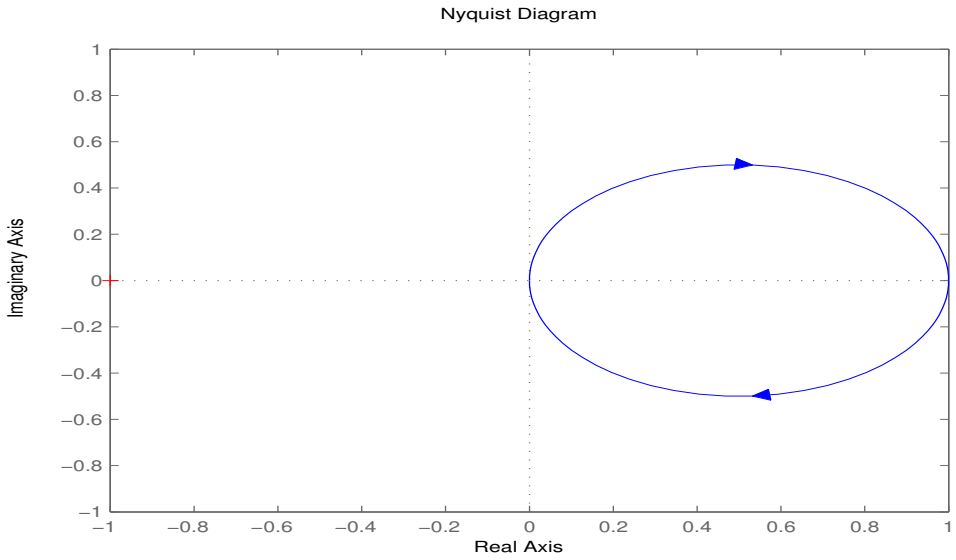

Figure 2.4: Nyquist plot of system in (2.8) without delay

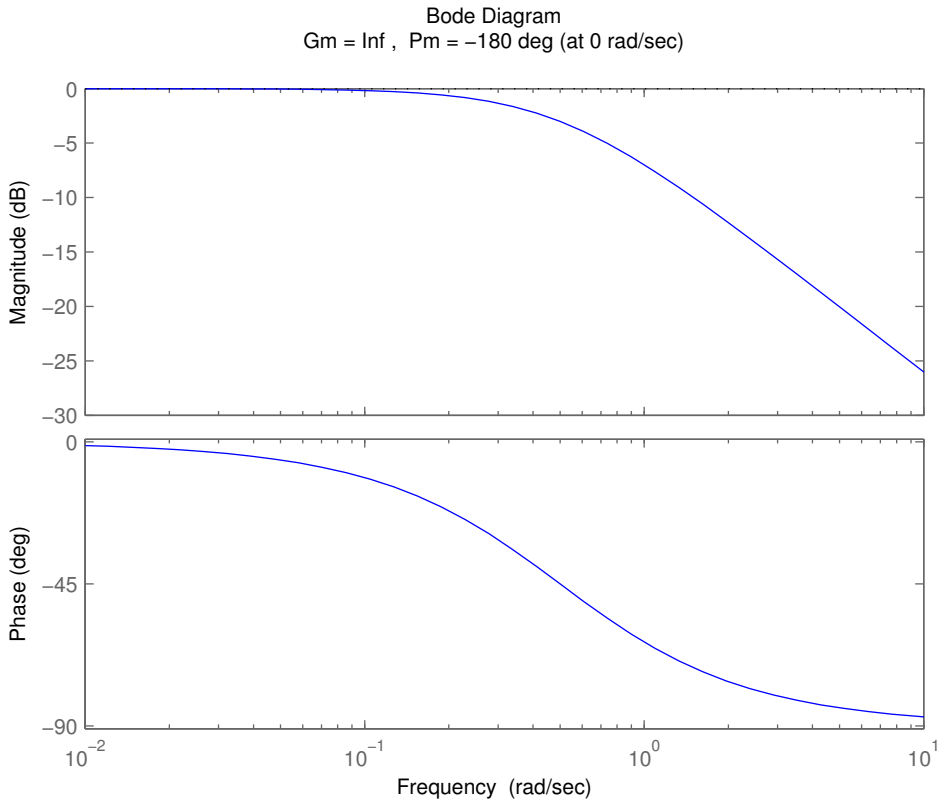

Figure 2.5: Bode plot of system in (2.8) without delay

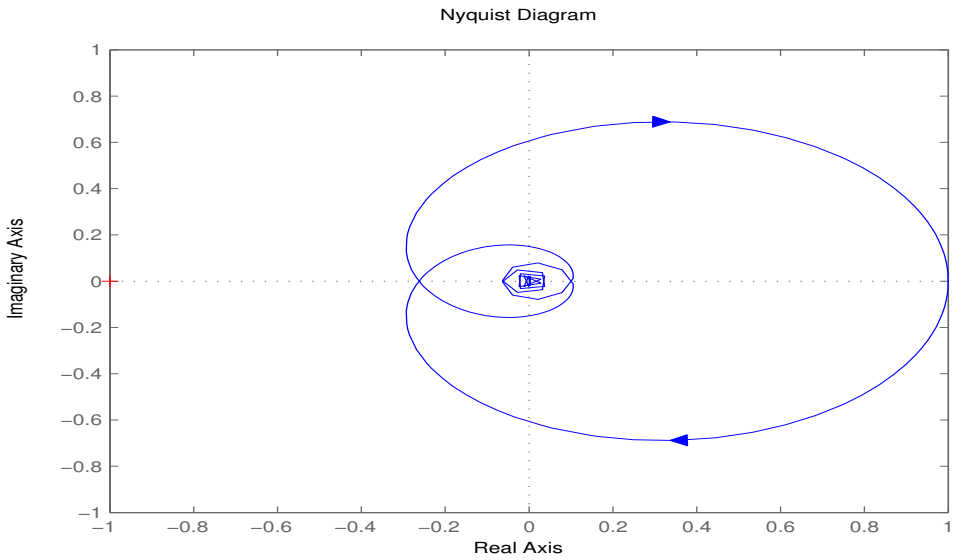

Figure 2.6: Nyquist plot of system in (2.8) with delay equal to half the time constant

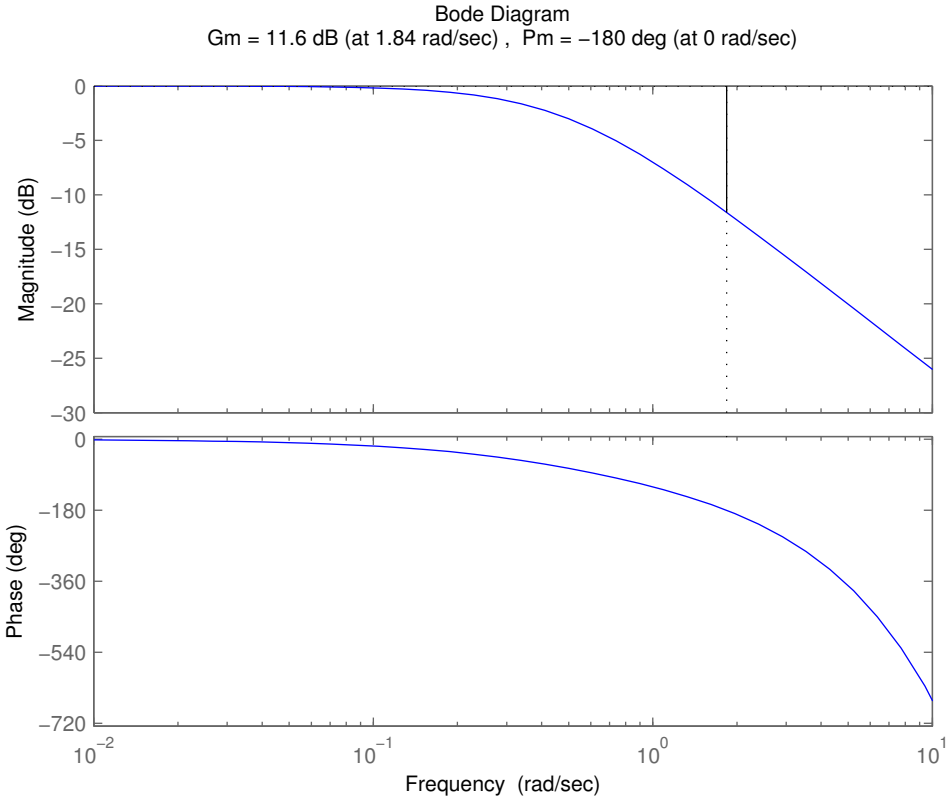

Figure 2.7: Bode plot of system in (2.8) with delay equal to half the time constant

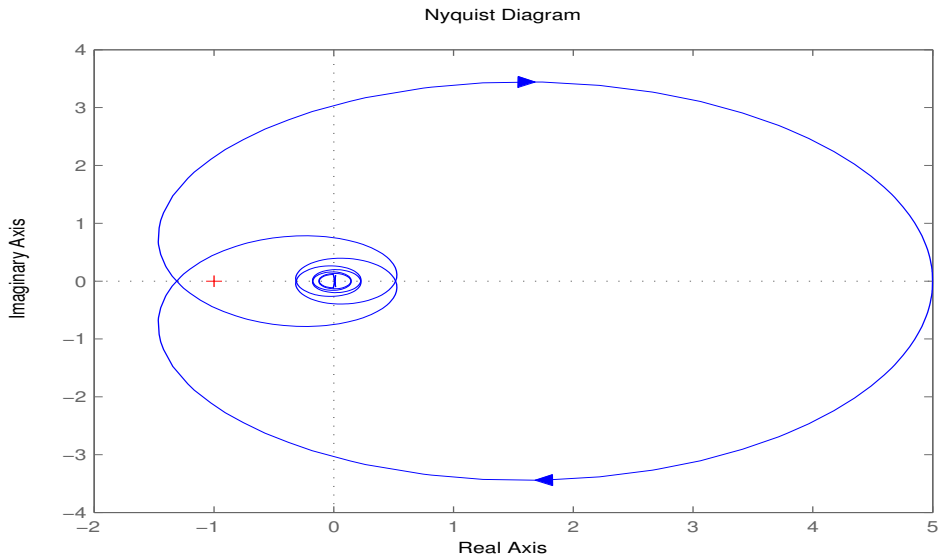

Figure 2.8: Nyquist plot of system in (2.8) with delay equal to half the time constant and gain K = 5

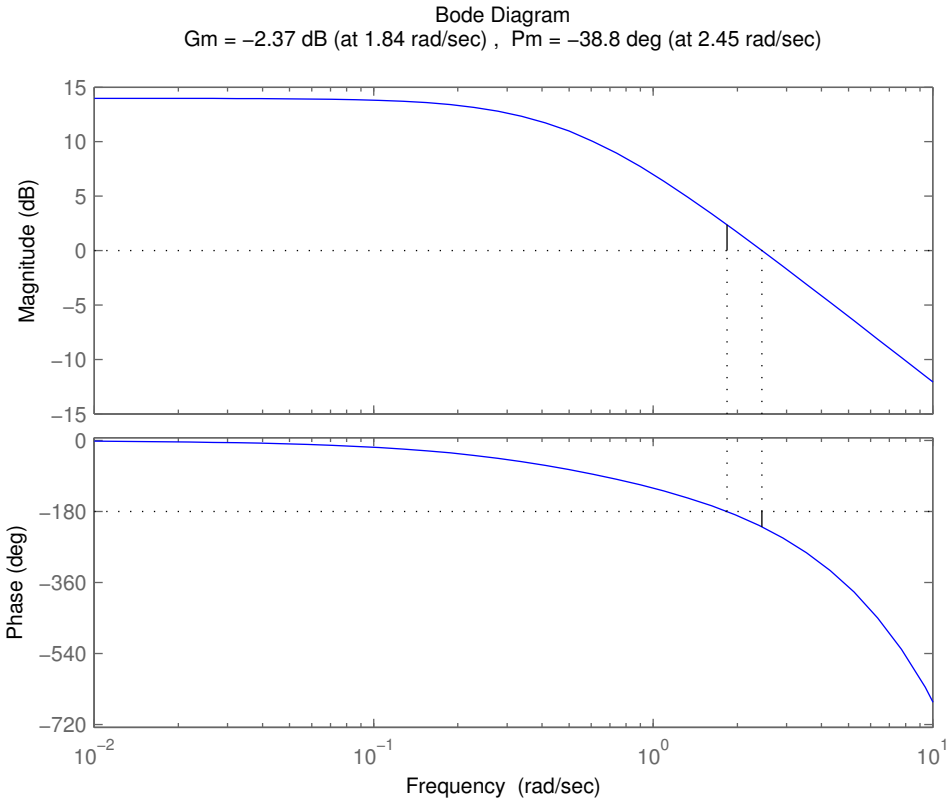

Figure 2.9: Bode plot of system in (2.8) with delay equal to half the time constant and gain K = 5

2.2 System Modeling

As the main result of this dissertation is the stabilization of time delayed teleoperation system, so in order to simulate the stabilizing behavior of the approaches presented in chapters 6 and 7, a simulation setup is developed in Matlab/Simulink utilizing also the accelerated code blocks developed in C/C++. This simulation setup consists of one-degree-of-freedom master and slave arms in addition to human operator and environment models. Communication channel will be modified according to whether a constant time delay is required or a variable time delay is needed. The master- and slave-arms are simulated using continuous-time models. The controller and transmission is executed in a sampled manner at a rate of 100HZ. A force-velocity architecture is used in the simulation model of the teleoperation system. The environment model uses a stiffness constant $K = 5N/m$ and has a damping constant of $0.5N.s/m$. The master arm uses a force control loop to output the reflected force from the environment while the slave arm uses velocity control to follow the operator commands. Velocity command is set to a sine wave with an amplitude of $0.24m/s$ and a frequency of $2Hz$. The sampling interval of the main control and feedback loop is set at $10ms$. This simulation setup will also be used in this chapter in order to highlight the instabilities in delayed teleoperation.

A generalized dynamic model of a port based teleoperation system, without time delay, can be written as:

$$M_m\ddot{x}(t)_m + G_m(x_m(t), \dot{x}_m(t)) = u_m(t) + f_h(t) \quad (2.9)$$

$$M_s\ddot{x}_s(t) + G_s(x_s(t), \dot{x}_s(t)) = u_s(t) - f_e(t) \quad (2.10)$$

where:

$x_m(t)$, $x_s(t)$	:	Master and slave generalized coordinate vectors
M_m, M_s	:	Respective inertia matrices
G_m, G_s	:	Respective force vectors
u_m, u_s	:	Control inputs to each master and slave manipulators, resp.
f_h, f_e	:	Force applied by human operator and environment, resp.

Considering Fig.2.1, in a simplified teleoperation system, $u_m(t)$ and $u_s(t)$ are computed as:

$$u_m(t) = f_{md}(t) = f_s(t) \quad (2.11)$$

$$u_s(t) = K_s x_{sd}(t) + B_s\dot{x}_{sd}(t) + M_s\ddot{x}_{sd}(t) \quad (2.12)$$

where $\dot{x}_{sd}$ is the desired slave velocity.

In the case of known or estimated time delay τ in both forward and reverse channels, Eqns. 2.11-2.12 change to:

$$u_m(t) = f_s(t-\tau) \quad (2.13)$$

$$u_s(t) = K_s x_{sd}(t) + B_s\dot{x}_{sd}(t) + M_s\ddot{x}_{sd}(t) \quad (2.14)$$

where:

$$\dot{x}_{sd}(t) = \dot{x}_m(t - \tau) \tag{2.15}$$

Now we proceed to the modeling of the simulation setup that is to be used in this work. First a general description of the physical components will be given upon which the mathematical models for simulation will be developed. After that the models for master and slave arms as well as the environment will be discussed.

2.2.1 Teleoperation Setup Description

To model both the master arm and slave, a type of motor, gear and encoder must be selected so as to make the model closer to real teleoperation setup. For slave, we select a 0.3 m long beam having a weight of 0.75 Kg which is attached to the shaft of the slave as shown in Fig. 2.12. When the slave receives a motion command from the communication channel, it moves and interacts with the environment, which will be modeled as impedance. A strain gauge measures the force exerted by the environment on the slave. This force is transmitted back to human operator.

For carrying out velocity and force commands in both master and slave arms, we select Maxon RE30, 60W DC motor, the parameters of which along with details of gears, etc. are given in Appendix B.

2.2.2 Master Arm

In teleoperation systems, the master arm accepts motion commands from the operator as well as displays the force to the same thus in addition to motion encoders, a force or torque control loop must be established in this device. The master arm is modeled as force controlled dc motor. In order to obtain the desired response, a PI controller is used for its simplicity and good performance.

The PI controller can be represented as:

$$G_c(s) = K_p + K_I/s \tag{2.16}$$

Thus the open loop transfer function becomes:

$$G_o(s) = G_c(s).G_p(s) \tag{2.17}$$

Consider Fig. 2.10, where K_t is motor torque constant, K_m is motor voltage constant, and J_m is rotor moment of inertia. We begin by the design of innermost torque(current) control loop.

The electrical time constant of the DC motor is calculated as:

$$\begin{aligned} \tau_e &= \frac{L_a}{R_a} \\ &= \frac{0.03 \times 10^{-3}}{0.207} = 0.14\ ms \end{aligned} \tag{2.18}$$

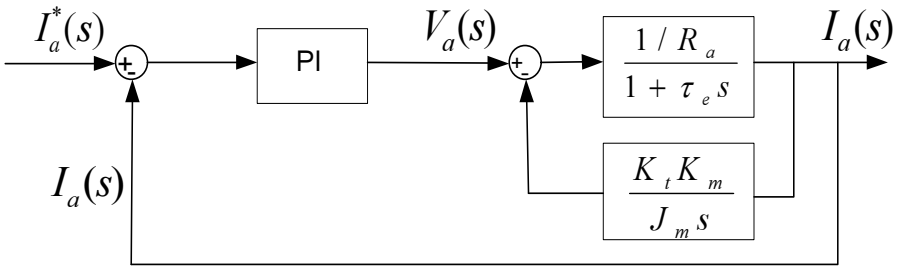

Figure 2.10: Block diagram of master arm current control loop

Using the current PI controller design method proposed in [24], the current loop is simplified by assuming that the total inertia is sufficiently large and T_w (equivalent torque caused by weight of the load) can be neglected. The open loop transfer function of the simplified current loop is then obtained as:

$$G_o(s) = \frac{K_I}{s}\left(1 + \frac{s}{K_I/K_P}\right)\frac{1/R_a}{1 + \dfrac{s}{1/\tau_e}} \tag{2.19}$$

We select the zero of PI controller to cancel the motor pole at $1/\tau_e$, to gain a phase margin of 90 degrees:

$$\frac{K_I}{K_P} = \frac{1}{\tau_e} \tag{2.20}$$

Therefore the open-loop transfer function is given as:

$$G_o(s) = \frac{K_I}{s}\left(\frac{1}{R_a}\right) \tag{2.21}$$

and the closed loop transfer function of the master arm becomes:

$$G_{cm} = \frac{1}{\left(\dfrac{R_a}{K_I}\right)s + 1} \tag{2.22}$$

As the torque in DC motor is given as:

$$T = K_t I_a \tag{2.23}$$

so, the same represented in mechanical terms is:

$$T = fr \tag{2.24}$$

where f is the force applied to the end of the beam, and r is the length of the beam. After converting the input force to proper current and vice versa on the output, the final closed loop model is given in Fig. 2.11 where f_{md} and f_h represent forces and correspond to the same quantities as given in Fig. 2.1.

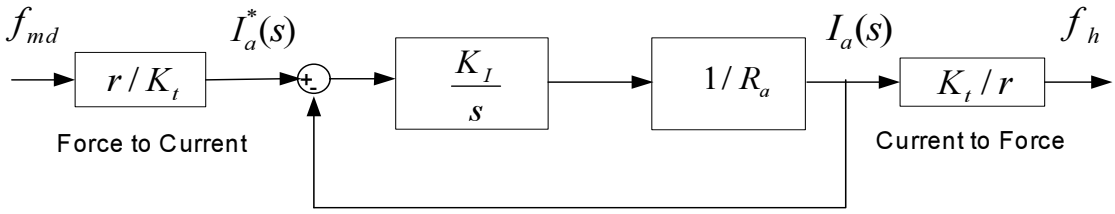

Figure 2.11: Master arm force control loop

2.2.3 Slave Arm

The slave arm is the robot which implements the desired task, depending on the motion commands from the master side whether the commands are velocity or position values. In our case, as a velocity-force architecture is used, so the the commands are velocity signals. The slave is modeled as a velocity controlled loop. Again a PI controller is used in the closed loop. A schematic diagram of a 1 d.o.f. slave arm is shown in Fig. 2.12.

The transfer function of a DC motor for motion control is given as [25]:

$$J\dot{\omega}_m(t) = T(t) \tag{2.25}$$

where J is the moment of intertia, ω is angular velocity, and $T(t)$ is motor torque.

Motor torque can be expressed as:

$$T(t) = K_t.I_a(t) \tag{2.26}$$

where K_t is the motor torque constant and I_a is the armature current. Equations (2.25) and (2.26) give us the following velocity control model:

$$\frac{\omega(s)}{I_a(s)} = \frac{K_t}{Js} \tag{2.27}$$

The open loop transfer function of slave velocity control loop with a PI controller is given as:

$$G_{os}(s) = \left(K_P + \frac{K_I}{s}\right)\left(\frac{K_t}{Js}\right) \tag{2.28}$$

which gives us the slave closed loop function as:

$$G_{cs}(s) = \frac{K_P K_t s + K_I K_t}{Js^2 + K_P K_t s + K_I K_t} \tag{2.29}$$

In order to change the angular to linear velocity to be used in the simulations, linear to angular velocity conversion ($v = r\omega$) is used. Slave arm velocity control loop is shown in Fig. 2.13.

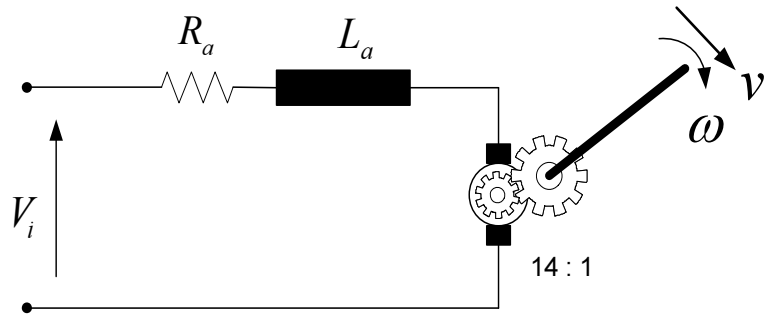

Figure 2.12: 1 d.o.f. slave arm showing gears and attached beam

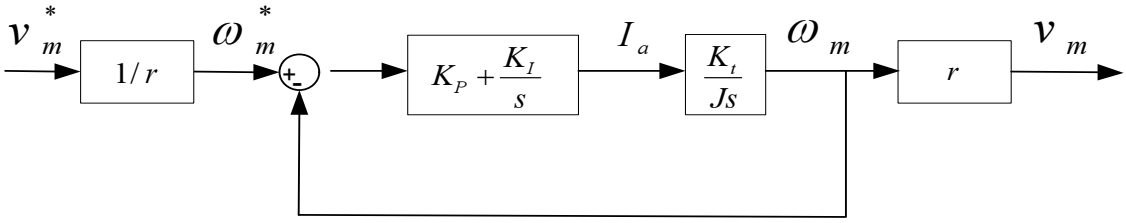

Figure 2.13: Block diagram of slave arm velocity control loop

2.2.4 Environment

Possible environment causalities[1] include impedance based (position/velocity as input and force as output), admittance based (force as input, position/velocity as output), or constraint based (position input / position output) causalities. In case of an impedance based environment (typical of many implemented systems), a virtual spring and damper in parallel are typically connected to the slave robot model. Fig. 2.14 shows an impedance based environment that is used in the simulations which can be described as:

$$f_e = k_e x + b_e \dot{x} \tag{2.30}$$

By varying the parameters k_e and b_e, a variable environment can be simulated as desired. In this work, we assume that both environment and human operator are passive, which means that they can not add energy to the system. In some applications , however, the environment may act as an active component depending on the surrounding conditions. Even in delay-free environments, a high stiffness constant can bring the system to instability. As the stiffness of environment increases ($K_e > 100$), the environment can be described as a high stiffness contact. In a variable environment simulation, we will consider a case when environment

[1]Causality refers to a temporal cause and effect relationship.

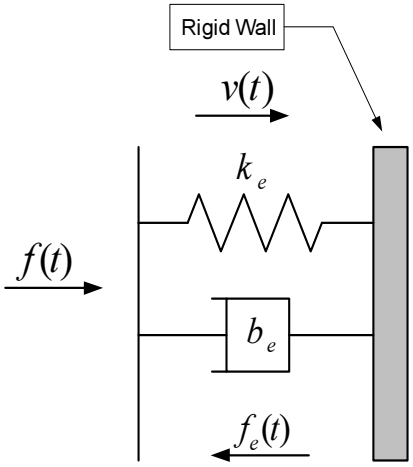

Figure 2.14: Impedance causality based model of the environment

stiffness changes from 0 to 10 N/m.

If the value of the environment damper coefficient is positive, this means the environment dissipates energy and can be called a passive element. Alternatively, if the value is negative, the environment adds energy to the system and is called an active environment.

2.2.5 Feedback Impedance

Teleoperation systems are designed to enable the carrying out of delicate tasks via master-slave robotic manipulators. Second only to stability, transparency is also a highly desirable property of a teleoperation system. When the teleoperation system is transparent, the operator feels as if he were manipulating the task directly.

Transparency is a well researched area in teleoperation. However, in the case of the presence of time delays, it becomes increasingly difficult to achieve high transparency in the system. As in a teleoperation system, the operator input is velocity and the output to operator is force, (see Fig. 2.1), the feedback force must be converted to velocity, in a simulation system, so as to compare the input and the output of the system in the same units. It is known that:

$$z = \frac{f}{v} \tag{2.31}$$

where z is impedance. In order to achieve high transparency, a close matching between master and environment impedances must be achieved. Following (2.30), as $v(s) = x(s).s$, the impedance of environment model is given by:

$$z_e = \left(\frac{k_e}{s}\right) + b_e \tag{2.32}$$

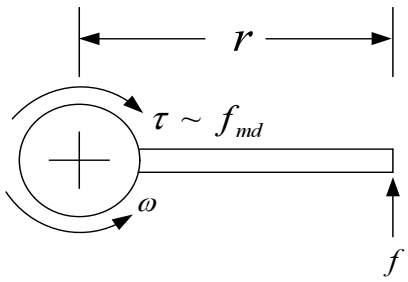

Figure 2.15: Physical model of master arm

Now the master impedance needs to be calculated. A single d.o.f. freedom master manipulator is used to display forces and read motion commands as shown in Fig. 2.15 where f is force exerted by human operator.

The dynamic equation of the master arm becomes:

$$f(t)r - \tau(t) = m\left(\frac{r}{2}\right)^2 \dot{\omega}(t) + b_m\omega(t) \tag{2.33}$$

where b_m represents the damping factor in master arm, r is the length- , m is the mass of the arm, and τ is the motor torque. In Laplace domain:

$$F(s) - \tau(s) = m\left(\frac{r}{2}\right)^2 s\omega(s) + b_m\omega(s) \tag{2.34}$$

where $F(s) = r.\mathscr{L}(f(t))$. Rearranging:

$$\frac{F(s) - \tau(s)}{\omega(s)} = m\left(\frac{r}{2}\right)^2 s + b_m \tag{2.35}$$

For linear velocity:

$$\frac{F(s) - \tau(s)}{V_m(s)} = \frac{m\left(\frac{r}{2}\right)^2 s + b_m}{r} \tag{2.36}$$

where $v_m(t)$ is the linear velocity of the edge of the beam attached to master arm. Following that and using (2.31):

$$V_m(s) = \frac{r(F(s) - \tau(s))}{\left(\frac{mr^2}{4}\right)s + b_m} \tag{2.37}$$

which gives us the impedance of master arm as:

$$\begin{aligned} z_m &= \frac{\left(\frac{mr^2}{4}\right)s + b_m}{r} \\ &= (1/r)\left[\left(\frac{mr^2}{4}\right)s + b_m\right] \end{aligned} \tag{2.38}$$

In order to achieve higher transparency, environment and slave impedances in (2.32) and (2.38) should be matched as closely as possible.

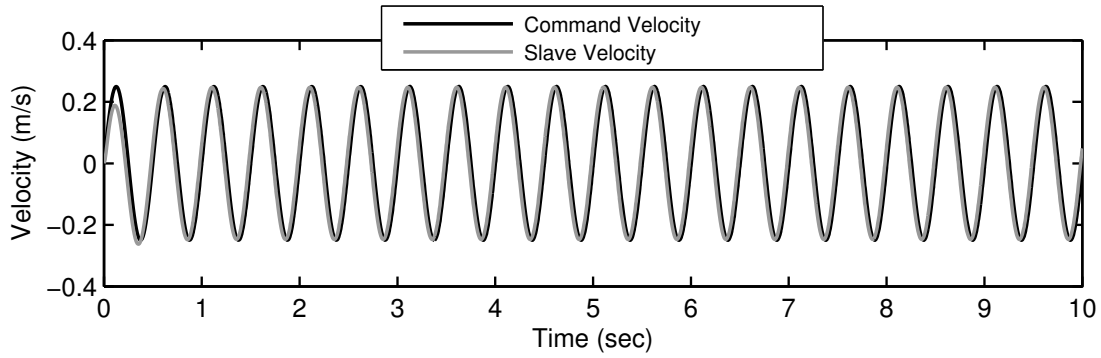

Figure 2.16: Teleoperation without time-delay, command and slave velocities

2.2.6 Communication Channel

The communication channel is the interface between the master and slave arms. As described earlier in chapter 1, it can be a radio link, internet, or any other communication link. In the simulation models used in this work, different kinds of delays will be employed as communication channel. In some cases constant time delays will be used. On the other hand, to highlight real-time issues, time-varying delays are better suitable and will be made use of later. In chapter 7, TrueTime Network based delays are used to evaluate the system performance.

2.3 Manifestation of the Problem using Simulations

Using the system modeled in Section 2.2, we can now proceed to the manifestation of the problem in actual systems. The notion of energy, as used in these results, is defined in section 5.1.1.

First the teleoperation system is simulated without any delay in the forward and backward channels, and because it consists of inherently stable components, the response is stable, as shown in Fig. 2.16.

Continuous time and discrete energies of the system measured at the master and slave sides of the network 2-port (Ref: Fig. 4.3) are shown in Figs. 2.17 and 2.18. It is clear that net energy in both continuous and discrete domain is zero and the system, having a lossless communication, is perfectly stable.

Now Figs. 2.19-2.20 show the same teleoperation system when simulated with delays of $200ms$ in the forward- and $300ms$ in the backward-channel, respectively. Clearly the net energy in the system is rapidly growing negative and because of the time delay, the system has become unstable.

These simulations solidly indicate the stability problem in bilaterally controlled delayed teleoperation. The solution to this problem, using Time Domain Passivity Control will be discussed in the succeeding chapters.

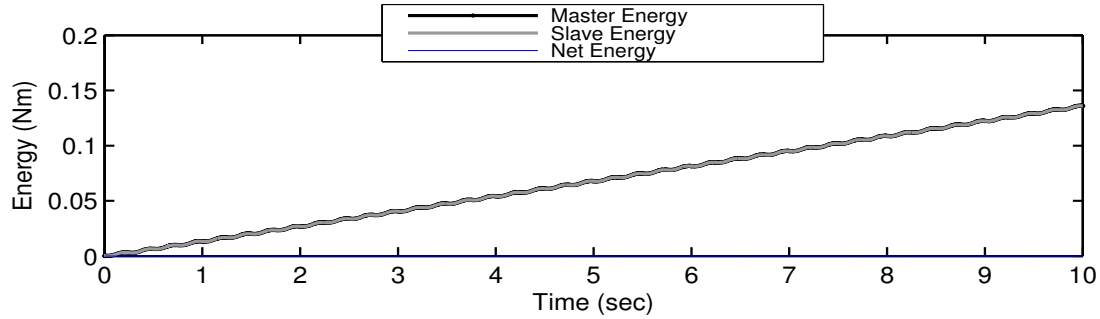

Figure 2.17: Teleoperation without time-delay, continuous-time energies

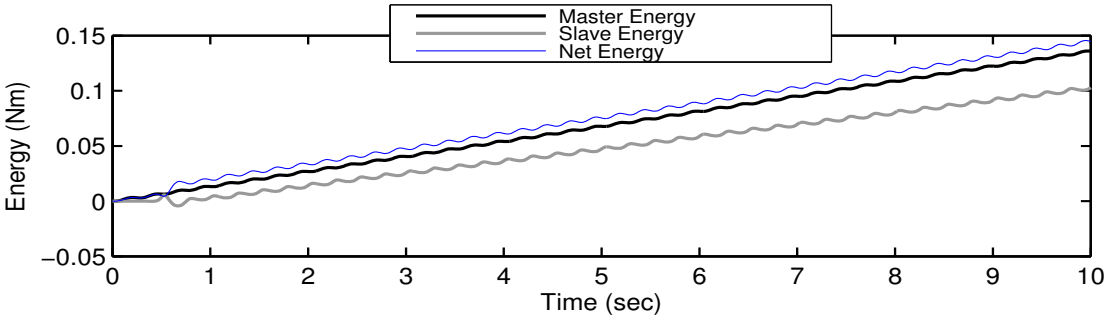

Figure 2.18: Teleoperation without time-delay, discrete energies

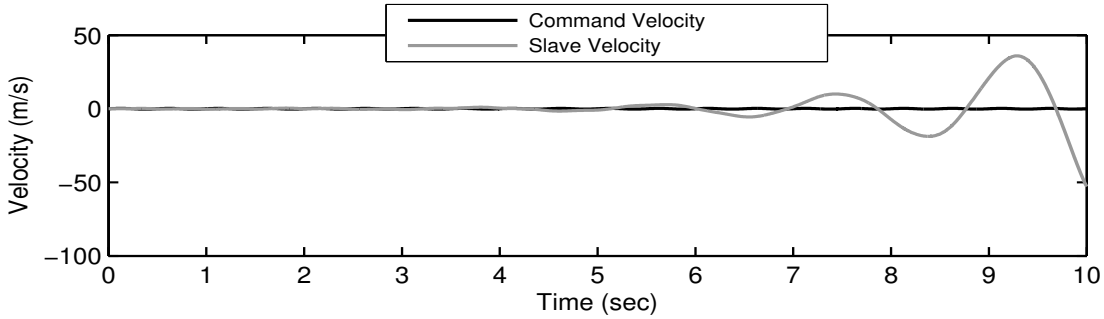

Figure 2.19: Teleoperation with time delay, $T_f = 200ms$, $T_b = 300ms$, command and slave velocities

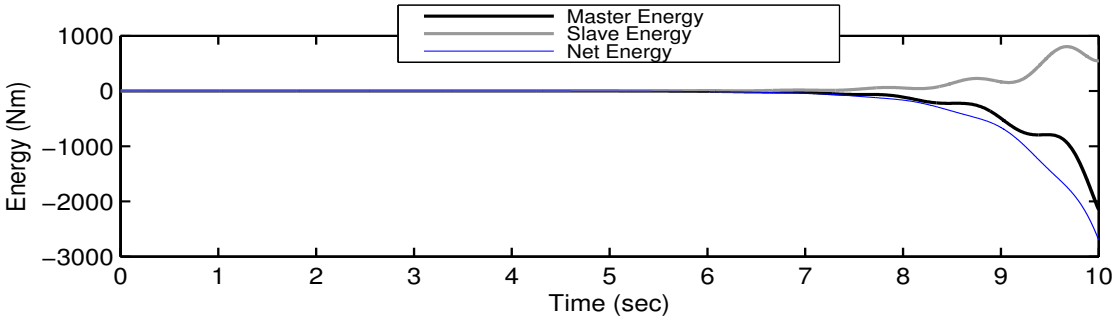

Figure 2.20: Teleoperation with time delay, $T_f = 200ms$, $T_b = 300ms$, continuous-time energies

Chapter 3

Literature Review

The literature in the stabilization of time-delay systems is vast and lies in many different domains of science and technology ranging from pure mathematics, electrical engineering to control systems engineering and mechatronics. In this chapter, a brief overview of the literature pertinent to the telepresence and teleoperation systems will be given. The problem of stability in the bilateral teleoperation with time delay was first reported as:*If forces are fed-back to the hand which also provides the positioning command to the manipulator, they will tend to move the operator's hand. With a delay, the feedback is not only the source of information, but may act as a disturbance input as well. Just as one would predict instability for any closed loop system having a long delay and a loop gain greater than unity, it would be expected that, if the delay and the rate at which feedback force changes with the position of the remote hand are great enough, a manipulator can become uncontrollable[3].* Two solutions were given for the problem, i) that the force feedback should be applied to the other hand of the operator rather than the one issuing motion commands, or ii) some form of supervisory control on the slave side should be implemented to limit the forces applied by the operator. Sheridan[1] defined many basic terms of teleoperation, telepresence, etc. as well as insisted that the research in these areas had direct transferability to the non-government sector for use in manufacturing, construction, mining, agriculture, medicine and other areas.

A very good work on the analysis and design of bilateral master-slave teleoperation systems is given by Yokokohji et. al. in [26]. The discussion does not take into account the effects of time-delays but gives an informative overview of stability of such systems with the help of passivity theory, impedance matching, as well as introduces the concept of performance indices for bilateral teleoperated systems in terms of position and force tracking. The performance indices are given as the minimization of the drift between the master and slave, position and force transfer functions, over the bandwidth of human operator manipulation frequencies as:

$$J_p = \int_0^{\omega_{max}} |G_{mp}(j\omega) - G_{sp}(j\omega)| \left| \frac{1}{1 + j\omega T} \right| d\omega \tag{3.1}$$

$$J_f = \int_0^{\omega_{max}} |G_{mf}(j\omega) - G_{sf}(j\omega)| \left| \frac{1}{1 + j\omega T} \right| d\omega \tag{3.2}$$

where J_p and J_f are position and force tracking performance indices. $G_{mp}(j\omega)$, $G_{sp}(j\omega)$, $G_{mf}(j\omega)$, and $G_{sf}(j\omega)$ are the transfer functions for the master-slave system from the operator's force τ_{op} to the master side displacement x_m, slave side displacement x_s, master side force f_m, and slave side force f_s, respectively.

It is shown that the operator is normally taken as passive but its parameters like damping and spring constants change during the operation. The authors have defined three kinds of ideal responses for master-slave systems as given below:

Ideal Response I The position responses x_m and x_s by the operator's input τ_{op} are identical, whatever the object dynamics are.

Ideal Response II The force responses f_m and f_s by the operator's input τ_{op} are identical, whatever the object dynamics are.

Ideal Response III Both the position response x_m and x_s, and the force responses f_m, and f_s by the operator's input τ_{op} are identical respectively, whatever the object dynamics are.

Ideal response III can be regarded as a final goal of master-slave systems or as an *Ideal Kinesthetic Coupling* . In using passivity for the stability, authors maintain that, passivity of the system can be a sufficient condition only when the system interacts with passive environments. The presented control schemes realize the ideal responses when the master and slave dynamics are available.

Stabilization of the time-delayed teleoperation systems is carried out using different techniques and the relevant literature can be categorized as given in the following sections.

3.1 Passivation

Anderson and Spong[8] published the first solid result on the stabilization of bilateral control of teleoperators with time delay by passivation of the system using scattering theory. The system consists of five sub-systems: the human operator, the master, the communication block, the slave, and the environment. In the working, it is essentially a hybrid system that switches between position (or velocity) and force control. The system can be represented as a network consisting of $n - ports$ where master, communication block, and slave are represented by two-ports and the operator and environment by one-ports as shown in figure (2.1)

An $n-port$ is characterized by the relationship between effort f(force, voltage), and flow v(velocity, current) which for a LTI one-port is given as the impedance described by the relationship:

$$f(s) = Z(s)v(s) \tag{3.3}$$

For a LTI two-port, the relationship is given by the *hybrid matrix* $H(s)$. Authors use scattering operator S to define passivity of the system which can be defined in terms of hybrid matrix. In simple words, *passivity* can be attributed to an n-port

that may dissipate energy but will never increase the total energy of a system, of which it is an element. Furthermore, it is proved that if the norm of the scattering operator of a system is less than or equal to zero, the system will be passive and hence stable. An important aspect of such an approach is that a coordinating torque term f_s, and not the contact force f_e is available to the communication block for transmission to the master. This setting requires the establishment of a local force feedback on the slave side *because* f_s *is insensitive to changes in* f_e.

The main contribution of the work is to choose a control law so that the two-port characteristics of the communication block are identical to a $2-port$ lossless (or less restrictively, passive) transmission line. Once the norm of the scattering matrix is less than unity, the system becomes stable even in the presence of time delays of large magnitude. However, the environment and the human operator belong to the class of passive systems. If the master contains active actuation, stability is no more guaranteed.

The authors give formal proof of their passivity-based approach for the stabilization of time-delayed teleoperated systems in [8]. The stability is proved for a full n-d.o.f. nonlinear system using PHIDE (Passive Hilbert-network with Identity-valued Dynamic Elements) derived from network modeling and operator theory. The stability of the system using PHIDE is guaranteed in the sense of Lyapunov. Power scaling, a basic requirement in teleoperation, is shown not to disturb the passivity of the system as far as the amplifier (element for power scaling) gain is constant and scalar valued. Main result of the work is proved by constructing a Lyapunov functional and using Tellegen's theorem that shows the asymptotic stability of master and slave velocities when, 1) both the human operator and environment can be modeled by a PHIDE network and, 2) the environment force, f_e, and the applied human force, f_h, are a bounded function of the human and environment states.

3.1.1 Wave Variable based Passivity

Niemeyer and Slotine[15] used wave variables based approach for the passivation in telerobotic system, validating, at the end, the results of [8]. The analysis by authros showed that wave-variables are the same as scattering approach used by Anderson and Spong[8]. It is reported that passivity features closure properties which implies that a combination of two passive systems connected in either feedback or in parallel configuration is again passive. The notion of wave scattering is shown to be closely related to the passivity formulation. Power flow in the system is separated into power -input and -output of the system which are then associated with the input and output waves.

A complimentary pair of wave variables $(\mathbf{u}, \mathbf{v})$ can be defined as:

$$\mathbf{u} = \frac{b\dot{\mathbf{x}} + \mathbf{f}}{\sqrt{2b}} \tag{3.4}$$

$$\mathbf{v} = \frac{b\dot{\mathbf{x}} - \mathbf{f}}{\sqrt{2b}} \tag{3.5}$$

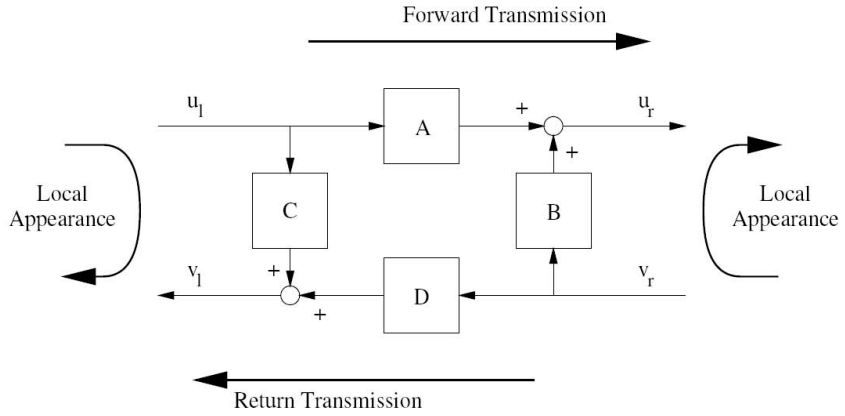

Figure 3.1: Wave variable based teleoperation

As the transformation is invertible[18], so the power variables can be computed using the following relations:

$$\dot{\mathbf{x}} = \frac{1}{\sqrt{2b}}(\mathbf{u}+\mathbf{v}) \tag{3.6}$$

$$\dot{\mathbf{f}} = \sqrt{\frac{b}{2}}(\mathbf{u}-\mathbf{v}) \tag{3.7}$$

where $\mathbf{f}$ and $\dot{\mathbf{x}}$ are standard power variables representing force and velocity, respectively.

On similar grounds, using the passivity formulation, the power flow in a 2-port with left-hand port l and right-hand port r can be written as[15]:

$$P = \dot{x}_l^T F_l - \dot{x}_r^T F_r = \frac{1}{2}u_l^T u_l - \frac{1}{2}v_l^T v_l + \frac{1}{2}u_r^T u_r - \frac{1}{2}v_r^T v_r \tag{3.8}$$

Vectors u_l and u_r, are interpreted as *input waves* while v_l and v_r are *output waves*. Fig. 3.1 shows a basic wave-variable based teleoperation system.

Because of the passivity of wave scattering, it is shown that if wave variables are transmitted instead of the power variables, the system becomes inherently passive, and thus stable regardless of the time delay present in communication. Wave impedance matching is also carried out at the end points to avoid wave reflections. It is argued that the master and slave manipulators should be velocity-commanded rather than force-commanded in order to avoid positional drift. This is also required for the impedance matching at the terminating points. Another important aspect of wave approach is wave filtering which enables filtering of the signals without affecting stability. It can be used to find derivative of force signal, noise reduction, as well as for the frequency shaping of the perceived information. Wave predictors are used in developed framework to establish an immediate local feedback loop on the operator side.

Finally, an adaptive controller is used on the slave side to provide consistent dynamic performance locally. It also helps to factor out most of the complex dynamics of the task from the dynamics perceived by the operator. The adaptive

tracking controller also uses wave filters to acquire the derivative of the force that can in turn be used as feed-forward to achieve desired tracking. The adaptive controller also serves to make a well-behaved contact by using a stiffness controller. The operator receives the forces as applied by the stiffness controller on the contact rather than the contact forces directly. These reflected forces are a good representation of the contact forces if the manipulator is easily back-drivable which can be ensured by providing a local force feedback loop. It is noted that reflecting data from a force sensor directly to the operator is not passive and should thus be avoided. The resulting controller in this case coincides with that of the Anderson and Spong[8].

A comparison of both scattering theory based and wave-variable based passivation approaches is given in [9]. It is also reported that the digital implementation of a continuous time passive system may no longer remain passive/stable which requires the introduction of strictly causal and stable linear filters to ensure stability of sampled-data master/slave systems.

A basic framework for system analysis and robot control using wave variables is given in [27]. The paper discusses alternate control approaches in the wave variable domain. An intuitive interpretation of wave variables in terms of symmetry, hybrid encoding, move or push commands, and wave impedance is also given. It states that a wave variable acts simultaneously as a force and a velocity command. While in contact, it creates a force, whereas in free space, motion is generated. Wave impedance b is presented as a tuning parameter to trade off the speed of motion and levels of force. Wave variables can also be used for filtering purposes which being in wave domain is inherently passive. Furthermore, wave controllers can be used in teleoperation and share the following characteristics:

1. The transmission paths consist of low-pass filters.

2. The reflective paths are high-pass filters, serving as local feedback during transients at high frequency.

3. The cut-off frequencies are same thus splitting effectively the wave signal between transmission and reflection.

Bandwidth limitation in the transformation can be enforced using simple low-pass filters. In order to overcome the position drift in the teleoperation applications, the authors present the idea of wave integrals, thus integrating the velocity to determine the tracking information and then transmitting this information alongside the normal wave variable. Both can be combined together and can be recovered using a stable filter on the other side. In this case, one variable encodes all the information- position, velocity, and force- required to operate the system. In the end, authors propose the use of pure wave controllers that can, i) enforce physical limitations by the unmodeled dynamics, and ii) further reduce the bandwidth to loosen the connection.

Adams et. al.[11] present an approach to guarantee stability in man-machine interface for a virtual coupling that acts as an artificial link between the haptic

display and the virtual environment. The system is modeled as a network of passive ports. The criterion to maximize the performance of the virtual coupling is to maximize the impedance and minimize the admittance of the artificial link irrespective of whether it is an impedance or an admittance device. It is assumed that the human operator and the environment are passive. Llewellyn's stability criterion is used to ensure unconditional stability while maximizing the performance. Using this method, optimal parameters for the spring-mass-damper model for the virtual coupling can be found and stability bounds can be generated for the given system. It should also be noted that the system can adapt to both impedance as well as admittance type display, and if a display is to perform as both types, 1) it has to be equipped with force sensors(admittance), and 2) should be back-drivable(impedance).

Passivation of bilateral operators with time-varying delay is discussed in [13]. First it is shown that when the wave-variables[15] are used for communication, the system is always passive when the time-delay is constant although no guarantee of acceptable performance is assumed with mere passivation. When the delay becomes time-varying, the system presented in [15] no longer remains passive and thus becomes unstable. In the model presented, the forward and backward time delays are taken as functions of time with the assumption that the bound on their rates of change is known. Then it is shown that putting suitable (variable) gains in both directions can render the system as passive. These variable gains can be thought of as energy dissipators dependent on time-delay. Simulation results show the effectiveness of the proposed approach. In order to improve the tracking performance, a saturator is put on the forward path on the client side that forces the slave velocity to remain within the allowable limits of the master velocity. This modification is shown to considerably improve the position tracking. In this work, the gains are not variable but the authors wish to continue in this direction to have switchable gains depending on the statistical properties of the delay.

Niemeyer et. al.[18] have given a comprehensive review of wave-variable based approaches for the stabilization of teleoperation systems.

Hirche and Buss [6] describe the use of impedance matching and low-pass filters in the passivity framework to obtain a smooth, near-transparent bilateral teleoperation. Time delays are taken to be constant though different in the forward and backward paths. The system, however, behaves well, in the presence of even time-varying delays. Design of impedance matching filters requires complete models of both the teleoperator, as well as the environment which is not a trivial problem. To overcome this issue, a numerical technique is used to find the parameters of a lead-lag filter that is based on minimizing the distance (in the Nyquist Diagram) between the current terminating impedance and the ideal terminating impedance. High frequency disturbances are canceled by using a low-pass wave filter in the forward path. It is shown that lowering the cut-off frequency of the filter decreases the distortion induced because of long-delays at the cost of additional phase (time) lag between master and teleoperator positions. Loss of single-packets, i.e., of every second packet does not affect the performance. However, if this loss is in a way that consecutive 50% of 1000 packets are lost, then it becomes crucial to sys-

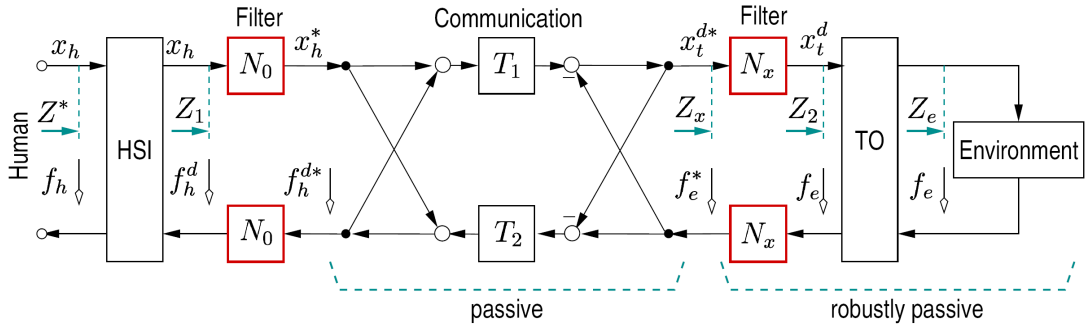

Figure 3.2: Force-Position Control

tem performance and stability. It is foreseen that use of IPv6 will help facilitate real-time teleoperation through better control over time-delay, jitter, and packet loss.

In [20, 4], Hirche and Buss give the performance evaluation of their passivation approach, in a force-position control, using real-time network emulation. The delay is assumed to be constant. Position control is used in place of velocity, mainly to avoid the drift between master and slave positions. In order to passivate the force-position architecture, impedance matching filters are used based on the environment model which is a second order spring-damper model. The proposed system uses a stricter passivity criterion named as IOSP(Input-Output Strict Passivity) that is met if there exists some $\alpha,\ \gamma > 0$ for which

$$\int_0^t u^T y d\tau \geq \int_0^t \alpha u^2 + \gamma y^2 d\tau \quad \forall t > 0 \tag{3.9}$$

holds, where u and y are input and output vectors of the passive element. The system block-diagram is given in figure (3.2). The impedance matching filters N_0 and N_x make the teleoperation transparent by making $Z_1 = Z_2$ where the HSI and teleoperator are assumed to be ideally matched in the impedance sense. Filter parameters are computed using an optimization in the Nyquist domain where transparency and passivity are taken as objective functions. Because the filter N_0 is based on the environment model, so an immediate local feedback is available to the HSI without delay. Experimental results are given showing good system performance in terms of stability, tracking, and transparency. It is shown that the impact of model uncertainty on the tracking performance decreases for lower delays.

Despite the benefits offered by wave-variables, they still lack a direct physical meaning when compared to their counterparts like force and velocity[18]. The absence of physical manifestation can pose challenges when designing controllers and it can be difficult to make use of intuitive insight in this process.

3.1.2 Time Domain Passivity Control

TDPC has been introduced by Hannaford and Ryu, et. al. [28, 29, 30, 31, 31, 32] and makes use of time domain properties of power variables to design an energy based control scheme in order to stabilize stiff haptic interfaces without time delay. As current work is based on an extension of their results in order to stabilize the delayed teleoperation, so this concept will be discussed in detail in chapter 5.

3.2 Lyapunov Functionals

Lyapunov functionals are the most studied tool for the stability analysis of non-linear systems. Considerable research in time-delay teleoperated systems has been carried out using Lyapunov functionals in order to guarantee the stability. The basic idea is to find a Lyapunov functional for a teleoperation system consisting, mostly of the energy variables or states of the system. In the following, a review of some of the relevant work is presented.

In [33], Gallegos et. al. have used some of the concepts from the passivation approach[8] to obtain the basic framework for the feedback like coordinating torque, etc. and then further modeled the system in a state vector form. In the presence of the time delay, the system should try to achieve the following objectives while ensuring asymptotic stability of the system:

1. $x_s(t) \rightarrow x_m(t-T)$

2. $v_s(t) \rightarrow v_m(t-T)$

3. $u_m(t) \rightarrow f_e(t-T)$

where x_s, v_s, x_m, v_m are the slave- and master-, positions and velocities, respectively. u_m is the master control input whereas f_e is the contact force. This state-space representation is then used to attain a Lyapunov function which is used to derive the conditions for the asymptotic stability of the system. It is also stated that by changing the control law from that given in [8], the master and slave can be made to experience the same environmental force.

Ni et. al.[34] discuss the application of Lyapunov functionals to design a set of stabilizing controllers for the force-reflecting teleoperators. These cost based controllers are then used in gain-switching fashion to stabilize and to enhance the transparency of a given teleoperation system.

Montestruque and Antsaklis[35] use Lyapunov functionals to study stability of model based NCS in the presence of time varying delays. It is shown that for stochastically modeled transmission delays, using *Almost Sure Stability* criterion gives extended stability margins in terms of delays than using Lyapunov functionals guarantees.

3.3 Other Work

Elhajj et. al. [36] discussed the design of a telerobotic system that is asymptotically stable in the presence of time delays without any assumption regarding latency, packet-loss, etc. The idea presented is very simple to understand. The system is referenced on events rather than time. The events occur in a deterministic order and no event can execute prior to its predecessor. The system waits until next event occurs. A similar approach in classical teleoperation is move-n-wait strategy. The only condition is that the robot be a stable system in the absence of any remote human operator which is not very restrictive. But it must be stated that, this stability is achieved by reducing the control bandwidth very severely. The reflected feedback is a function of the distance between the robot and remote environment which enables the human operator to get an intuitive knowledge of the position of remote objects by the magnitude of the force being fedback.

In [37], Cavusoglu et. al. use *small gain theorem* to find stability conditions for a teleoperation system capable of monitoring compliance changes in the remote environment. The authors suggest to target task based performance goals rather than seeking general ideal teleoperation response, a practice that generally yields marginally stable or poorly transparent teleoperation. It is also suggested to quantify and to incorporate human perception in the teleoperation control scheme.

Chapter 4

Stability and Passivity

Stability ensures that the output of a dynamic system will be bounded, at least asymptotically. On the other hand, passivity is concerned with the energies of a system and relates the energy balance in such a way that a passive system does not deliver more energy than the energy input into the system. Despite seemingly different, passivity and stability are strongly correlated. In fact, passivity is a sufficient condition for stability [38].

4.1 Stability

Stability for systems may be defined in many different ways, depending upon the needs of a particular application. The most common definitions of stability are given below[39, 40]:

4.1.1 Asymptotic Stability

A system is asymptotically stable if for all possible initial conditions, its zero-input response approaches zero with time. In a linear, time-invariant system with a rational transfer function $T(s)$, there is a zero-input response term for each denominator root of $T(s)$. A term expands or decays in time according to whether its roots are in the RHP or the LHP. Hence such a system is asymptotically stable if and only if all of the characteristic denominator roots of $T(s)$ are in the LHP. In state-space models, consider a system with n-dimensional state model $[A(t), B(t), C(t)]$, and the homogeneous equation

$$\dot{x}(t) = A(t)x(t),\ t > t_0 \tag{4.1}$$

with solution

$$x(t) = \Phi(t, t_0)x(t_0),\ t > t_0 \tag{4.2}$$

The system is said to be asymptotically stable if the solution $x(t)$ satisfies the condition $\|x(t)\| \to 0$ as $t \to \infty$ for any initial state $x(t_0)$ at initial time t_0. A system is asymptotically stable if, and only if,

$$\|\Phi(t, t_0)\| \to 0 \text{ as } t \to \infty \tag{4.3}$$

where $\|\Phi(t, t_0)\|$ is the matrix norm equal to the square root of the largest eigenvalue of $\Phi^T(t, t_0)\Phi(t, t_0)$.

4.1.2 Uniform Exponential Stability

A stronger notion of stability is uniform exponential stability which requires that positive constants c and λ exist so that

$$\|\Phi(t, t_0)\| \leq ce^{-\lambda(t-t_0)},\ t \geq t_0 \tag{4.4}$$

which also implies that

$$\|x(t)\| \leq ce^{-\lambda(t-t_0)}\|x(t_0)\|,\ t \geq t_0 \tag{4.5}$$

4.1.3 Bounded Input Bounded Output(BIBO) Stability

A system is BIBO stable if, for every bounded input, its output is bounded. A necessary and sufficient condition for BIBO stability of a linear, time-invariant system with rational transfer function $T(s)$ is also that all of the characteristic denominator roots of $T(s)$ be in the LHP. For state space models, if the system is described by equation (4.1) and the uniform exponential stability is given by equations (4.4) and (4.5), then the bounded entries of $B(t)$ and $C(t)$ ensure that the system is BIBO stable.

4.1.4 Impulse Response Stability

A system is said to be stable in the impulse response sense if its response to an impulse input approaches zero with time. For a unit impulse input to a linear, time-invariant system with transfer function $T(s)$, the Laplace transform of the output is

$$\begin{aligned} Y(s) &= T(s) \cdot 1 \\ y(t) &= \mathscr{L}^{-1}[T(s)] = h(1) \end{aligned}$$

where the last term is quite common in communication literature. Expanding a rational transfer function $T(s)$ in partial fractions gives response terms which decay in time only if the roots of the characteristic polynomial of $T(s)$ are all in the LHP. For linear, time-varying systems, one must consider the impulse response as a function of the starting time of the impulse, because such systems behave differently at different times.

4.1.5 Lyapunov Stability

Physical systems, particularly those which are time-varying and non-linear, can be considered to stable if the provided energy is conserved in them, for their response

to "blow up" would require a continual supply of energy. Lyapunov stability is a means of checking the stability of a system in terms of the energy of the system . Lyapunov's Direct Method makes use of certain functions which are defined below:

Positive Semi-Definite Function: A scalar function (of a possibly vector valued variable), $f(x)$, is said to be positive semi-definite if (i) $f(x) \geq 0$ for all x, (ii) $f(x) = 0$ if and only if x=0. Strict equality in (i) makes it a **Positive Definite Function**.

Positive Definite Matrix: An $n \times n$ matrix, $Q = Q^T$, is said to be positive definite if, for all $x \in \mathbb{R}^n$, $f(x) = x^T Q x$ is a positive definite function.

Negative Semi-Definite Function: A scalar function (of a possibly vector variable), $f(x)$, is said to be negative semi-definite if (i) $f(x) \leq 0$ for all x, (ii) $f(x) = 0$ if and only if x=0. Strict equality in (i) makes it a **Negative Definite Function**.

Theorem 4.1.1 *(Lyapunov's Direct Method) The origin is stable if there is a continuously differentiable positive definite function $V(x)$ so that $\dot{V}(x)$ is negative semi-definite, and asymptotically stable if $\dot{V}(x)$ is negative definite.*

If a Lyapunov function is to be plotted against time, it can be seen that it is always a positive function with a negative derivative. This method is a generalization of the idea that if there is some *measure of energy* in a system, then one can study the rate of change of the energy of the system to ensure its stability. By the "Monotone Convergence Theorem", the function must converge to some limiting value. In particular, it must converge to the point where $V(x) = \dot{V}(x) = 0$ as $t \to \infty$. But $V(x) = 0$ only when $x = 0$. This means that the system with state x must have its state convergence to the origin. This is basically the same as asymptotic stability. For the state-space model given in equation (4.1), if one changes it to a time-invariant system, the Lyapunov stability theorem is given as:

Theorem 4.1.2 *(Stability Theorem) Suppose that $u = 0$ and there exist two positive-definite matrices, $P > 0$ and $Q > 0$ such that*

$$A^T P + PA + Q = 0$$

then the given system is stable in the Lyapunov sense.

Fig 4.1 shows a depiction of asymptotic stability in Lyapunov sense where the Lyapunov function, comprising of the states x_1 and x_2, of a given dynamic system converges to an equilibrium point.

4.1.6 ISS(Input to State Stability)

There are conceptually two different ways of formulating the notion of stability of control systems. One relies on the input-output relationships and automatically translates to the concept of BIBO stability. On the other hand, there is a

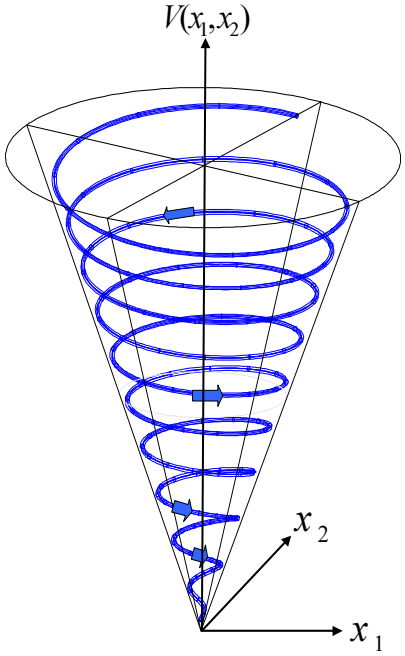

Figure 4.1: Illustration of Lyapunov Stability

state-space approach to systems and stability, where the basic object is a forced dynamical system, typically described by differential or difference equations. In this approach, a standard notion of stability given in section (4.1.5) as Lyapunov stability is used. A new perspective of stability given by Sontag[41] known as ISS(Input to State Stability) addresses the question as to what extent Lyapunov-like stability notions for a state-space system are related to the stability described in the terms of input and output. This concept is now being used to define the stability of many non-linear and time-delayed systems. For linear systems, generally, the definitions for state-space and input-output stability are equivalent. But for non-linear systems, these two concepts are not so closely related. Consider a system given as

$$\dot{x} = f(x, u) \tag{4.6}$$

where $x \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$. It is assumed that $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz and satisfies $f(0, 0) = 0$.

A Lipschitz function f is such that

$$|f(x) - f(y)| \leq C|x - y|$$

for all x and y, where C is a constant independent of x and y. For example, any function with a bounded first derivative must be Lipschitz.

Controls or *inputs* are measurable locally essentially bounded functions:

$$u : \mathbb{R} \geq 0 \to \mathbb{R}^m$$

The set of all such functions is denoted by $L^m_{\infty,e}$ (Lebesque measurable) and one denotes:

$$\|u\|_\infty = \text{ess sup}\{|u(t)|, t \geq 0\} \leq \infty$$

where *ess sup* is the essential supermum of a function. For each $x_0 \in \mathbb{R}^n$ and each $u \in L^m_\infty$, $x(t, x_0, u)$ denotes the trajectory of the system (4.6) with initial state $x(0) = x_0$ and input u. This is a priori defined only on some maximal interval $[0, T_{x_0,u}]$, with $T_{x_0,u} \leq +\infty$. If the initial state and input are clear from the context, one writes just $x(\cdot)$ for the ensuing trajectory. The system is *(forward-) complete* if $T_{x_0,u} = +\infty$ for all x_0 and u. The simplest way to introduce the notion of ISS system is a generalization of GAS, global asymptotic stability of the trivial solution $x \equiv 0$ for (4.6) where $u = 0$. GAS requires the system to be *complete* and the following two properties hold:

1. *Stability:* the map $x_0 \mapsto x(\cdot)$ is continuous at 0, when seen as a map from $\mathbb{R}^n$ into $\mathcal{C}^0([0, +\infty), \mathbb{R}^n)$, and

2. *Attractivity:* which is attractivity of the origin and is defined as: $\lim_{t \to +\infty} |x(t, x(t_0))| = 0$

The foregoing definition of GAS can be easily related to the graphical depiction of Lyapunov function given in Fig. 4.1. A system will be GAS when the hyper-cone becomes of infinite dimension and no matter where the states are located in hyperplane, $V(x)$ will always converge to the equilibrium point.

The system (4.6) is said to be ISS(input to state stable) if it is complete and the following properties, which now involve nonzero inputs, hold [41]:

1. the map $(x_0, u) \mapsto x(\cdot)$ is continuous at $(0, 0)$ (seen as a map from $\mathbb{R}^n \times L^m_\infty$ to $\mathcal{C}^0([0, +\infty), \mathbb{R}^n)$, and

2. there exists a "nonlinear asymptotic gain" $\gamma \in \mathcal{K}$ so that

$$\overline{\lim_{t \to +\infty}} |x(t, x_0, u) \leq \gamma(\|u\|_\infty)$$

uniformly on x_0 in any compact and all u.

The class $\mathcal{K}$ consists of all functions $\gamma : \mathbb{R} \geq 0 \mapsto \mathbb{R} \geq 0$ which are continuous, strictly increasing, and satisfy $\gamma(0) = 0$. The uniformity requirement means, explicitly: for each r and ϵ positive, there is a $T > 0$ so that $|x(t, x_0, u)| \leq \epsilon + \gamma(\|u\|_\infty)$ for all u and all $|x_0| \leq r$ and $t \geq T$.

4.1.7 Stability and Stabilizability of Time-Delay Systems

Stability is the basic property of any system. The stability problem of a system with time delay is much more complex than the one of a non-delayed system. Stability and stabilizability concepts for time-delay systems can be defined in two major classifications[42]:

- Deterministic time-delay systems
- Stochastic time-delay systems

Brief definitions for both of these kinds are given in the following sections.

Deterministic Time-Delay Systems

A general mathematical representation of the class of dynamical linear systems with time delay can be described by the following dynamics:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + A_d(t)x(t-\tau) + B_1(t)w(t) + B(t)u(t) \\ z(t) &= C_1(t)x(t) + D_{11}(t)w(t) + D_{12}(t)u(t) \\ y(t) &= C_2(t)x(t) + D_{21}(t)w(t) \\ x(t) &= \phi(t), t \in [-\tau, 0] \end{aligned} \tag{4.7}$$

where $w(t) \in \mathbb{R}^l$ is the square-integrable disturbance input vector at time t which is unknown but not necessarily random, $z(t) \in \mathbb{R}^p$ is the controlled output vector at time t, τ is the time delay, ϕ is the initial function. Time-varying matrices $A(t)$, $A_d(t)$, $B_1(t)$, $B(t)$, $C_1(t)$, $C_2(t)$, $D_{11}(t)$, $D_{12}(t)$, and $D_{21}(t)$ can be defined to include parametric uncertainties. For details, see [42].

Definition 4.1.1 *(Stability) System (4.7) with $u(t) \equiv 0$ and $w(t) \equiv 0$ is said to be stable if for every positive ϵ, there exists a positive δ, which may depend on the initial time and ϵ, such that if $\|\phi(\cdot)\| < \delta$, then $\|x(t, t_0, \phi(\cdot))\| < \epsilon \ \forall t \geq t_0$.*

The norm used is the *standard uniform norm* defined as

$$\|\phi(\cdot)\| = \max_{s \in [-\overline{\tau}, 0]} \|\phi(s)\|$$

Definition 4.1.2 *(Asymptotic Stability) The given system would be asymptotically stable if δ is chosen such that the solution $x(t, t_0, \phi(\cdot))$ goes to zero as time t goes to infinity.*

Remark 4.1.1 *System (4.7) with $u(t) \equiv 0$ is referred to as unforced system or free system.*

Definition 4.1.3 *(Equilibrium State)A state $x_0 \in \mathbb{R}^n$ is called an equilibrium state of system (4.7) if for some $t_0 \geq 0$*

$$x(t_0, \phi(\cdot)) = x_0 \Longrightarrow x(t, \phi) = x_0, \forall t \geq t_0 \tag{4.8}$$

This means that once the state trajectory reaches the point x_0, it will stay there forever.

Definition 4.1.4 *(Zero State Equilibrium Stability Measures) The zero state equilibrium is said to be*

(a) stable as $t \to \infty$, if for any given positive numbers t_0 and ϵ, there exists $\delta > 0$ that may depend on t_0 and ϵ such that, if

$$\max_{t_0 \le t \le t_0 + \overline{\tau}} \|x(t)\| \le \delta \tag{4.9}$$

then

$$\max_{t_0 \le t < \infty} \|x(t)\| \le \epsilon \tag{4.10}$$

holds;

(b) uniformly stable if, for any given $\epsilon > 0$, there exists a $\delta > 0$ (dependent on ϵ, but not on t_0 such that if $x(t)$ satisfies (4.9) for any $t_0 \ge 0$, then $x(t)$ satisfies (4.10).

(c) asymptotically stable if

(i) it is stable, and

(ii) for each $t_0 > 0$, there is a $\delta(t_0)$ such that

$$\max_{t \to \infty} x(t) = 0, \tag{4.11}$$

for any initial $\phi(\cdot)$ in $\{g \in C[-\overline{\tau}, 0] | \|g\| \le \delta(t_0)\}$

In the case of time-delay systems, it is required that the stability test depends on the system time delay in order to define the stability boundaries.

Definition 4.1.5 *(Admissible Control) A piecewise continuous function $u : [t_0, \infty) \to \mathbb{R}^m$ is called an admissible control, or control, for simplicity, if for an arbitrary $(t_1, \psi) \in [t_0, \infty) \times C[-\overline{\tau}, 0]$, (4.7) has a solution on $[t_1, \infty)$ with initial condition ψ under this control law $u(\cdot)$.*

Definition 4.1.6 *(Stabilizable System) A dynamical time delay system (4.7) with no disturbance ($w(t) \equiv 0$) and no parametric uncertainties, is said to be stabilizable if there exists a control law $u(t)$ such that the closed-loop system is stable in the sense of Definition (4.1.1).*

Stochastic Time-Delay Systems

The study of dynamical systems with random variations in their structures, in the form of parameter changes, component failures, etc. is becoming the focus of research in the recent years. Markov Jump Linear Systems(MJLS), for example, is an important class of such stochastic dynamical systems. A MJLS is a hybrid system with a state vector that has two components, $x(t)$ and $r(t)$. The first one is in general referred to as the state and the second one is referred to as the mode. The system jumps from one mode to another in a random way whereas the switching between the modes is governed by a continuous-time Markov process with discrete and finite state space. If there is only one mode, the system behaves

like a deterministic system. A general mathematical representation of the class of dynamical linear systems with Markov jump and time delay can be described by the following dynamics:

$$\begin{aligned}
\dot{x}(t) &= A(r_t,t)x(t) + A_d(r_t,t)x(t-\tau) + B_1(r_t,t)w(t) + B(r_t,t)u(t) \\
z(t) &= C_1(r_t,t)x(t) + D_{11}(r_t,t)w(t) + D_{12}(r_t,t)u(t) \\
y(t) &= C_2(r_t,t)x(t) + D_{21}(r_t,t)w(t) \\
x(t) &= \phi(t), t \in [-\tau, 0]
\end{aligned} \tag{4.12}$$

where $A(r_t,t)$, $A_d(r_t,t)$, $B_1(r_t,t)$, $B(r_t,t)$, $C_1(r_t,t)$, $D_{11}(r_t,t)$, $D_{12}(r_t,t)$, $C_2(r_t,t)$, $D_{21}(r_t,t)$ are time-varing system matrices depending on the current mode (r_t) of Markov process at time t.

The switching between the different modes of the stochastic systems is described by the probability transitions

$$P[r_{t+\Delta t} = j | r_t = i] = \begin{cases} \lambda_{ij}\Delta T + o(\Delta T) & \text{if } i \neq j \\ 1 + \lambda_{ii}\Delta T + o(\Delta T) & \text{otherwise} \end{cases} \tag{4.13}$$

with $\lambda_{ij} \geq 0$ for all $i \neq j$, $\lambda_{ii} = -\sum_{j \neq i} \lambda_{ij}$ and $\lim_{\Delta t \to 0} \frac{o(\Delta T)}{\Delta T} = 0$ and r_t represents a Markov process describing the system mode at time t.

Remark 4.1.2 *Note that the jump rates λ_{ij} for all i and j in $\mathcal{S}$ are constant, where $\mathcal{S}$ is the set of all possible modes of the system. The case when these rates are dependent on $x(t)$ and $u(t)$ is of more theoretical nature and is not addressed in the following definitions.*

Definition 4.1.7 *(Stochastic Stability) System (4.12) is said to be*

(a) stochastically stable (SS) if there exists a constant $T(r_0, \phi(\cdot))$ such that

$$\mathbb{E}\left[\int_0^\infty \|x(t)\|^2 dt | \phi(\cdot), r_0\right] \leq T(r_0, \phi(\cdot)) \tag{4.14}$$

where L.H.S. is conditional expectation given ϕ and r_0.

(b) mean square stable (MSS) if

$$\lim_{t \to \infty} \mathbb{E}\left[\|x(t)\|^2\right] = 0 \tag{4.15}$$

holds for any initial condition $(r_0, \phi(\cdot))$; and

(c) mean exponentially stable (MES) if there exist constants $\alpha > 0$, $\beta > 0$ such that the following holds for any initial conditions $(r_0, \phi(\cdot))$:

$$\mathbb{E}[\|x(t)\|^2 | \phi(\cdot), r_0] \leq \alpha \|\phi(\cdot)\| e^{-\beta t} \tag{4.16}$$

Remark 4.1.3 *MES implies MSS and SS.*

If the admissible parametric uncertainties are also present in the system matrices, then the stochastic stability definitions change to robust stochastic stability.

Definition 4.1.8 *(Robust Stochastic Stability) System (4.12) with $u(t) \equiv 0$ and $w(t)$ said to be*

(a) robustly stochastically stable (RSS) if there exists a constant $T(r_0, \phi(\cdot))$ such that

$$\mathbb{E}\left[\int_0^\infty \|x(t)\|^2 dt | \phi(\cdot), r_0\right] \leq T(r_0, \phi(\cdot)) \tag{4.17}$$

holds for all admissible uncertainties; and

(b) robust mean exponentially stable (RMES) if there exist constants $\alpha > 0$, $\beta > 0$ such that the following holds

$$\mathbb{E}[\|x(t)\|^2 | \phi(\cdot), r_0] \leq \alpha \|\phi(\cdot)\| e^{-\beta t} \tag{4.18}$$

Remark 4.1.4 *Note that RMES implies RSS.*

Definition 4.1.9 *(Stabilizability) System (4.12) is said to be stabilizable in the SS sense if there exists a state feedback controller*

$$u(t) = K(r_t)x(t) \tag{4.19}$$

such that the closed-loop system is SS, where $K(i)$, $i \in \mathcal{S}$ are constant gain matrices.

In the presence of uncertainties, robust stabilizability is defined as:

Definition 4.1.10 *System (4.12) is said to be robustly stabilizable in the RSS sense if there exists a stable feedback controller (4.19) such that the closed-loop system is RSS, where $K(i)$, $i \in \mathcal{S}$ are constant gain matrices.*

4.2 Passivity

Passivity is a sufficient condition for the stability of a 2-port coupled to an arbitrary network [38]. Therefore, most of the stabilization schemes that are based solely on passivity, are rather conservative due the use of over damping to effectuate passivity. In haptic literature, human operator is often taken as a passive element on frequencies of interest[43]. Environment is also considered to be passive. Master, and slave, even if not, can be made passive by proper control schemes. The communication channel, however, if containing delay, can be a difficult task to be shown passive. Different approaches have been suggested to passivate it, like scattering theory[8], wave-variables[15], impedance-matching[4], etc. Using passivity for stability analysis offers following advantages[44]:

- Intuitive and straight-forward energy concepts.
- Global stability conclusions can be stated by considering individual blocks.
- Can be applied to both linear and non-linear system.

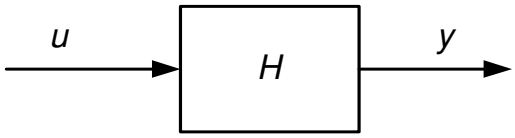

Figure 4.2: Input-Output description of a general dynamic system

4.2.1 Definitions

Let us take a dynamic system H, shown in Fig.4.2 as:

$$(H)\begin{cases} \dot{x} &= f(x,y), \quad x \in \mathbb{R}^n \\ y &= h(x,u), \quad u,y \in \mathbb{R}^m \end{cases} \tag{4.20}$$

In literature, there are quite a number of good works showing the relationship between passivity and stability of such a system, see e.g., [45, 46]. In the following, the stability of passive systems is briefly described.

Definition 4.2.1 *(Dissipitivity) Assume that associated with the system H given in (4.20) is a function $w : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, called the "supply rate", which is locally integrable for every $u \in U$, that is, it satisfies $\int_{t_0}^{t_1} |w(u(t), y(t))|dt < \infty$ for all $t_0 \leq t_1$. Let X be a connected subset of $\mathbb{R}^n$ containing the origin. One says that the system H is "dissipative" in X with the supply rate $w(u, y)$ if there exists a function $S(x)$, $S(0) = 0$, such that for all $x \in X$:*

$$S(x) \geq 0 \text{ and } S(x(T)) - S(x(0)) \leq \int_0^T w(u(t), y(t))dt \tag{4.21}$$

for all $u \in U$ and all $T \geq 0$ such that $x(t) \in X$ for all $t \in [0, T]$. The function $S(x)$ is then called a "storage function".

This definition can then be used to specify passivity as:

Definition 4.2.2 *(Passivity) System H is said to be "passive" if it is dissipative with supply rate $w(u, y) = u^T y$.*

Here $u^T y$ denotes the net power and for an impedance causality, u and y will be replaced by v(velocity) and f(force), respectively. The above definition will be used to establish passivity in chapters 5-7.

Definition 4.2.3 *(Zero State Detectibilty (ZSD) and Zero State Observability (ZSO)) Consider the system H with zero input; that is $\dot{x} = f(x, 0)$, $y = h(x, 0)$, and let $Z \subset \mathbb{R}^n$ be its largest positively invariant set contained in $\{x \in \mathbb{R}^n | y = h(x, 0) = 0\}$. It is said that H is zero-state detectable (ZSD) if $x = 0$ is asymptotically stable conditionally to Z. If $Z = 0$, one says that H is zero-state observable (ZSO).*

4.3 Stability of Passive Systems

The following theorem establishes the relation between passivity and stability:

Theorem 4.3.1 *(Passivity and Stability) Let the system H be passive with a C^1 (Continuously Differentiable) storage function S and $h(x,u)$ be C^1 in u for all x. Then the following properties hold:*

1. *If S is positive definite, then the equilibrium $x=0$ of H with $u=0$ is stable.*
2. *If H is ZSD, then the equilibrium $x=0$ of H with $u=0$ is stable.*
3. *When there is no throughput, $y=h(x)$, then the feedback $u=-y$ achieves asymptotic stability of $x=0$ if and only if H is ZSD.*

When the storage function S is radially unbounded, these properties are global.

For proof of the theorem and further discussion on stability of passive systems, see [46].

4.4 Passivity In Teleoperators

Let E_i denote the initial energy of a system. Following Definition 4.2.2, and Theorem 4.3.1, a teleoperation system will be called passive and stable if:

$$E(t) = \int_0^t P d\tau + E_i \geq 0 \tag{4.22}$$

where $E(t)$ is the total energy of the system at time t. P denotes the net power at input and output ports. Assuming initial system energy to be zero, a well-known expression for energy of the system is obtained:

$$E(t) = \int_0^t P d\tau = \int_0^t \boldsymbol{u}^T \boldsymbol{y}\ d\tau \geq 0 \tag{4.23}$$

$\boldsymbol{u}$ and $\boldsymbol{y}$ represent system input and output vectors respectively. In the case of a teleoperation system, they are usually the causal pair $\boldsymbol{f}$ and $\boldsymbol{v}$. As an n-port is characterized by the causal relationship between effort f(force, voltage), and flow v(velocity, current). So for a network 2-port, as shown in Fig. 4.3, Equation 4.23 changes to:

$$E(t) = \int_0^t P d\tau = \int_0^t (f_1(\tau)v_1(\tau) - f_2(\tau)v_2(\tau))d\tau \geq 0 \tag{4.24}$$

Signs should be carefully selected in this expression depending on the flow of energy and should remain consistent throughout the energy calculations. Here f_1, v_1 are considered to be input- whereas f_2, v_2 are output-variables. Equation 4.24 means that a passive system must not generate energy by itself. It can only dissipate the

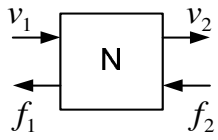

Figure 4.3: A network 2-port

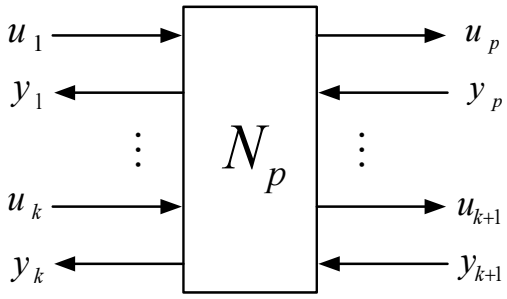

Figure 4.4: A network p-port

input energy or in ideal conditions can function as a lossless channel in which case $E(t) = 0$.

For a network of p connected elements, as shown in Fig. 4.4, with each port having a causal relationship between input u and output y, the condition of passivity (see [47]) can be written as:

$$E_{total}(t) = \sum_{k=1}^{p} E_k(t) \geq 0 \tag{4.25}$$

In certain teleoperation systems, one may come across such network 2-ports that not only dissipate energy but also induce energy other than conventional force and velocity information. For example, when the link between master and slave acts as a controller then there can be an external energy intake in the form of electrical energy. In expanding (4.25), these additional energy sources must be taken into account for a correct assessment of passivity.

Chapter 5

Time Domain Passivity Control

Passivity can be perceived as a tool that can be utilized in a number of ways to guarantee the stability of dynamic systems. For example wave variables, as described in section 3.1.1, transform the power variables into wave-domain and then use the passive transmission property of wave variables to ensure stability of time-delay systems. Time Domain Passivity Control (TDPC) is another manifestation of stabilizing properties of passivity. TDPC, introduced by Hannaford and Ryu, is a rather new concept in the stabilization of haptic interfaces[44]. The benefits of this method include a direct use of power variables and the provision of maximum transparency. The latter is achieved through the fact that TDPC stays out of the control loop as long as the system remains stable. Its function begins with the detection of system becoming active. It then regulates the energy so as to drive the system back to passivity and hence stability. In this chapter a brief description of TDPC is given. It should be noted that Time Domain Passivity Control in its original form does not deal with the issue of communication channel delay and thus extensions to it are proposed by the author in Chapters 6 and 7.

5.1 Introduction

TDPC approach does not require for the power variables to be transformed into wave-variables. Rather a very clear notion of energy is used to define passivity of the system. The proposed framework has been used in several applications to stabilize teleoperation systems showing very good results, see, for example, [28, 29, 30]. Time domain passivity control is based on two components:

1. Passivity Observer (PO)

2. Passivity Controller (PC)

A passivity observer keeps track of the energies i) that go into the system and, ii) the ones that come out of it. If at any moment, a net value of these energies show an active behavior of the given system in the form of negative energy, then Passivity Controller is used to dissipate any surplus energy violating the passivity.

Considering Fig. 4.4, a network $p-port$ with initial energy storage $E(0)$ is passive if any only if [48]:

$$\int_0^t (f_1(\tau)v_1(\tau) + f_2(\tau)v_2(\tau) + \cdots + f_p(\tau)v_p(\tau))d\tau + E(0) \geq 0, \ \forall t \geq 0 \qquad (5.1)$$

for all admissible forces $(f_1, f_2 \ldots f_p)$ and velocities $(v_1, v_2, \ldots, v_p)$. The *positive sign* will change to a *negative* one, should a particular port withdraws energy from the system. Keeping this definition in view, Passivity Observer and Passivity Controller can now be easily described.

5.1.1 Continuous and Discrete Energies

Througout this work, for a network 2-port, *discrete energy* at time t_n is defined by the following equation:

$$E_{net}(n) = T_s \left[f_1(n)v_1(n) - f_2(n)v_2(n)\right] \qquad (5.2)$$

Here the negative sign indicates the difference between input- and output-energy. The *continuous energy*, on the other hand is defined as:

$$E(t) = \int_0^t (f_1(\tau)v_1(\tau) - f_2(\tau)v_2(\tau))d\tau \qquad (5.3)$$

This quantitiy is computed by continuous-time integration of the respective quantities in a Simulink model.

5.2 Passivity Observer

As stated earlier, a PO is used to monitor the power flow through a network $n-port$. Considering a teleoperation $p-port$, PO must make use of force and velocity signals to provide an estimate of the net system energy. Assuming the rectangular summation of power values to be a good estimate of energy, a discrete value PO can be written as[30]:

$$E_{net}(n) = T_s \sum_{k=0}^{p} [f_1(k)v_1(k) + f_2(k)v_2(k) + \cdots + f_p(k)v_p(k)] \qquad (5.4)$$

where $E_{net}(n)$ is the estimate of net energy at time t_n and T_s is sufficiently small sampling interval. Keeping (5.1) in view, if $E_{net} \geq 0$, it means that the given network $p-port$ is dissipating energy and the system is thus stable. On the contrary, if $E_{net} < 0$, it shows that the network $p-port$ has generated some energy (means induction of energy from unaccounted-for sources) and the system is no more passive hence stability can not be guaranteed. The net negative energy, i.e., $-E_{net}$ is the amount of *surplus* or *generated energy*. Fig. 5.1 shows a PO monitoring the the energy flow through a network $2-port$.

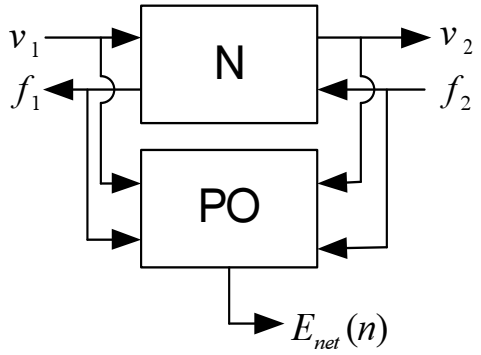

Figure 5.1: PO estimating net energy

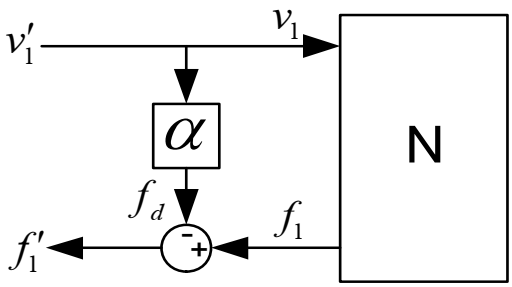

Figure 5.2: Series PC

5.3 Passivity Controller

Depending on the states of a teleoperation system, the net energy E_{net} may change over time and at times may even get negative according to the stability properties of the system. Once $E_{net} < 0$, the system can be regarded as unstable. Now the PC will engage itself in the control loop to dissipate the extra energy induced because of active components in the system. Passivity controller can be attached to the network port depending on the causality of the signal. In the case of a series passivity controller, as shown in Fig. 5.2, the PC dissipates the surplus negative energy in the form of modifications in reflected force signals while preserving velocity signal.

Considering the network $1-port$ given in Fig. 5.2:

$$v_1'(n) = v_1(n) \tag{5.5}$$

which means the velocity is preserved in this controller. The force equation

can be written as:

$$\begin{aligned} f_1'(n) &= f_1(n) - \alpha(n)v_1'(n) \\ &= f_1(n) - \alpha(n)v_1(n) \\ &= f_1(n) - f_d(n) \end{aligned} \tag{5.6}$$

where $f_d(n)$ is the correction in force signal.

A summing power function $W(n)$ can be defined as follows:

$$W(n) = W(n-1) + f_1(n)v_1(n) + \alpha(n-1)v_1(n-1)^2 \tag{5.7}$$

through which (5.4) can be rewritten as:

$$E_{net}(n) = T_s W(n) \tag{5.8}$$

whereas $\alpha(n)$ can be computed as:

$$\alpha(n) = \begin{cases} -\dfrac{W(n)}{v_1^2(n)} & \text{if } W(n) < 0 \\ 0 & \text{if } W(n) \geq 0 \end{cases} \tag{5.9}$$

Substituting value of $\alpha(n)$ from (5.9) for non-trivial case where $W(n) < 0$ into (5.6) yields:

$$f_1'(n) = f_1(n) - \frac{W(n)}{v_1^2(n)}v_1'(n) \tag{5.10}$$

Or:

$$\begin{aligned} f_1'(n) &= f_1(n) - \frac{W(n)}{v_1^2(n)}v_1(n) \\ &= f_1(n) - \frac{W(n)}{v_1(n)} \end{aligned} \tag{5.11}$$

If $W(n)$ is considered as a summation of force and velocity products, as is the case, then this equation shows the dissipated energy in terms of force quantities that are scaled by the ratio of velocity signals.

With sufficiently small T_s, $E_{net}(n)$ closely matches the system energy at instant n. $W(n)$ is evaluated at each step and if found negative, a corrective action is taken. In normal passive operation, $W(n)$ and hence $E_{net}(n)$ should always be positive. In cases when $E_{net}(n) < 0$, passivity observer indicates that the system is generating energy and exhibiting an active behavior. Sometimes, control action may not be required, for example, when this is due to some noise, or for a very brief duration of time.

The controller computed in (5.9) results in a passive system and can be shown easily[28]. Equation (5.4) with $T_s = 1$, for a $1 - port$ can be written as:

$$E_{net}(n) = \sum_{k=1}^{n} f_1'(k)v_1'(k) \tag{5.12}$$

Using (5.5) and (5.6):

$$
\begin{aligned}
E_{net}(n) &= \sum_{k=1}^{n}(f_1(k) + \alpha(k)v_1(k))v_1(k) && (5.13)\\
&= \sum_{k=1}^{n} f_1(k)v_1(k) + \sum_{k=1}^{n-1}\alpha(k)v_1^2(k) + \alpha(n)v_1^2(n) && (5.14)\\
&= W(n) + \alpha(n)v_1(n) && (5.15)\\
&= \geq 0 \quad \text{considering (5.7) and (5.9).} && (5.16)
\end{aligned}
$$

5.3.1 Dissipation Buildup Cancellation

The PC operates only when E_{net} becomes negative. Now for example, in the case of an environment change, where a prior positive energy buildup exists in PO, the stabilizing action of PC will not start operating until all the built up energy has been dissipated and E_{net} dropped to a negative value. One such scenario is a stiff environment in which every contact yields a lot of energy dissipation. In such systems, if this energy can be released after the contact force becomes almost zero or the manipulator has free motion then stable haptic contacts can be made even while operating with highly stiff environments.

Chapter 6

Stabilization in the Presence of Constant Time Delays

This chapter describes the application of TDPC theory, as described in chapter 5 to stabilize bilateral teleoperation in the presence of *constant* time delays. TDPC has been introduced for stabilization of highly stiff haptic interfaces without any communication delay. The contribution of this dissertation is to extend this concept to the stabilization of teleoperation systems *with* time delays. The delays that occur in communication channels can be either constant or time-varying. For applications like fixed orbit satellites or between two fixed points, on a dedicated communication channel, the delays can be generally taken as constant. On the contrary, on a shared channel like internet, and in situations where one of the point is moving with respect to the other one, the delays are of time-varying nature. Even though the delays generated by moving objects are variable, they are of deterministic nature owing to calculated relationship between velocity of the moving object and the time-delay. The delays generated by internet, on the other hand, are essentially non-deterministic because of the random nature of traffic load, queuing, collision, etc. This, generally, is the type found in many of the modern day teleoperated systems. While in this chapter, a treatment based on constant time delays is given, later in chapter 7, stabilization with variable time delays will be discussed.

If the communication network is taken as a component that is to be stabilized, then original TDPC approach, as described by Hannaford and Ryu et. al. [28, 29, 30, 31] can be applied to it, provided there are no delays involved. This requirement is vital because the function of passivity controller is based on accurate measurement of *net energy* by passivity observer or in other words, one needs to compute master and slave energies, in real-time without delay. In the presence of time delays, it becomes impossible:

1. to compute and,
2. to convey the energy information

in a timely fashion, from both sides across the communication channel. This information is used in real-time by the PC design block that provides the time

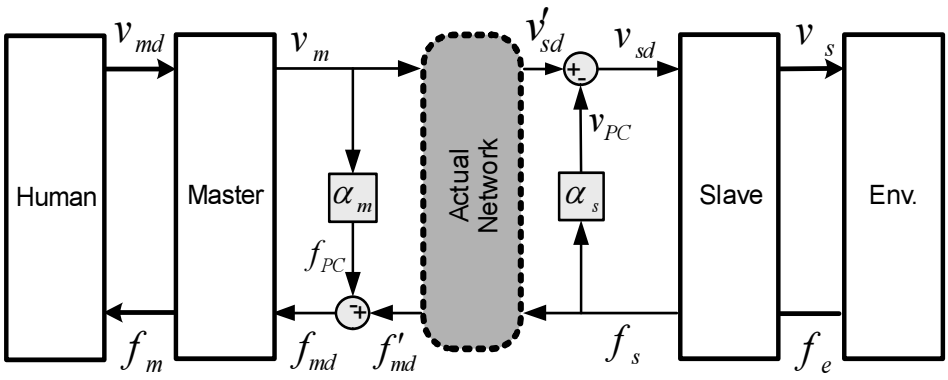

Figure 6.1: Stabilization approach for a network 2-port using TDPC

varying parameters of passivity controllers for both master and slave sides of the communication channel in order to regulate the energy. The PC design block, in our case, is located on master side. If the controller design is carried out on master side, the real-time value of slave energy is unknown because of the delay. In this contribution, it is attempted to solve this problem using a novel energy prediction approach to be used in passivity controller design.

Consider Fig. 6.1, a network $2-port$ acting as a communication link with a transmission delay between master and slave sides. In order to measure energy on slave side, one needs:

$$E_{net}(n) = E_{net_m}(n) - E_{net_s}(n) \tag{6.1}$$

where $E_{net_s}(n)$ and $E_{net_m}(n)$ denote energies at master and slave ports, respectively. Negative sign indicates the direction of energy flow signals out of network port.

Now as we know:

$$E_{net_s}(n) = \sum_{k=1}^{n} f_s(k) v'_{sd}(k) \tag{6.2}$$

and

$$E_{net_m}(n) = \sum_{k=1}^{n} f'_{md}(k) v_m(k) \tag{6.3}$$

So in order to evaluate (6.1) on the master side of Fig. 6.1, i.e., while staying on the left side of the communication channel, requires knowledge of the values of $v'_{sd}(n)$ and $f_s(n)$ in order to have a knowledge of $E_{net_s}(n)$ *in real-time at time instant* t_n. E_{net_m}, however, can be computed readily with the available data. As because of the time delay involved in the communication channel, we do not have timely access to both $v'_{sd}(n)$ and $f_s(n)$, so in order to apply TDPC to stabilize the communication channel, a prediction of slave energy $E_{net_s}(n)$ becomes the

only option to evaluate (6.1). This prediction of slave energy will be discussed in Section 6.2.

6.1 Extension of TDPC to Delayed Teleoperation

In order to stabilize a bilaterally controlled teleoperation system with time delay using TDPC, a regulation of energy flow that includes two passivity controllers α_m and α_s is proposed as shown in Fig. 6.1. Here α_m dissipates surplus energy in the form of force modifications while preserving velocity signals. This is because the velocity signal is an input to the communication channel and modifying it will cause the energy input to the same thus affecting the causality of network port.

Slave PC α_s dissipates negative energy on the slave side of the network $2-port$. Here the force signal, being the actual and un-delayed response of environment, is not modified. The velocity signal, on the other hand, is a delayed signal and *may* have been changed by the induction of negative energy. It is thus subjected to modification by the passivity controller in order to dissipate the negative energy. The rationale behind this scheme is that velocity signal is the delayed signal and can contain the energy induced by the time delay during transmission and so it must be modified in the process of passivation.

As α_m and α_s are time varying parameters depending on the amount of negative energy to be dissipated, so they must be determined in real-time during teleoperation. This design, as it requires the net energy estimate across both ports of communication channel, can be placed on either master or slave side. The controller parameters thus computed can be transfered afterward. In this work, the controller design block is placed on the master side because of the following reasons:

1. The available velocity data $v_m(k) \dots v_m(n)$ can be used along with the time delay information to compute $v'_{sd}(n)$.

2. In most of the situations where instability occurs in teleoperation, it is because of the haptic device (master) or because of human operator, and in both cases it is advantageous to locally correct this behavior using α_m

3. Because the remote location might be inaccessible at times, it is better to place the design block on master side to facilitate any changes in the control algorithm should such a need arise.

The energy prediction will enable us to use the general principles of TDPC as laid out in chapter 5 to be used in the design of α_m and α_s. The TDPC theory can thus be utilized in the stabilization of delayed teleoperation. Once calculated, α_m can be applied locally on the master side while α_s can be sent over the network to the remote side to be used till further update is received.

6.2 Energy Prediction for Time Domain Passivity

Computation of energy on the slave side, i.e., $E_{net_s}(n)$ requires the knowledge of $v'_{sd}(n)$ and $f_s(n)$. Out of these two, $v'_{sd}(n)$ can be determined given knowledge of the forward time-delay and of the forward gains, if any. However $f_s(n)$ requires the actual response of the combined transfer function from $v'_{sd}(n)$ to $f_s(n)$ or a discretized version of $\mathscr{L}[f_s(t)/v'_{sd}(t)]$, see Fig. 6.1.

As the environment and slave conditions may change over time, so an offline identified model is not a promising solution. In this case, one can design a recursive model of the slave-side system which includes both slave and the environment models. The parameters of this model would be estimated online in a recurse manner using a Kalman filter. The force predictor can then, using this estimated model, predict $\hat{f}_s(n)$ using $v_m(n)$ and the delay information. The constant time delay is assumed to be known. Once estimated, $\hat{f}_s(n)$ can be used for computation of α_s and α_m as if no delay existed in the forward channel.

To estimate the slave energy in real-time requires a non-linear recursive estimator of the parameters of slave robot as well as of environment because these parameters can change over time. There can be different solutions to this problem like:

1. Assigning exponential weights to the measurements, thus discounting the weight of previous measurements as the new ones become available, or

2. To use a linear estimator, like Kalman filter, and then to postulate that the true parameter vector is not constant but rather varies like a random walk, etc.

The second approach is selected in this work.

If $\hat{\theta}(t)$ denotes the parameters of online discrete predictor, the recursive parameter update equation can be written as:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n) \cdot \Big[f_s(n) - \hat{f}_s(n) \Big] \tag{6.4}$$

or

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n) \cdot \Big[f_s(n) - \Psi^T(n)\hat{\theta}(n-1) \Big] \tag{6.5}$$

where

$$\hat{\theta}(n) = [\beta_1 \ \beta_2 \ \gamma_1 \ \gamma_2] \tag{6.6}$$

and $\Psi(n)$ is the regressor vector containing delayed input and output values at time t_n.

Hence, the following 2^{nd} order model for prediction of $f(n)$:

$$\hat{f}_s(n) = \gamma_1(n)v'_{sd}(n-1) + \gamma_2(n)v'_{sd}(n-2) - \beta_1(n)f_s(n-1) - \beta_2(n)f_s(n-2) \tag{6.7}$$

Using the Kalman estimator from [49], the Kalman gain $K(n)$ can be computed as:

$$\begin{aligned} K(n) &= Q(n) \cdot \Psi(n) && (6.8)\\ Q(n) &= P(n-1) \cdot \left[\Psi^T \cdot P(n-1) \cdot \Psi(n) + R\right]^{-1} && (6.9)\\ P(n) &= P(n-1) - P(n-1) \cdot \Psi(n) \\ &\quad \cdot \left[\Psi(n)^T \cdot P(n-1) \cdot \Psi(n) + R\right]^{-1} \\ &\quad \cdot \Psi(n)^T \cdot P(n-1) && (6.10) \end{aligned}$$

where $P(n)$ and R are output error- , and measurement error- covariances, respectively.

Because f'_{md} is delayed output of remote system, so the input and out pair that is fed to Kalman filter must be synchronized. At any time instant t_n, the input to Kalman filter is given by:

$$\Psi(n-\zeta_b) = [\hat{v}'_{sd}(n-\zeta_b) \quad f'_{md}(n-\zeta_b)] \quad \zeta_b \geq 0 \tag{6.11}$$

where

$$\hat{v}'_{sd}(n-\zeta_b) = v_m(n-\zeta_f-\zeta_b) \quad \zeta_f, \zeta_b \geq 0 \tag{6.12}$$

which is obtained by backward time-shifting v_m because both the forward and backward time-delays are known. Here ζ_f and ζ_b are defined as:

$$\begin{aligned} \zeta_f &= \lfloor T_f/T_s \rfloor && (6.13)\\ \zeta_b &= \lfloor T_b/T_s \rfloor && (6.14) \end{aligned}$$

where T_f and T_b are delays in forward and backward channels.

The complete stabilization scheme using online estimation and $k-$step ahead energy prediction is shown in Fig. 6.2.

After Kalman filter provides recursively updated estimates of parameters $\hat{\theta}(n)$, the remaining ζ_d inputs can be used in $k-step$ ahead predictor where ζ_d is defined as:

$$\zeta_d = \lfloor (T_b + T_f)/T_s \rfloor \tag{6.15}$$

The k^{th} output corresponds to current time t_n and is given as $\hat{f}_s(n)$ which is the predictive estimate of environment force. Corresponding to equation (5.7), this information can be used as following to estimate the energy:

$$\begin{aligned} W(n) &= W(n-1) \\ &\quad + f_m(n)v_m(n) - \hat{f}_s(n)\hat{v}'_{sd}(n) \\ &\quad + \alpha_m(n-1)v_m(n-1)^2 \\ &\quad + \alpha_s(n-1)\hat{f}_s(n-1)^2 && (6.16) \end{aligned}$$

where

$$\hat{v}'_{sd}(n) = v_m(n-\zeta_f) \tag{6.17}$$

Here $\alpha_m(n-1)v_m(n-1)^2$ denotes the energy dissipated in the last step by PC α_m whereas $\alpha_s(n-1)\hat{f}_s(n-1)^2$ is the energy dissipated by slave PC α_s.

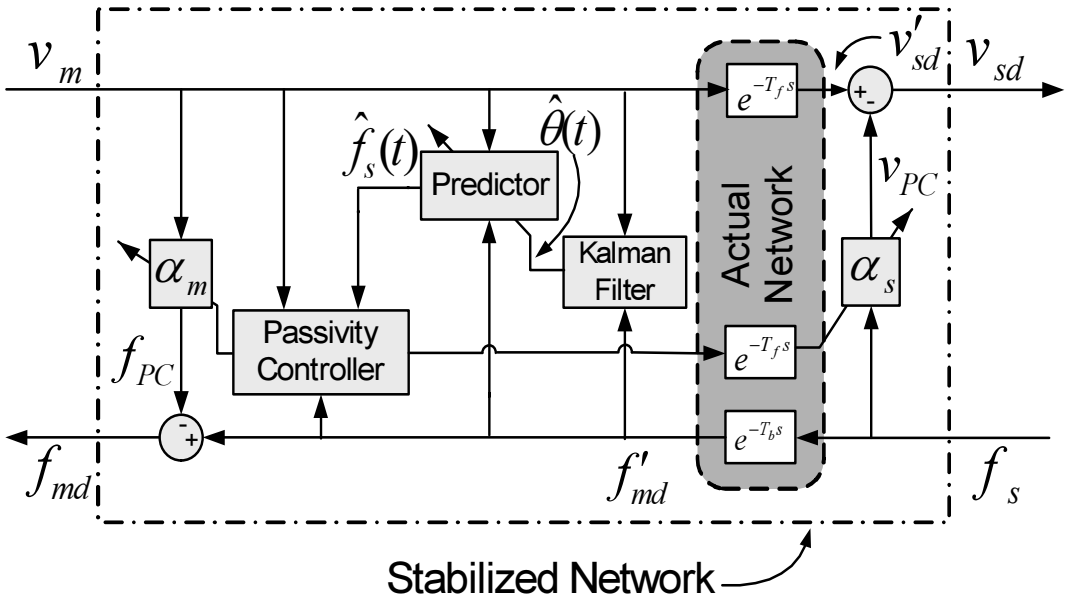

Figure 6.2: Passive network 2-port

Variable	Value
$W(n)$	< 0
$W(n-1)$	≥ 0
$f'_{md}(n)v_m(n)$	> 0
$\hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0

Table 6.1: Conditions for α_s-based energy dissipation, case 1

6.2.1 Controller Design

After $W(n)$ is computed, controllers α_m and α_s can now be calculated in the light of guidelines stated in chapter 5. The calculation of α_m follows the case of impedance causality in which the velocity signal is preserved.

Depending on the state of $W(n)$ and its previous values, there can be different methods to dissipate the accumulated energy through PCs α_m and α_s. Here both E_{net_m} and E_{net_s} must be considered. First let us take the case when all of the conditions in Table 6.1 are true. In this case, it is clear that the net energy is negative as well as that this negative energy is contributed by the slave side of network $2-port$ through $\hat{f}_s(n)\hat{v}'_{sd}(n) < 0$.

So for this case, PC $\alpha_s(n)$ can be computed as:

$$\alpha_s(n) = \frac{-W(n)}{\left[\hat{f}_s(n)\right]^2} \tag{6.18}$$

which dissipates the negative energy on the slave side of network $2-port$.

Variable	Value
$W(n)$	< 0
$W(n-1) + \hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0
$W(n-1) + f'_{md}(n)v_m(n)$	> 0
$f'_{md}(n)v_m(n)$	< 0
$\hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0

Table 6.2: Conditions for α_s-based energy dissipation, case 2

Let us now further consider a case given in Table 6.2. In this case the system is *active* and both the master and slave powers ($f'_{md}(n)v_m(n)$ and $\hat{f}_s(n)\hat{v}'_{sd}(n)$) are negative. When both master and slave ports become negative, the previous energy value $W(n-1)$ is also used, to make, in our case, the slave side additionally conservative. Examining the previous value of power summation function, i.e., $W(n-1)$ indicates that only the slave side was active in the previous sample time. This is testified by the following two conditions:

$$W(n-1) + \hat{f}_s(n-1)\hat{v}'_{sd}(n-1) \quad < 0 \tag{6.19}$$
$$W(n-1) + f'_{md}(n-1)v_m(n-1) \quad > 0 \tag{6.20}$$

In this case, there are two options. Either we dissipate the negative energy on both master and slave sides, or keeping in mind the fact that slave was already active in the previous sample time, discharge all of the accumulated energy on the slave. Here the latter option is selected. So for the case in Table 6.2, master and slave passivity controllers are given as:

$$\alpha_m(n) = 0 \tag{6.21}$$
$$\alpha_s(n) = -\frac{W(n-1) + [P_s(n) - P_m(n)]}{\left[\hat{f}_s(n)\right]^2} \tag{6.22}$$

where $P_m(n)$ and $P_s(n)$ are defined as:

$$P_m(n) = f'_{md}(n)v_m(n) \tag{6.23}$$
$$P_s(n) = \hat{f}_s(n)\hat{v}'_{sd}(n) \tag{6.24}$$

The modified slave velocity command in this case will become:

$$v_{sd}(n) = v'_{sd} + \left(\frac{W(n-1) + [P_s(n) - P_m(n)]}{\left[\hat{f}_s(n)\right]^2} \right) f_s(n) \tag{6.25}$$

Similarly for the case given in Table 6.3, the passivity controllers can be computed as:

$$\alpha_m(n) = -\frac{W(n)}{[v_m(n)]^2} \tag{6.26}$$
$$\alpha_s(n) = 0 \tag{6.27}$$

Variable	Value
$W(n)$	< 0
$W(n-1)$	≥ 0
$f'_{md}(n)v_m(n)$	< 0
$\hat{f}_s(n)\hat{v}'_{sd}(n)$	> 0

Table 6.3: Conditions for α_m-based energy dissipation, case 3

Variable	Value
$W(n)$	< 0
$W(n-1) + \hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0
$W(n-1) + f'_{md}(n)v_m(n)$	< 0
$f'_{md}(n)v_m(n)$	< 0
$\hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0

Table 6.4: Conditions for α_s- and α_m-based energy dissipation, case 4

whereby the modified reflected force coming out of the stabilized network block becomes:

$$f_{md} = f'_{md} + \left(\frac{W(n)}{[v_m(n)]^2}\right) v_m(n) \tag{6.28}$$

Now the case in which both ports have become active is considered and α_m and α_s have to dissipate energy simultaneously. Taking the conditions given in Table 6.4, the passivity controllers are computed as:

$$\alpha_m = -\frac{P_m(n)}{[v_m(n)]^2} \tag{6.29}$$

$$\alpha_s = -\frac{W(n-1) + P_s(n)}{\left[\hat{f}_s(n)\right]^2} \tag{6.30}$$

Here the modified force and velocity signals are given by:

$$f_{md} = f'_{md} + \left(\frac{P_m(n)}{[v_m(n)]^2}\right) v_m(n) \tag{6.31}$$

$$v_{sd}(n) = v'_{sd} + \left(\frac{W(n-1) + P_s(n)}{\left[\hat{f}_s(n)\right]^2}\right) f_s(n) \tag{6.32}$$

Based on the previous energy values, the final option becomes as given in Table 6.5 where the master side of network port is responsible for negative energy.

In this case, clearly, the controller equations are given as:

$$\alpha_m(n) = -\frac{W(n-1) + [P_m(n) - P_s(n)]}{[v_m(n)]^2} \tag{6.33}$$

$$\alpha_s(n) = 0 \tag{6.34}$$

Variable	Value
$W(n)$	< 0
$W(n-1) + \hat{f}_s(n)\hat{v}'_{sd}(n)$	> 0
$W(n-1) + f'_{md}(n)v_m(n)$	< 0
$f'_{md}(n)v_m(n)$	< 0
$\hat{f}_s(n)\hat{v}'_{sd}(n)$	< 0

Table 6.5: Conditions for α_m-based energy dissipation, case 5

and the modified force signal on master side becomes:

$$f_{md} = f'_{md} + \left(\frac{W(n-1) + [P_m(n) - P_s(n)]}{[v_m(n)]^2} \right) v_m(n) \tag{6.35}$$

All of the cases for controller design are summed up in Table 6.6. Cases 4 and 5 have been designed to dissipate the surplus energy symmetrically on both master and slave ports. In the work of Ryu et. al.[29], these two cases are made to dissipate either $-\frac{W(n-1) + P_s(n)}{f_s(n)^2}$ or $-\frac{W(n-1) + P_m(n)}{v_m(n)^2}$, respectively, while in our case, these terms are modified as $-\frac{W(n-1) + (P_s(n) - P_m(n))}{f_s(n)^2}$ and $-\frac{W(n-1) + (P_m(n) - P_s(n))}{v_m(n)^2}$, respectively to make the transmission as lossless as possible. As the calculation of α_s is based on estimated force information, which may contain noise, so the system response becomes quite noisy if α_s is used directly. To counter this problem, a low-pass filter for α_s is suggested which is designed to pass only those frequencies that are closer to the expected correction rate. The concept of filters in teleoperation is not new. Spong et. al.[9] report that a digital implementation of a continuous time passive system may no longer remain passive/stable and would require the introduction of strictly causal and stable linear filters to ensure stability of sampled-data master/slave systems. In addition, simple low-pass filters can also serve to help limit the bandwidth utilization [22].

Case	Energy Variables' Conditions					Designed Controllers	
	$W(n)$	$P_m(n)$	$P_s(n)$	$W(n-1)+P_m(n)$	$W(n-1)+P_s(n)$	$\alpha_m(n)$	$\alpha_s(n)$
1	< 0	< 0	≥ 0	n/a	n/a	$-\frac{W(n)}{[v_m(n)]^2}$	0
2	< 0	≥ 0	< 0	n/a	n/a	0	$-\frac{W(n)}{\left[\hat{f}_s(n)\right]^2}$
3	< 0	< 0	< 0	< 0	< 0	$-\frac{P_m(n)}{[v_m(n)]^2}$	$-\frac{W(n-1)+P_s(n)}{\left[\hat{f}_s(n)\right]^2}$
4	< 0	< 0	< 0	≥ 0	< 0	0	$-\frac{W(n-1)+[P_s(n)-P_m(n)]}{\left[\hat{f}_s(n)\right]^2}$
5	< 0	< 0	< 0	< 0	≥ 0	$-\frac{W(n-1)+[P_m(n)-P_s(n)]}{[v_m(n)]^2}$	0

Table 6.6: Calculation of passivity controllers

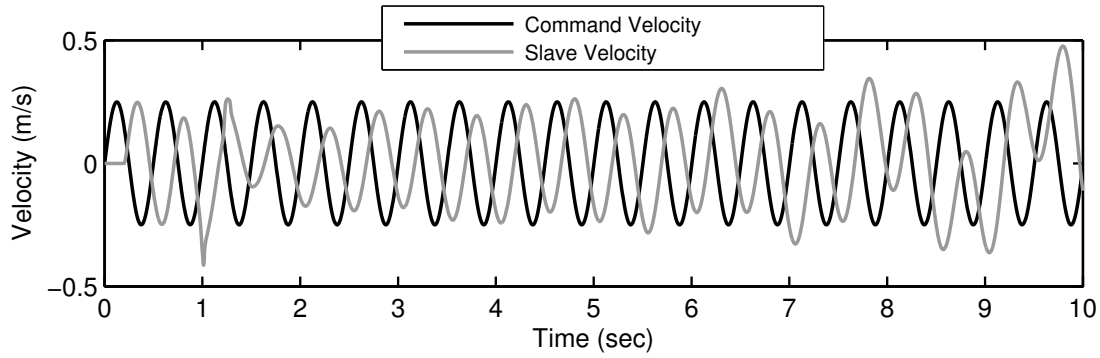

Figure 6.3: Delays $T_f = 0.3s$, $T_b = 0.2s$, command and slave velocities

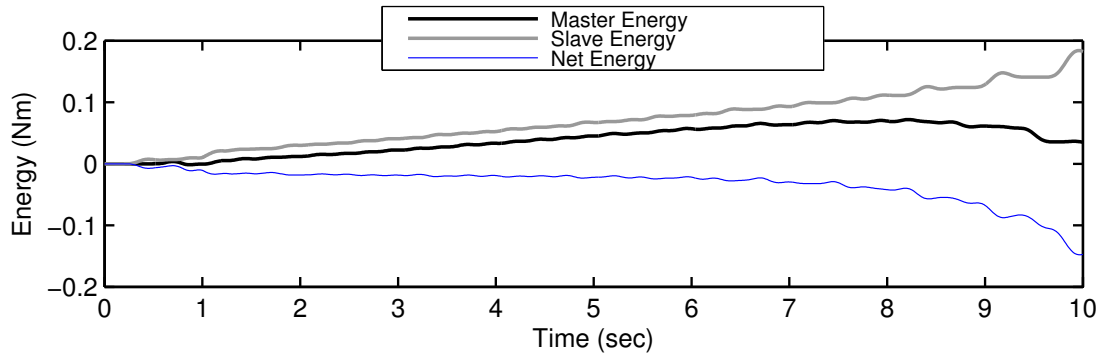

Figure 6.4: Delays $T_f = 0.3s$, $T_b = 0.2s$, energies in continuous time

6.2.2 Simulation Results

In the section the results of the simulations using the teleoperation framework developed in section 2.2 in the presence of constant time delays of $300ms$ in the forward- and $200ms$ in the reverse channel, are provided. The sampling time is fixed at $10ms$ while the command signal frequency is 12.56 rad/s

Fig. 6.3 shows a stable teleoperation system with the help of TDPC as compared to the simulation result given in Fig. 2.19 in the absence of TDPC. The divergence at $t = 1s$ can be explained by the prediction error encountered in the initialization of Kalman filter based predictor as can also be seen in Fig. 6.8 at the same time instant.

Net continuous energy is only about $-0.15Nm$ in Fig. 6.4 compared to $-3000Nm$ without TDPC as shown in Fig. 2.20. Discrete energies on master and slave side of network $2-port$ as well as net discrete energy are shown in Fig. 6.5. This net energy is computed on master side and is used in the computation of master and slave passivity controllers as illustrated in Fig. 6.2.

Stabilizing force and velocity modifications f_{PC} and v_{PC} by PCs α_m and α_s are shown in Figs. 6.6 and 6.7.

Closely matching predicted $\hat{f}_s(n)$ and actual $f_s(n)$ environment forces are given in Fig. 6.8 showing good parameter estimation on part of Kalman filter based estimator of slave and environment model.

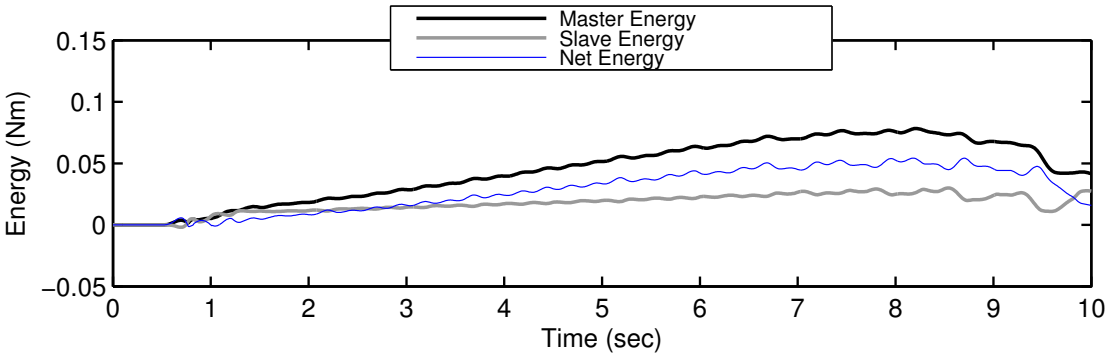

Figure 6.5: Delays $T_f = 0.3s$, $T_b = 0.2s$, discrete energies

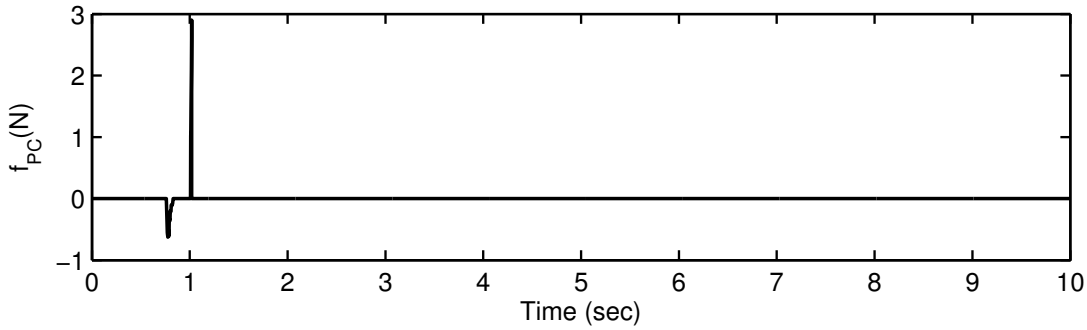

Figure 6.6: Delays $T_f = 0.3s$, $T_b = 0.2s$, f_{PC}

The choice of proper filter for α_s is very important because excessive filtering can leave the whole scheme very conservative while too less or no filtering can lead to noisy outputs. A proper filter should have reasonable correspondence to the frequency of the main control loop. These results are obtained using a 1^{st}-order filter with a cutoff frequency of about $20Hz$ the effects of which are clear in Fig. 6.7.

It should be noted that although the system is stable with the proposed algorithm, certain aspects of it need improvement. For example, although in a stable state, the net discrete energy is still decreasing in Fig. 6.5 which can be improved so that the system always dissipates energy in the sense of GAS. Another issue, that needs to be addressed is the impulsive nature of force and velocity correction indicating accumulation of large sums of negative energy and then dissipating it in big chunks. These issues will be addressed in the following sections.

6.3 Using Energy Derivatives to Enhance Transparency

In order to address the issue of continuously decreasing net discrete energy, one needs to monitor the time varying nature of net discrete energy. As we are inter-

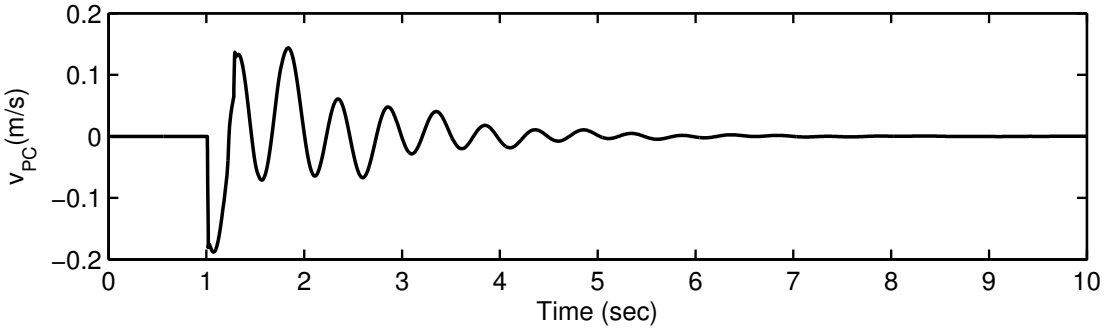

Figure 6.7: Delays $T_f = 0.3s$, $T_b = 0.2s$, v_{PC}

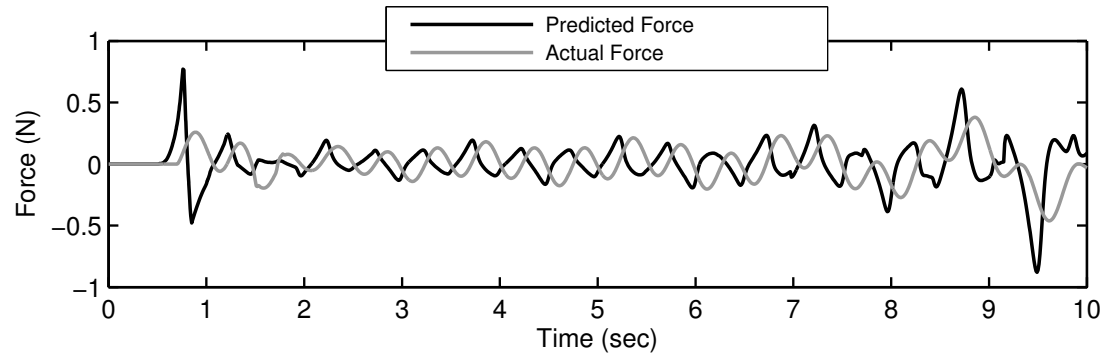

Figure 6.8: Delays $T_f = 0.3s$, $T_b = 0.2s$, predicted and actual environment forces

ested in the time instant where the trend in discrete energy changes to a decrease in net energy, so derivatives of net discrete energy is a suitable variable to monitor the net energy flow. As all the computations of passivity controllers are based on the power summation function $W(n)$, so for all practical reasons, we can use the derivatives of same. We are concerned when the following condition becomes true:

$$\nabla W(n) < 0 \tag{6.36}$$

in which case, we can proceed to dissipate the energy accumulated by the decrease in $W(n)$ thus making it a monotonically increasing function which would render the teleoperation system to be strictly a dissipative one.

6.3.1 Design of Passivity Controllers based on Energy Derivatives

When the condition in (6.36) is satisfied, the following amount of energy can be safely dissipated:

$$-[W(n) - W(n-1)] \tag{6.37}$$

Once computed, we need to decide which side of network port should dissipate this energy. Considering the sensitive nature of the active behavior of slave side

due to delay, it has been decided to dissipate this energy on slave port that results in the following slave passivity controller:

$$\alpha_s(n) = -\frac{[W(n) - W(n-1)]}{\left[\hat{f}_s(n)\right]^2} \tag{6.38}$$

The modified velocity signal becomes:

$$v_{sd}(n) = v'_{sd}(n) + \frac{[W(n) - W(n-1)]}{\left[\hat{f}_s(n)\right]^2} \tag{6.39}$$

To implement this feature, we keep track of the net energy $W(n)$ and its derivative $\left.\frac{dW}{dt}\right|_{t=t_n}$. Once *the derivative* gets negative, we do not wait for *net energy* to become negative. Rather this decrease in net energy is compensated immediately using (6.38). This approach gives another meaning to the *Reference Energy Following* approach described in [32].

There can, however, be instances depending on the type of haptic devices involved and the difference between forward backward delays, where the dissipation of this energy can be distributed on both sides or even dissipated on the master side of network port. We can see that energy derivative control comes into action only when net energy is positive, so the calculations of α_m and α_s are not affected in the case of net negative energy. The use of energy derivatives can, however, result in tracking errors because of additional damping. This effect can be compensated by appropriately scaling the input command. The resulting controllers that include the derivative based energy regulation are given in Table 6.7.

It will be shown in the simulations that the addition of energy derivative information in passivity controller design reduces the impulses in the velocity modifications by α_s that results in smooth teleoperation and better velocity tracking.

6.3.2 Placement of Kalman Filter based Parameter Estimator

In the previously proposed scheme, as shown in Fig. 6.2, the Kalman filter based parameter estimator is placed on master side of network $2-port$. However, during simulations, it was observed that estimating the slave and environment models in such a fashion across the network is not the optimal solution. The reasons include:

1. In such a scheme, Kalman estimator is exposed to the velocity modifications by slave passivity controller which is not a desirable effect.

2. On remote side, we can monitor the signals just before they enter the slave arm and from the force sensor when the come out whereas in the previously proposed approach, these signals are subject to deterioration by network delays and losses in addition to errors introduced by quantization.

Case	Energy Variables' Conditions						Designed Controllers	
	$W(n)$	$\left.\frac{dW}{dt}\right\|_{t=t_n}$	$P_m(n)$	$P_s(n)$	$W(n-1)+P_m(n)$	$W(n-1)+P_s(n)$	$\alpha_m(n)$	$\alpha_s(n)$
1	<0	n/a	<0	≥ 0	n/a	n/a	$-\frac{W(n)}{[v_m(n)]^2}$	0
2	<0	n/a	≥ 0	<0	n/a	n/a	0	$-\frac{W(n)}{\left[\hat{f}_s(n)\right]^2}$
3	<0	n/a	<0	<0	<0	<0	$-\frac{P_m(n)}{[v_m(n)]^2}$	$-\frac{W(n-1)+P_s(n)}{\left[\hat{f}_s(n)\right]^2}$
4	<0	n/a	<0	<0	≥ 0	<0	0	$-\frac{W(n-1)+(P_s(n)-P_m(n))}{\left[\hat{f}_s(n)\right]^2}$
5	<0	n/a	<0	<0	<0	≥ 0	$-\frac{W(n-1)+(P_m(n)-P_s(n))}{v_m(n)^2}$	0
6	≥ 0	<0	≥ 0	≥ 0	n/a	n/a	0	$-\frac{[W(n)-W(n-1)]}{\left[\hat{f}(n)\right]^2}$

Table 6.7: Calculation of passivity controllers using energy derivatives

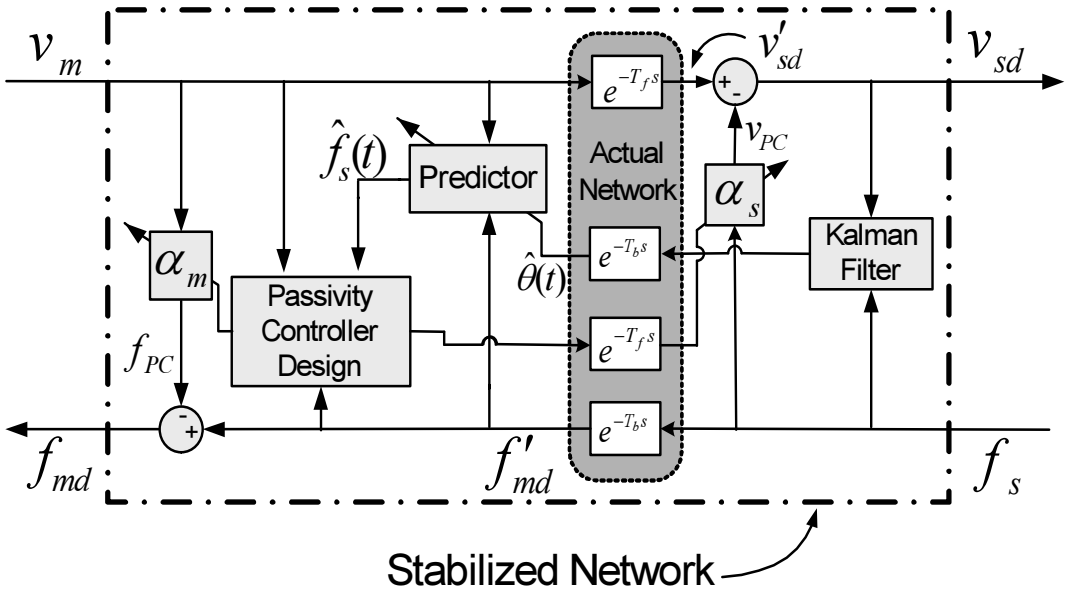

Figure 6.9: Modified passive network 2-port with Kalman filter on slave side

3. In the case of a sudden parameter change, the model is not updated until certain delay because of the communication channel.

Keeping these factors in view, the Kalman estimator is moved to the remote side and the parameters are sent over the network on every data send. These are then used locally on the master side in the prediction of remote force. This move also enables us to decouple the model update rate from the main control loop frequency as it can now be increased to even $1ms$ or beyond. Other than the benefits explained above, fast updating of this model on the remote side can be used in different control enhancements on remote side, including supervisory and model based control techniques.

The improved stabilization scheme is shown in Fig. 6.9.

6.3.3 Simulations using Predictive Time Domain Passivity with Energy Derivatives

Following are the results when the enhanced TDPC with energy derivatives is used to stabilize the given teleoperation system. Here the same system as in section 6.2.2 is simulated with the similar parameters, i.e., $300ms$ delay in the forward- and $200ms$ in the reverse channel. Very significant improvement in velocity tracking can be observed in Fig. 6.17 which shows the velocity results of both control schemes.

The net continuous energy with the use of energy derivatives in Fig. 6.11 has now dropped to about $-0.015Nm$ compared to $-0.15Nm$ in Fig. 6.4 with energy derivatives. This is a 10 fold improvement in passivity of the system based on

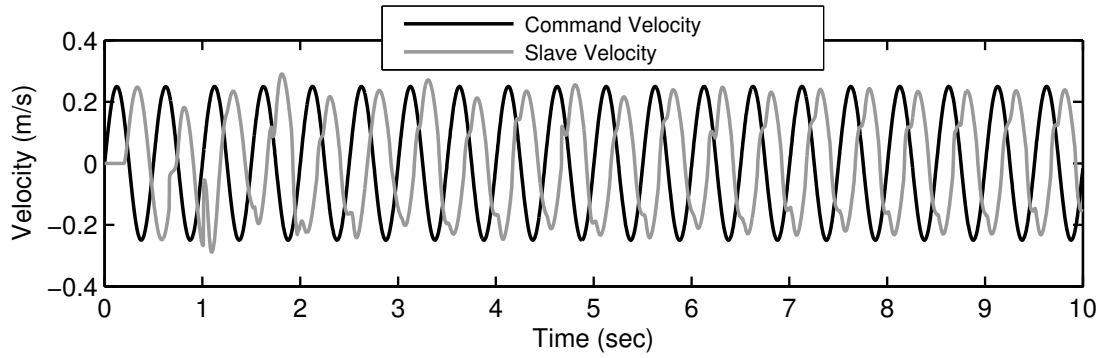

Figure 6.10: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, command and slave velocities

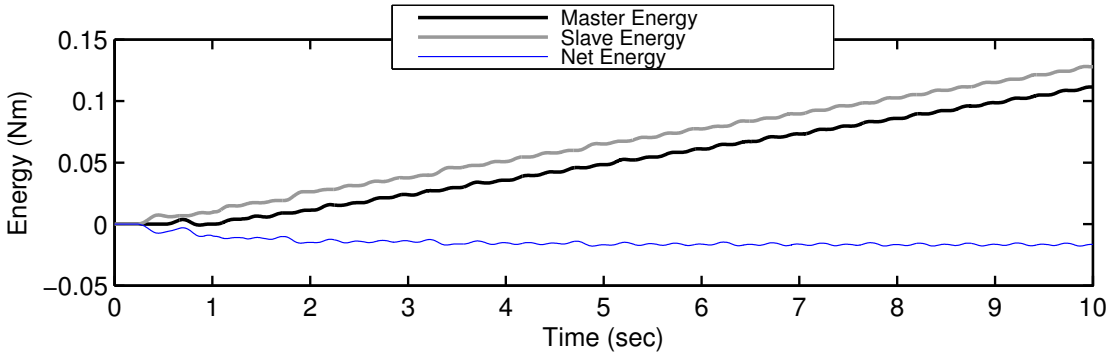

Figure 6.11: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, energies in continuous time

net continuous energies. Fig. 6.12 shows the discrete energies across network $2 - port$. We can see that the net discrete energy is always increasing indicting strict dissipitivity and hence passivity.

The velocity corrections v_{PC} by α_s are also less of impulsive nature and more gradual as shown in Fig. 6.14. This is due to continuous dissipation by energy derivatives based algorithm.

6.3.4 Comparing the Simulation Results with- and without-Energy Derivatives

The comparison between the results of control scheme using Energy derivatives and without using energy derivatives is given in the Figs. 6.16-6.19. All of these figures are worth careful study and highlight the difference between both methods as well as the gains of energy derivatives. Fig. 6.18, for example, shows that net continuous energy is almost zero compared to slowly decreasing net continuous energy in the case of control scheme without energy derivatives.

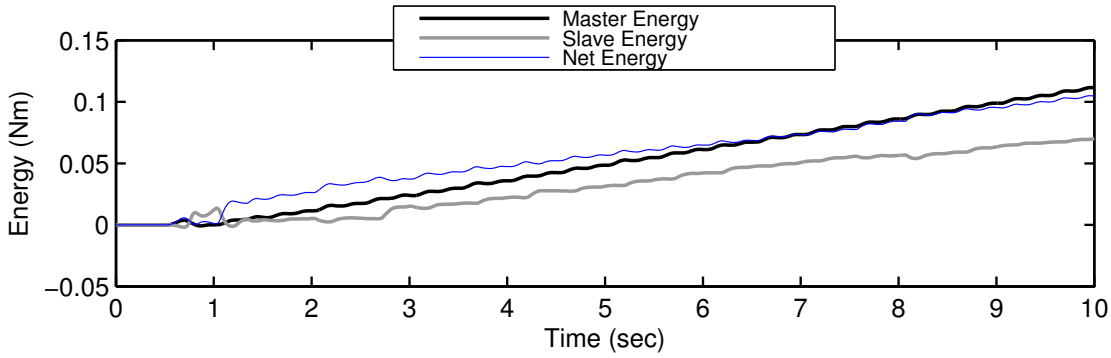

Figure 6.12: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, discrete energies

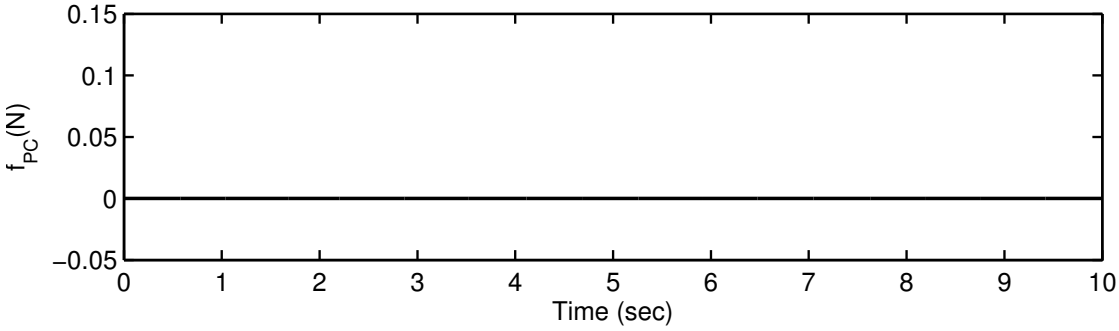

Figure 6.13: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, f_{PC}

6.4 Non-linear Recursive Estimation of Net Energy using Parabolic Power Integration

In all of the TDPC techniques described in the previous sections, computation of net energy is carried out using rectangular integration as shown in Fig. 6.20.

It can be seen that such an integration is not very accurate and may cause extra energy to be dissipated resulting in poor tracking performance and loss of transparency. On the other hand, errors in energy estimation can also lead to less energy being dissipated than is required thus resulting in unstable teleoperation. It becomes thus of practical interest to find alternate ways of energy computation that enhance the accuracy of power integration. It is proposed that Simpson's rule based parabolic power integration (PPI) should be used instead that will enable us to non-linearly integrate the power and would provide better estimates of net energy.

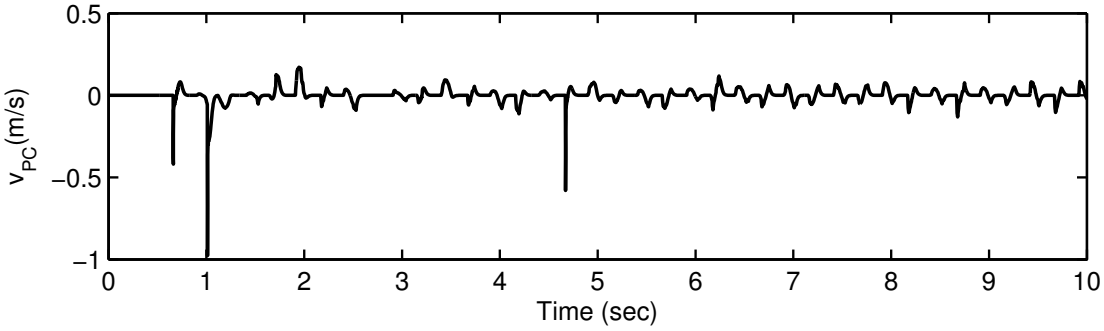

Figure 6.14: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, v_{PC}

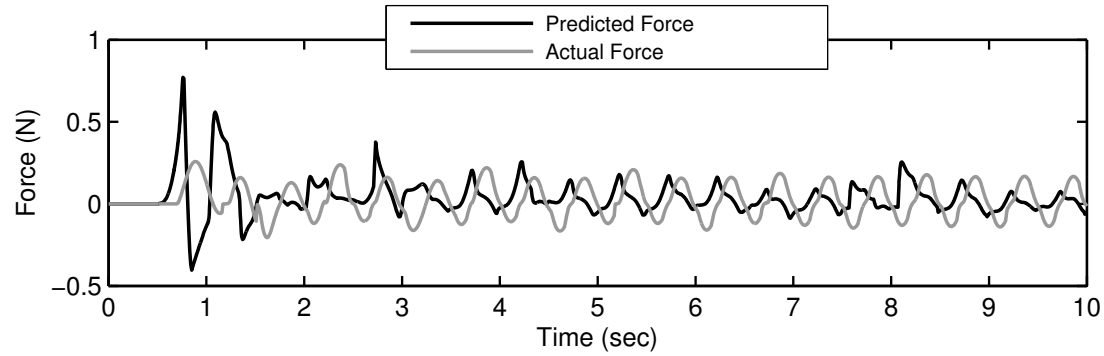

Figure 6.15: **With energy derivatives**, Delays $T_f = 0.3s$, $T_b = 0.2s$, predicted and actual environment forces

6.4.1 Design of Passivity Controllers based on Parabolic Power Integration

In order to integrate the net power parabolically, we first need to evaluate the following expressions in (6.40) and (6.43).

$$P_a(n) = P_m(n) - P_s(n) \tag{6.40}$$

Here $P_a(n)$ is the instantaneous net power at the given network $2-port$ representing communication channel without taking into consideration the energy dissipation by passivity controllers, whereas $P_m(n)$ and $P_s(n)$ represent the power at the master and slave ports and are given as:

$$P_m(n) = f'_{md}(n)v_m(n) \tag{6.41}$$
$$P_s(n) = \hat{f}_s(n)\hat{v}'_{sd}(n) \tag{6.42}$$

$$P_d(n) = -\alpha_m(n-1)v_m(n-1)^2 - \alpha_s(n-1)f_s(n-1)^2 \tag{6.43}$$

where $P_d(n)$ is the instantaneous power being dissipated by master and slave passivity controllers α_m and α_s in the previous sample time.

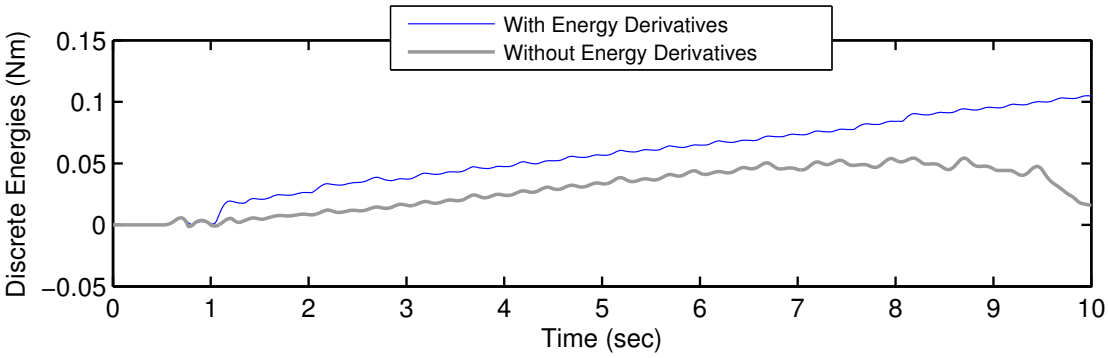

Figure 6.16: Comparison of net discrete energies with and without energy derivatives

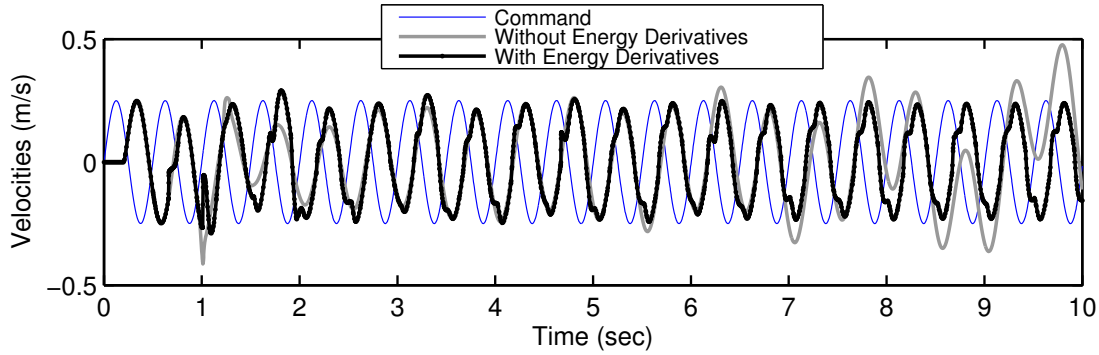

Figure 6.17: Comparison of slave velocities with and without energy derivatives

The following then describes the net energy that is to be computed recursively:

$$E_{net}(n) = \sum_{i=0}^{n} \left[\int_{t_{i-1}}^{t_i} P_a(\tau) d\tau - \int_{t_{i-2}}^{ti-1} P_d(\tau) d\tau \right] \tag{6.44}$$

$$where \qquad P_d, P_a = 0 \; \forall \; t \leq 0$$

As described earlier, $\hat{f}_s(n)$ is estimated based on the Kalman filter based predictor whereas $\hat{v}'_{sd}(n)$ is computed as:

$$\hat{v}'_{sd}(n) = v_m(n - \zeta_f) \tag{6.45}$$

and

$$\zeta_f = \lfloor T_f / T_s \rfloor \tag{6.46}$$

Equation (6.43) can be rewritten as:

$$E_{net}(n) = \sum_{i=0}^{n} \left[\int_{t_{i-1}}^{t_i} P_a(\tau) d\tau \right] - \sum_{i=0}^{n-1} \left[\int_{t_{i-1}}^{ti} P_d(\tau) d\tau \right] \tag{6.47}$$

The first term in this expression is the actual energy flow through the network $2-port$ without taking into consideration the passivity controllers. The integral

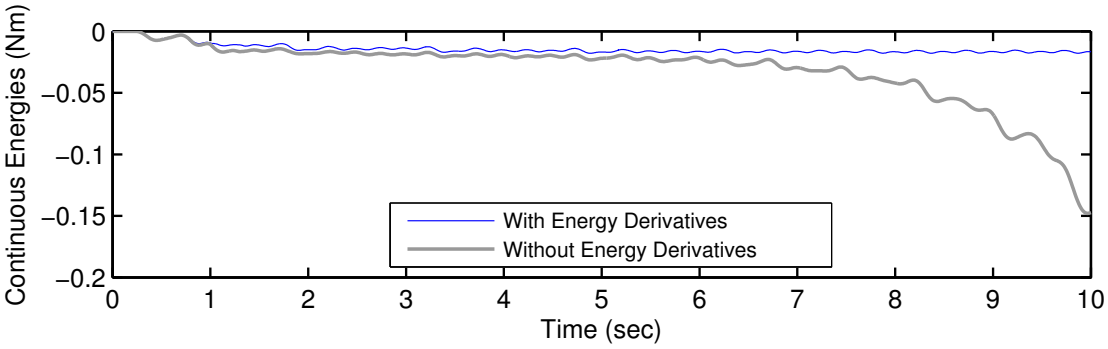

Figure 6.18: Comparison of net continuous energies with and without energy derivatives

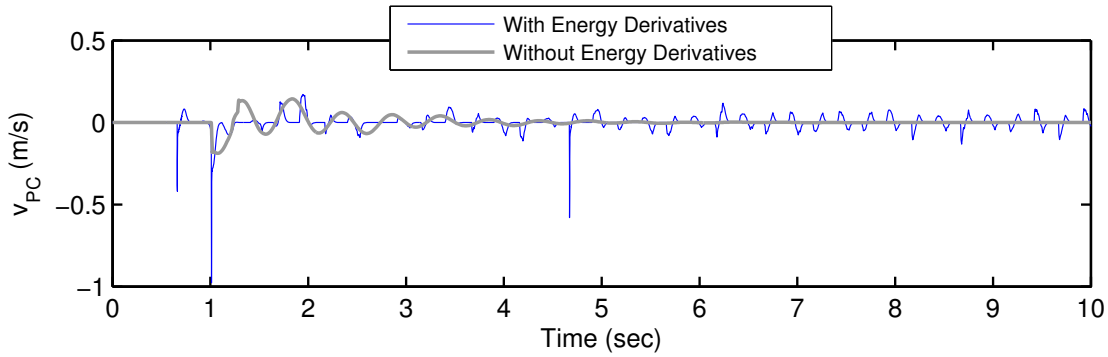

Figure 6.19: Comparison of v_{PC}'s with and without energy derivatives

has to be evaluated at each step in a parabolic fashion to provide accurate energy estimates. The second term indicates the net dissipated energy till $(n-1)^{th}$ step through both master and slave passivity controllers.

As Simpson's rule for discrete integration of energies is being used, both P_a and P_d are assumed to follow a parabolic function in the last three samples. The integral of a fitted parabola yielding observed energy for P_a from time t_{n-1} to t_n is given as:

$$\begin{aligned}\int_{t_{n-1}}^{t_n} P_a(\tau)dt &= \frac{a(t_n)}{3}(t_n^3 - t_{n-1}^3) + \frac{b(t_n)}{2}(t_n^2 - t_{n-1}^2) \\ &\quad + c(t_n)(t_n - t_{n-1}) \end{aligned} \tag{6.48}$$

where $a(t_n), b(t_n)$ and $c(t_n)$ are the time varying parameters of a parabola:

$$P_a(n) = a(t_n)t^2 + b(t_n)t + c(t_n) \tag{6.49}$$

that has been fitted through three points namely $P_a(t_{n-2})$, $P_a(t_{n-1})$ and $P_a(t_n)$. A pictorial description of this recursive parabolic energy computation is shown in Fig. 6.21.

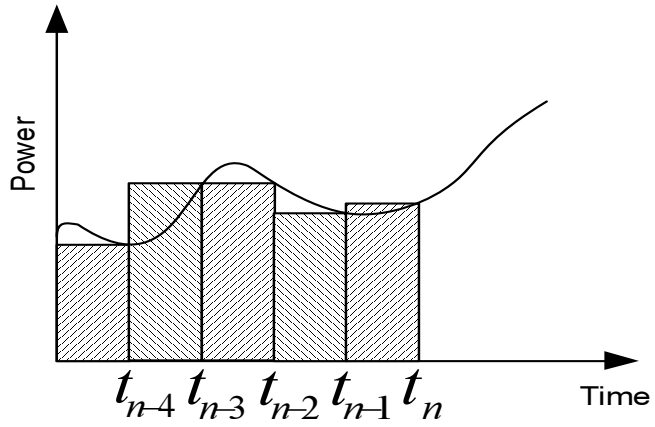

Figure 6.20: Rectangular integration of discrete energy

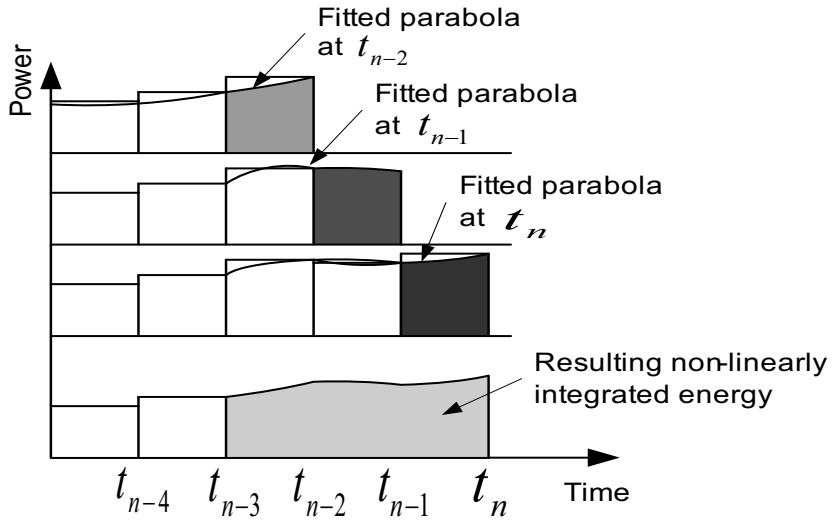

Figure 6.21: Recursive Parabolic Power Integration

The last term in (6.44), i.e., $\int_{t_{i-2}}^{t_{i-1}} P_d(\tau)d\tau$ can be evaluated in a similar fashion. This change in power integration also changes the previous derivations of passivity controllers given in Table 6.7. Because now $W(n)$ can not be used as a replacement of energy function, so we must use $E_{net}(n)$ instead in the computation of passivity controllers.

Let us take the previous case given in Table 6.2 where the slave side is responsible for the active behavior of network $2-port$. In the case of PPI, the updated conditions now look as shown in Table 6.8.

For this case without PPI, as the energy is to be dissipated on slave side, the passivity controller α_s given in (6.22), because of no access to the value of $W(n)$ in the case of PPI, now changes to:

Variable	Value
$E_{net}(n)$	< 0
$E_{net}(n-1) + P_s(n)T_s$	< 0
$E_{net}(n-1) + P_m(n)T_s$	> 0
$P_m(n)$	< 0
$P_s(n)$	< 0

Table 6.8: Updated conditions for α_s-based energy dissipation using PPI

$$\alpha_s(n) = -\frac{\left(\frac{E_{net}(n-1)}{T_s}\right) + [P_s(n) - P_m(n)]}{\left[\hat{f}_s(n)\right]^2} \tag{6.50}$$

Equations (6.22) and (6.50), considering (5.4), differ in the way P_s, P_m, and E_{net} are obtained which are non-linearly integrated in the case of PPI. The revised expressions for all the different cases of passivity controllers using PPI are given in Table 6.9.

6.4.2 Using 3^{rd}-Order Model for Slave and Environment

Simulations suggest that a 3^{rd}-order model best describes the dynamics of the system as well as smoothens the predicted energies considerably. In this case, if $\hat{\theta}(n)$ denotes the parameters of online predictor, the recursive parameter update equation can be written as:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n) \cdot \left[f_s(n) - \hat{f}_s(n)\right] \tag{6.51}$$

or

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n) \cdot \left[f_s(n) - \Psi^T(n)\hat{\theta}(n-1)\right] \tag{6.52}$$

where

$$\hat{\theta}(n) = [\beta_1(n) \ldots \beta_3(n) \quad \gamma_1(n) \ldots \gamma_3(n)] \tag{6.53}$$

are the time varying parameter of the following 3rd order model for prediction of $f_s(n)$:

$$\hat{f}_s(n) = -\sum_{i=1}^{3} (\beta_i(n) f_s(n-i)) + \sum_{i=1}^{3} (\gamma_i(n) v_{sd}(n-i)) + e(n) \tag{6.54}$$

The parameter estimation model can be described as:

$$\hat{\theta}(n+1) = G(n)\hat{\theta}(n) + w(n) \tag{6.55}$$

$$\hat{f}_s(n) = H(n)\hat{\theta(n)} + e(n) \tag{6.56}$$

where in our case, $G(n) = I$, and $H(n) = \Psi^T(n)$. $\hat{\theta}(n)$ is the parameter vector at time t_n, and $\Psi(n)$ is the regressor vector. w is white Gaussian noise and serves to

Case	Energy Variables' Conditions						Designed Controllers	
	$E_{net}(n)$	$\left.\frac{dE_{net}(t)}{dt}\right\|_{t=t_n}$	$P_m(n)$	$P_s(n)$	$E_{net}(n-1)+(P_m(n)T_s)$	$E_{net}(n-1)+(P_s(n)T_s)$	$\alpha_m(n)$	$\alpha_s(n)$
1	< 0	n/a	< 0	≥ 0	n/a	n/a	$-\frac{(E_{net}(n)/T_s)}{[v_m(n)]^2}$	0
2	< 0	n/a	≥ 0	< 0	n/a	n/a	0	$-\frac{(E_{net}(n)/T_s)}{\left[\hat{f}_s(n)\right]^2}$
3	< 0	n/a	< 0	< 0	< 0	< 0	$-\frac{P_m(n)}{[v_m(n)]^2}$	$-\frac{(E_{net}(n-1)/T_s)+P_s(n)}{\left[\hat{f}_s(n)\right]^2}$
4	< 0	n/a	< 0	< 0	≥ 0	< 0	0	$-\frac{(E_{net}(n-1)/T_s)+(P_s(n)-P_m(n))}{\left[\hat{f}_s(n)\right]^2}$
5	< 0	n/a	< 0	< 0	< 0	≥ 0	$-\frac{(E_{net}(n-1)/T_s)+(P_m(n)-P_s(n))}{[v_m(n)]^2}$	0
6	≥ 0	< 0	≥ 0	≥ 0	n/a	n/a	0	$-\frac{\left.\frac{dE_{net}(t)}{dt}\right\|_{t=t_n}}{\left[\hat{f}_s(n)\right]^2}$

Table 6.9: Calculation of passivity controllers using parabolic power integration

help drift the parameter vector in the manner of a random walk. v is white noise. The above equations can be rewritten as:

$$\hat{\theta}(n+1) = \hat{\theta(n)} + w(n) \tag{6.57}$$
$$\hat{f}_s(n) = \Psi^T(n)\hat{\theta}(n) + e(n) \tag{6.58}$$

where

$$\mathbb{E}\left[w(n)w^T(n)\right] = R_1(n) \tag{6.59}$$
$$\mathbb{E}\left[e^2\right] = R_2(n) \tag{6.60}$$

In such a case, equation (6.51) can be computed as[50]:

$$K(n) = P(n-1)\cdot\Psi(n)\cdot\left[\Psi^T\cdot P(n-1)\cdot\Psi(n) + R_2(n)\right]^{-1} \tag{6.61}$$
$$\begin{aligned} P(n) = {} & P(n-1) - P(n-1)\cdot\Psi(n) \\ & \cdot\left[\Psi^T(n)\cdot P(n-1)\cdot\Psi(n) + R_2(n)\right]^{-1} \\ & \cdot\Psi^T(n)\cdot P(n-1) + R_1(n) \end{aligned} \tag{6.62}$$

where $P(n)$ and R are output error- , and measurement error-covariances, respectively. $\Psi(n)$ is the vector containing input and output values at time t_n. The input and output pair that is fed to Kalman filter is synchronized taking into consideration the time delays involved as described before.

6.4.3 Simulation Results using Parabolic Power Integration

A comparison of the stabilization results using PPI and without PPI is given in Figs. 6.22-6.27. Fig. 6.22 shows the continuous energies in the system while Fig. 6.23 shows discrete energies in the system. The discrete energies, on which the passivity controller design works, show an increase in the dissipitivity of the system contributing to stability. On the other hand, a slight decrease in the net continuous energy in the case of PPI, as shown in Fig 6.22, indicates a decrease in system damping leading to a more responsive system as depicted in Figs. 6.26 and 6.27. One must understand that a decrease in *net discrete energy* is the immediate result of PPI because *net discrete energy* is the variable that we are directly manipulating through passivity controllers. The *net continuous energy*, contrarily, is a system performance indicator which is not as directly coupled to changes in passivity controllers as *net discrete energy* is.

The passivity controller outputs are shown in Figs. 6.24 and 6.25. Considering these velocity modifications from α_s, it becomes clear that not only the system is more dissipative in the case of PPI, but also the energy regulation is more smooth and more distributed as can be seen in Fig. 6.25. This shows that the improvements in the estimation of net energy have enabled the designed controllers to dissipate energy more regularly thus avoiding spontaneous large withdrawals of energy.

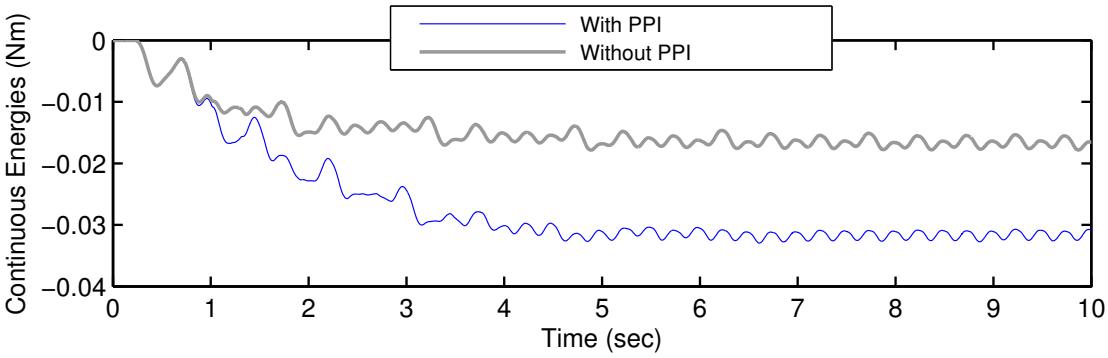

Figure 6.22: Comparison of net continuous energies with and without PPI

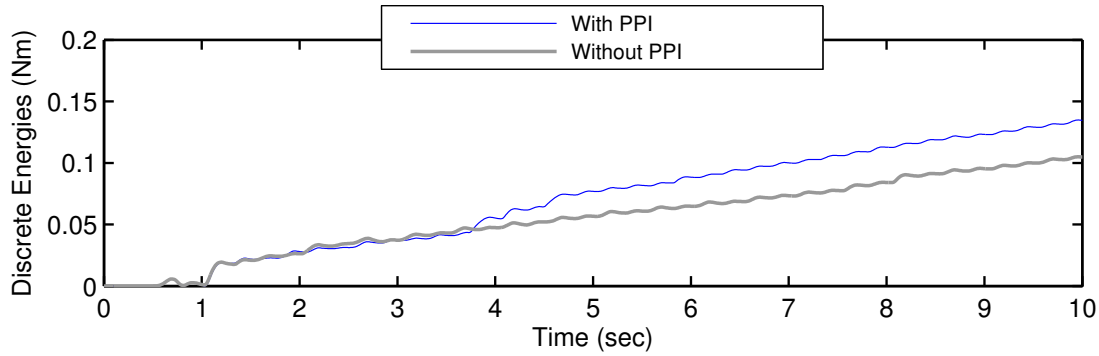

Figure 6.23: Comparison of net discrete energies with and without PPI

Fig. 6.26 shows the comparisons of slave velocity tracking as well as stability. In order to better visualize the gains by the use of PPI in terms of better tracking of velocity signals, a magnified section of the same is shown in Fig. 6.27. This shows a 10% improvement in velocity tracking in the case of PPI than without it.

6.5 Time Varying Environment

As rather than using an offline identified model of slave side components of network port, a Kalman filter based estimator of slave robot and environment models is used, so the proposed stabilization scheme can also be used in such cases where the stiffness or damping parameters of the environment or of slave are time varying. Fig. 6.28 shows the time-varying environment when its stiffness is changed from 0 to 10 N/m while the damping stays at 0.5 N.s/m.

There can be a concern to possible outcomes of stabilization when the changes in the environment parameters are rapid and because of time delay, the updated model information is not conveyed in-time. This and other related issues can be managed by imposing a bound on the energy that can be induced in the network $2-port$ in a given duration of time. This limit can be parameterized based on the time delay, slave velocity, as well as the operator impedance.

If $E_a(n)$ denotes the net energy that has entered the network $2-port$ port till

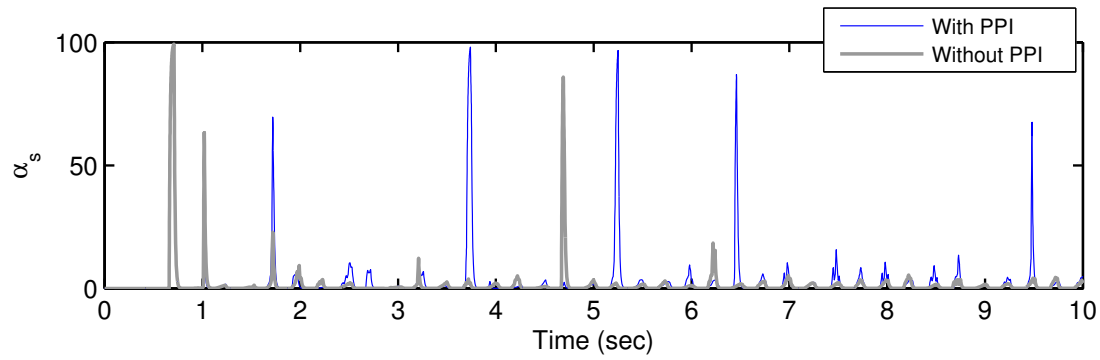

Figure 6.24: Comparison of α_s with and without PPI

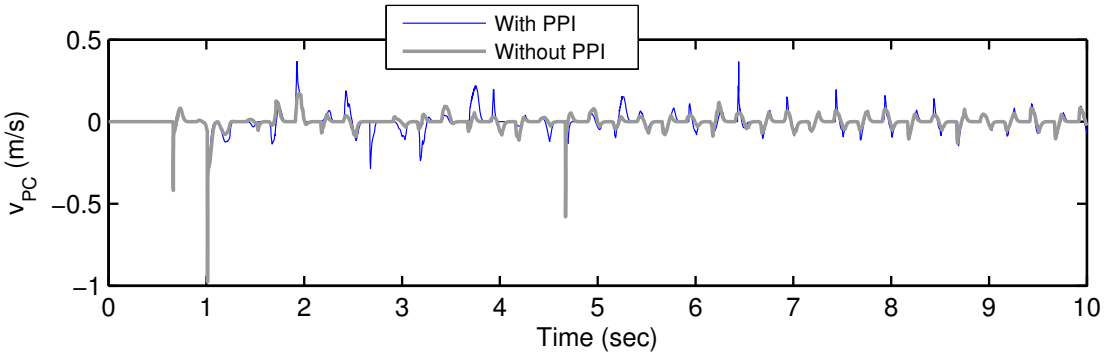

Figure 6.25: Comparison of v_{PC} with and without PPI

time t_n, then:

$$\Lambda(T_f, T_b, v_s, Z_{op}, \dots) \geq E_a(n) - E_a(n-k) \tag{6.63}$$

can be defined to limit the input energy during $(t_n - t_{n-k})$ in order to provide room for the abrupt changes in slave side dynamics.

6.5.1 Stabilization with Variable Environment

In this section, the simulation results when the parameters of the environment are changing with respect to time, are presented. The results are obtained with the following settings:

1. When the stiffness is time-varying
2. When the damping is time-varying
3. When both stiffness and damping are varying with respect to time

Variable Stiffness

Firstly, the stiffness is varied from 0 to 10 N/m and the results are given in Figs. 6.29 to 6.33. Fig. 6.29 shows the command and slave velocities when the gain K

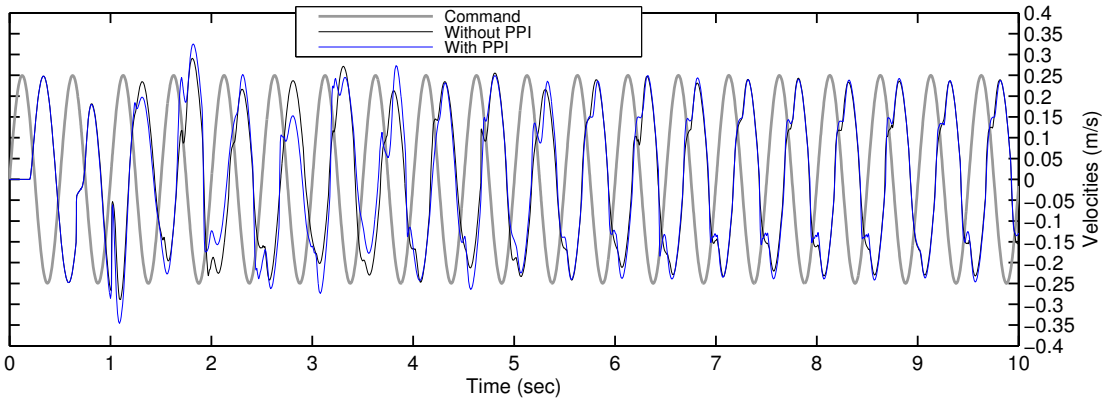

Figure 6.26: Comparison of slave velocities with and without PPI

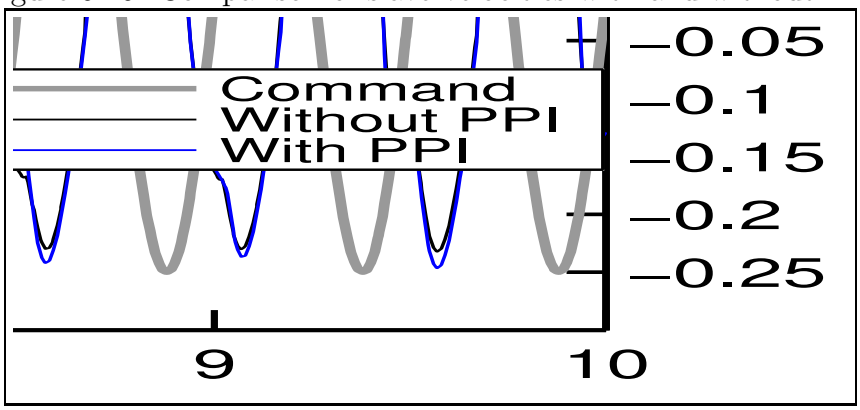

Figure 6.27: Magnified view showing better tracking in the case of PPI

is varying as seen in Fig. 6.30. It can be seen that the stabilization approach can reasonably manage the parameter change till the stiffness becomes 10 N/m. The energy dissipation, nevertheless, has to be increased as the reflected energy from the increased stiffness environment is increasing. This fact is visible in Fig. 6.33. Although the system is stable, but these corrective impulses degrade the tracking and hence transparency. If the high stiffness is continued, the modification from the passivity controllers will attain higher peaks. To retain tracking and transparency, either the stiffness has to be reduced or the loop delay has to be minimized. Reducing the delay will help, because it will facilitate higher dissipation frequency, thus reducing the chunks of energy at each sampling time. This fact can be stated as following:

$$\Delta E_d \propto T_d K \tag{6.64}$$

where ΔE_d is the energy dissipated in a sampling interval, T_d is the loop time delay, and K is environment stiffness.

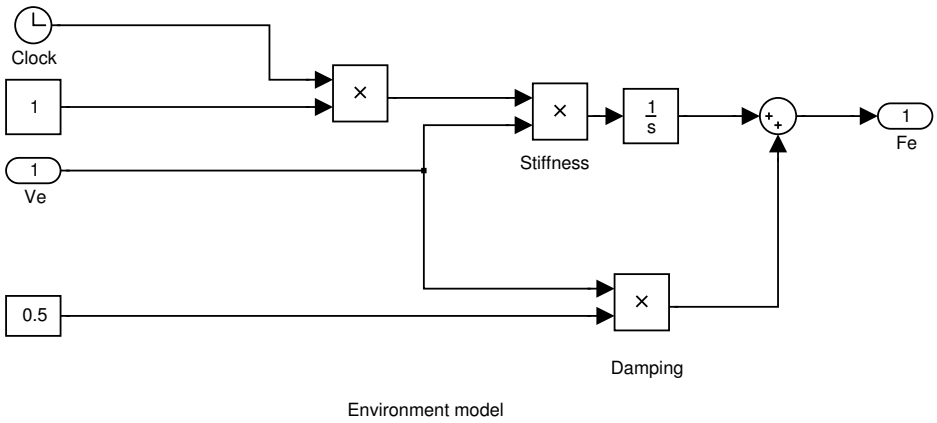

Figure 6.28: Variable Stiffness Environment

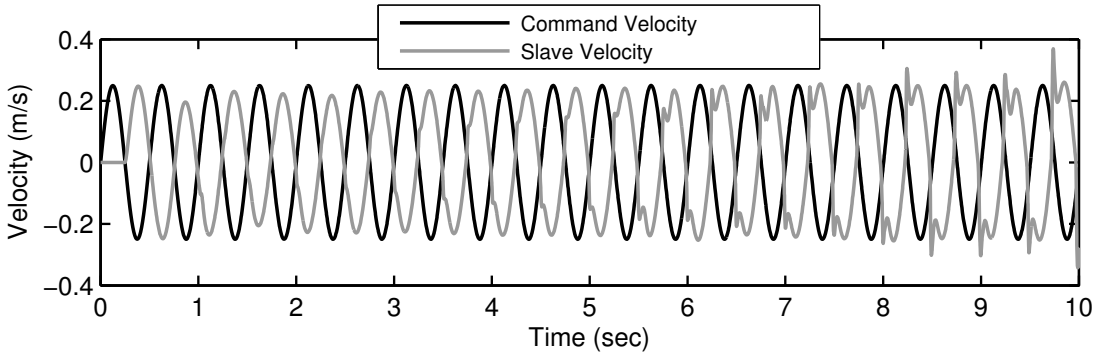

Figure 6.29: **Variable stiffness**, Delays $T_f = 0.25s$, $T_b = 0.25s$, command and slave velocities

Variable Damping

Alternatively, instead of variable stiffness, some environments can change the damping parameter during teleoperation. Here the simulation results for an environment where the stiffness K is fixed at 5 N/m are produced. The damping co-efficient B is varied from 0 to 1 N.s/m. Simulation results are shown in Figs. 6.34-6.38.

The velocity tracking is given in Fig. 6.34. The response of the system is a bit noisy in the beginning which quite understandably settles down as the damping increases. Fig. 6.38 shows decreasing energy dissipations with increasing damping. Extreme values of damping co-efficients can lead to position errors as the force response will decrease following (2.30). In such a case, improvement in tracking performance may require the addition of a position control loop on the slave arm.

Variable Stiffness and Damping

When both the stiffness constant K and the damping B are varied simultaneously, the results are given in Figs. 6.39-6.43. Here the stiffness is varied from 1 to

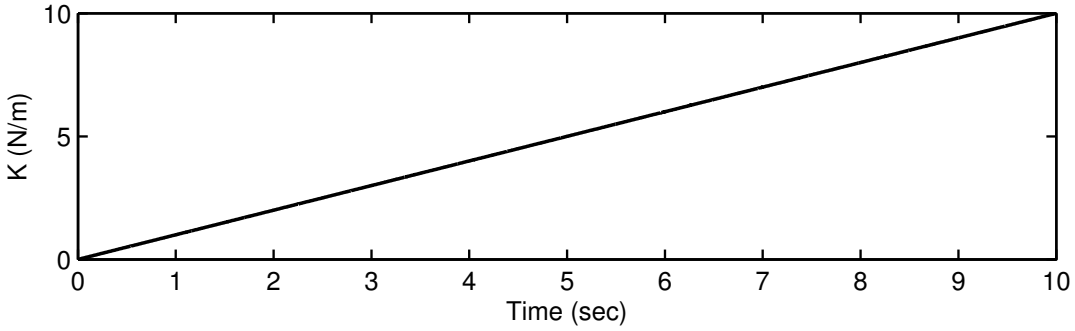

Figure 6.30: **Variable stiffness**, Delays $T_f = 0.25s$, $T_b = 0.25s$, stiffness constant K

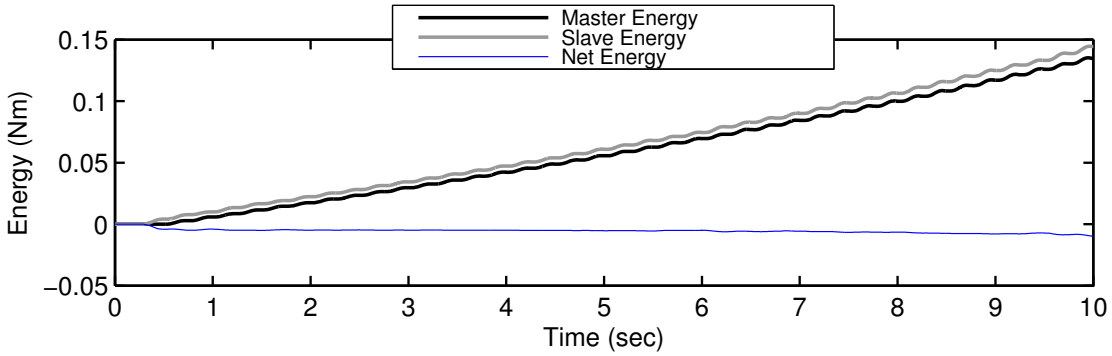

Figure 6.31: **Variable stiffness**, Delays $T_f = 0.25s$, $T_b = 0.25s$, energies in continuous time

10 N/m whereas B is varied from 0 to 1 $N.s/m$. This is an interesting result as the effects of stiffness and damping tend to cancel each other to the benefit of stability. Fig. 6.39 shows good velocity tracking. Continuous and discrete energies in Figs. 6.41 and 6.42 exhibit stable and almost passive behavior.

The slave passivity controller α_s dissipates the negative energy in a regular and smooth fashion as can be seen from Fig. 6.43.

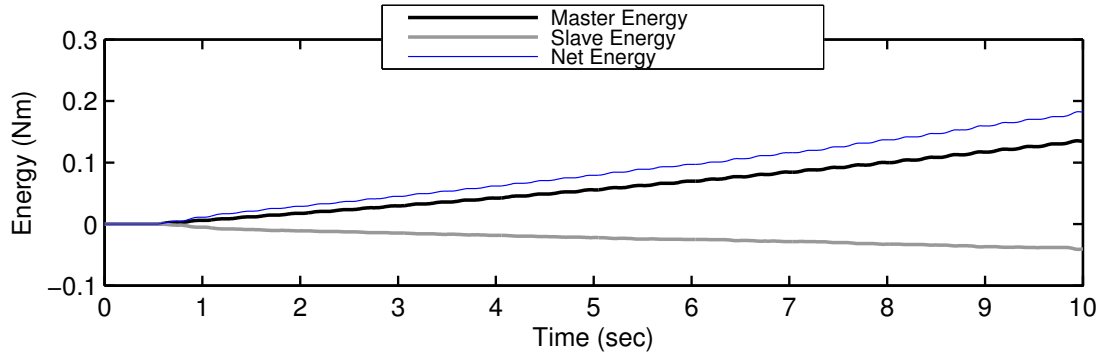

Figure 6.32: **Variable stiffness**, Delays $T_f = 0.25s$, $T_b = 0.25s$, discrete energies

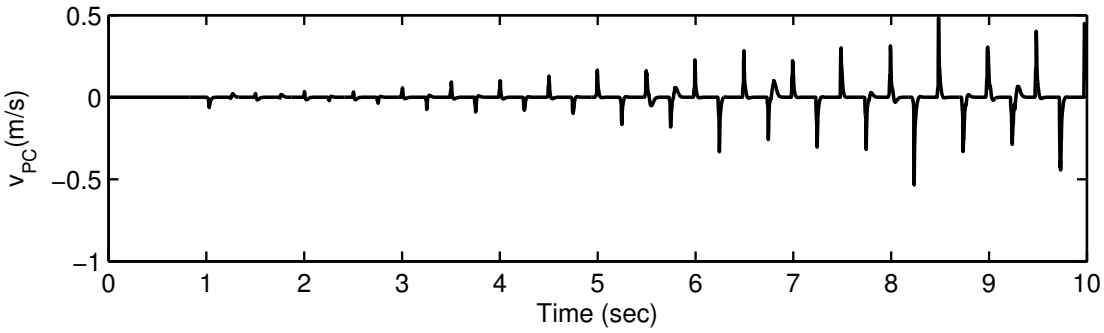

Figure 6.33: **Variable stiffness**, Delays $T_f = 0.25s$, $T_b = 0.25s$, v_{PC}

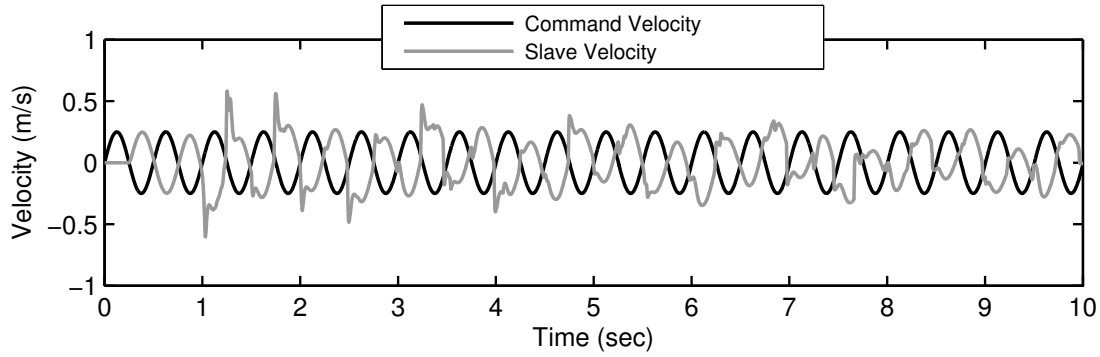

Figure 6.34: **Variable damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, command and slave velocities

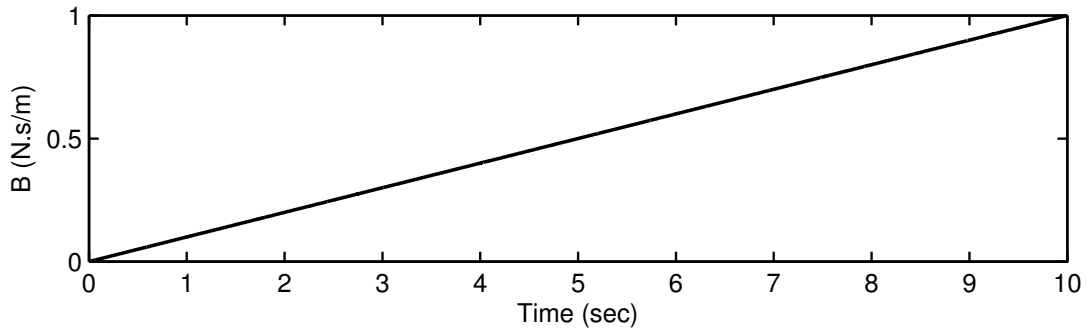

Figure 6.35: **Variable damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, B

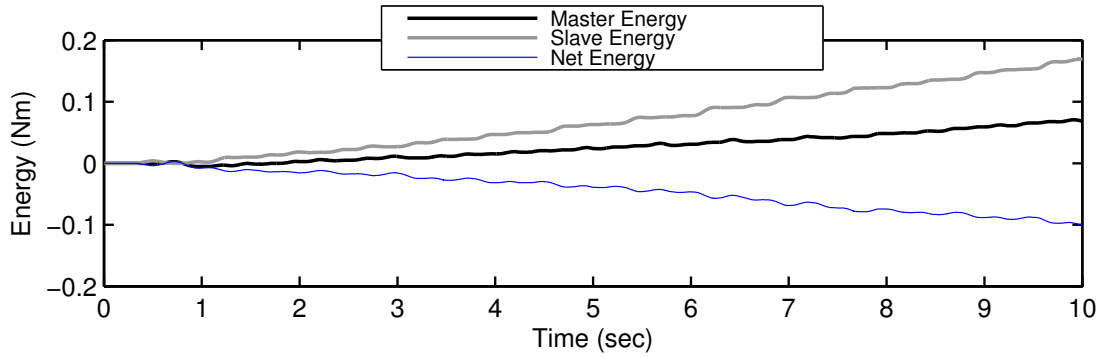

Figure 6.36: **Variable damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, energies in continuous time

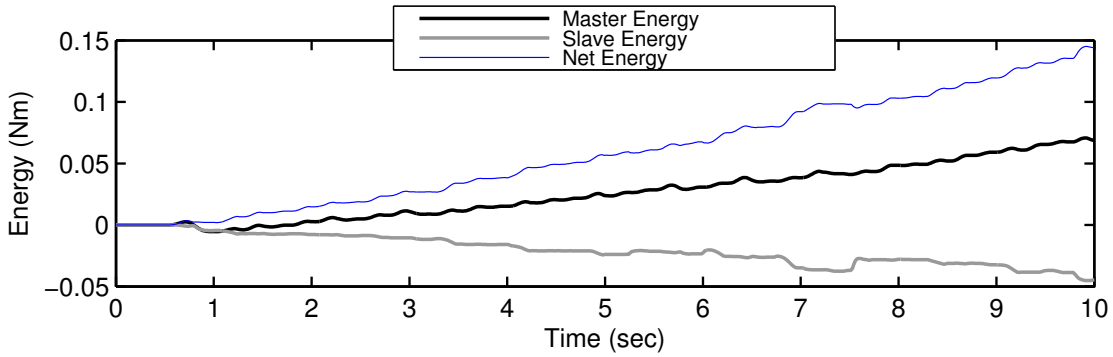

Figure 6.37: **Variable damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, discrete energies

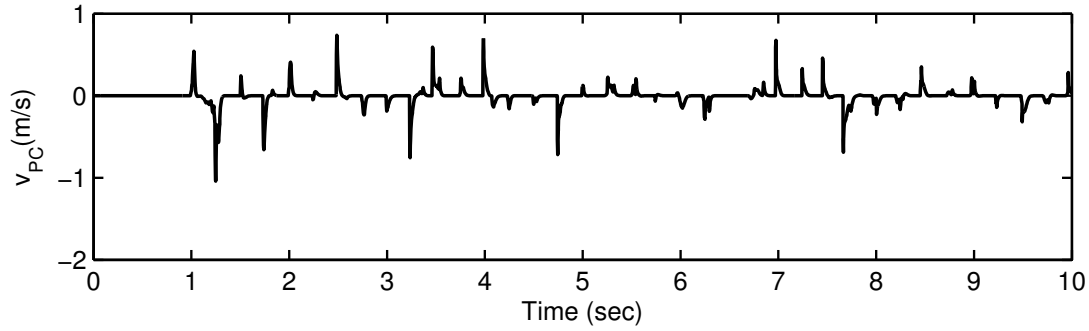

Figure 6.38: **Variable damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, v_{PC}

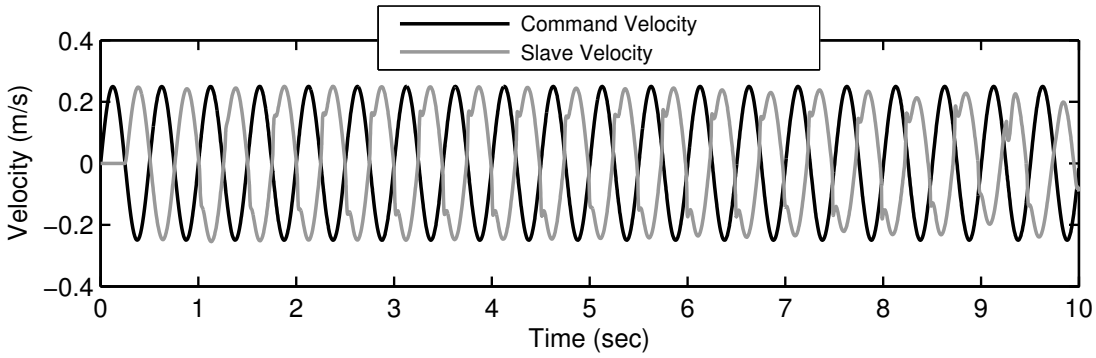

Figure 6.39: **Time varying stiffness and damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, command and slave velocities

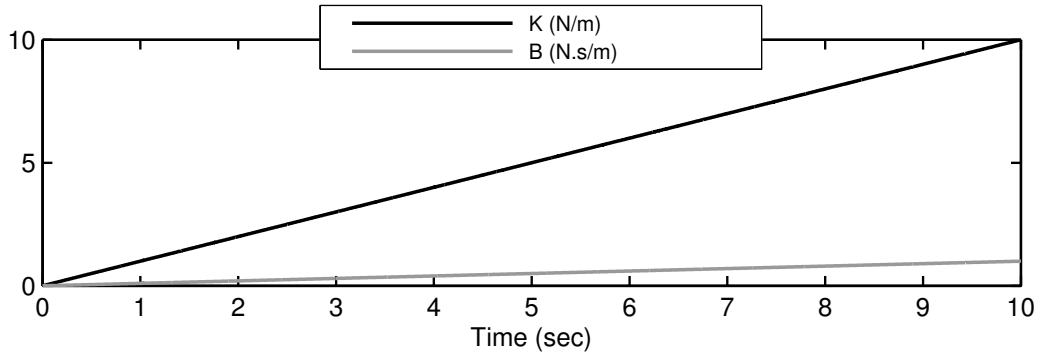

Figure 6.40: **Time varying stiffness and damping**, Delays $T_f = 0.25s$, K and B

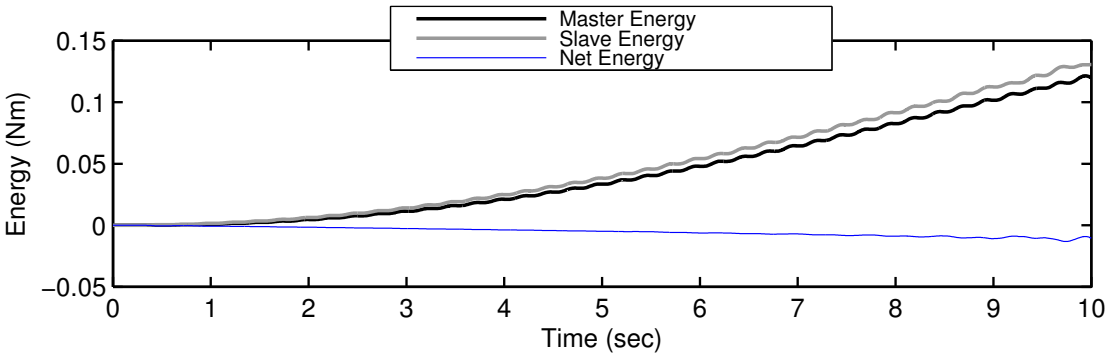

Figure 6.41: **Time varying stiffness and damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, energies in continuous time

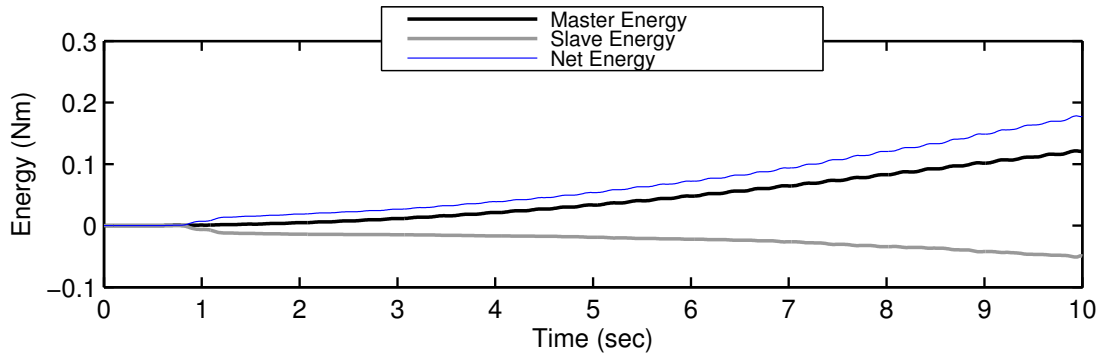

Figure 6.42: **Time varying stiffness and damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, discrete energies

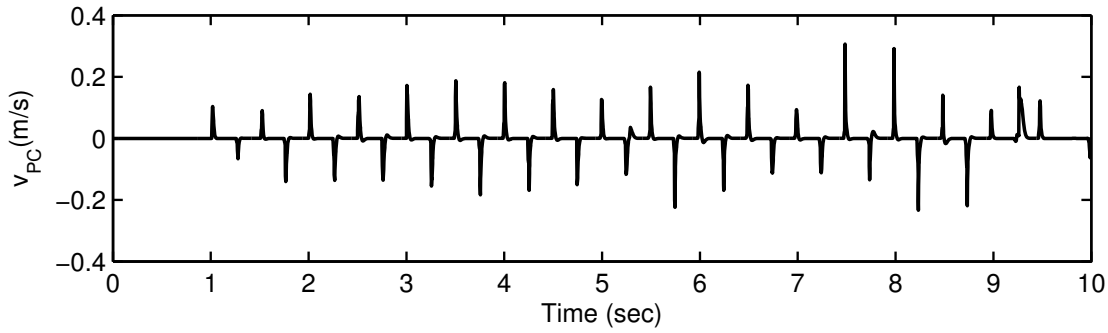

Figure 6.43: **Time varying stiffness and damping**, Delays $T_f = 0.25s$, $T_b = 0.25s$, v_{PC}

Chapter 7

Stabilization in the Presence of Time Varying Delays

This chapter discusses the application of the Time Domain Passivity Control to the stabilization of bilateral teleoperation when the delays in the communication channel are variable with respect to time. Very few contributions in teleoperation research have dealt with the issue of variable time delay and most of them are based on wave-variable based methods in the presence of time delays, see [13, 23, 51]. Technical literature other than teleoperation applications, however does exist dealing with the question of stability in the presence of variable time delay, see e.g., [10, 35]. To the best of author's knowledge, this is the first work dealing with the issue of variable (as well as constant) time delays in teleoperation using Time Domain Passivity Control.

The stabilization scheme described in chapter 6 depends on the knowledge of time delay in order to estimate the net value of energy, so prediction of RTT (Round Trip Time) is an integral part of this approach. RTT is defined as:

$$RTT = FTT + BTT \tag{7.1}$$

where FTT is forward trip time and BTT is backward trip time.

Mirfakhrai and Payandeh[52] proposed an AR-model based approach using parameter lookup tables to predict delays throughout different times of day. Ye et. al.[53] maintain that $RTTs$ have high correlation and given enough observations during certain times of days and weeks, a reliable model can be constructed for their prediction. Yang et. al. [54] have given a good survey of methods in use for the prediction of internet end-to-end delays. It has been reported in literature[55, 56] that Internet delays roughly follow a beta distribution. In this work, A linear one-step ahead predictor is used to estimate the varying time-delays.

7.1 Generation and Prediction of Time-Varying Delays

Keeping in view the results of [55, 56, 52, 51], it can be safely assumed that Internet time-delays in a given teleoperation setup over Internet can be predicted to a reasonable degree of accuracy provided the knowledge of traffic over sufficient period of time. Consequently in this work, the simulation delays are generated that follow beta pdf around a likely mean value:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{7.2}$$

where $0 < x < 1$ and:

$$B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx \tag{7.3}$$

The cumulative distribution function for a beta distributed random variable is given by:

$$F(x;\alpha,\beta) = \int_0^x \frac{1}{B(\alpha,\beta)} y^{\alpha-1}(1-y)^{\beta-1}dy \tag{7.4}$$

Beta distribution is quite flexible because it can fit many different probability distribution densities. Particularly it can be used to model a variable that takes on values over a bounded interval and assumes one of the shapes governed by the parameters[57]. In a practical setup, the mean value of this distribution would come from and would change based on the network traffic analysis. To generate β distributed random values for use in simulations, we can use the following well-known result from statistics.

If Y_1 and Y_2 are independent random variables, where Y_1 has a gamma distribution with parameters α and 1, and Y_2 follows a gamma distribution with parameters β and 1, then

$$X_\beta = \frac{Y_1}{Y_1 + Y_2} \tag{7.5}$$

follows a β distribution with parameters α and β. To generate gamma random variable for use in (7.5), the following result can be used[57].

The sum of t independent exponentials U with the same parameter λ, is a gamma random variable with parameters λ and t. This gives the following:

$$X_\gamma = -\frac{1}{\lambda} log U_1 - \cdots - -\frac{1}{\lambda} log U_t \tag{7.6}$$

which can be simplified as:

$$X_\gamma = -\frac{1}{\lambda} log(U_1 \times \cdots \times U_t) = -\frac{1}{\lambda} log\left(\prod_{i=1}^{t} U_i\right) \tag{7.7}$$

Using (7.5) and (7.7), desired β-distributed stochastic time delays can be generated to be used in the simulations.

To estimate these $RTTs$, a first order predictor is used. As noted in [54], RTT cannot be estimated by just doubling the FTT or BTT, so it is chosen to predict the RTT based on the previous measurements of the same. In such a case, if required, separate predictors would be required for each FTT and BTT. If $RTT(n)$ is described as $\Delta(n)$, then one-step ahead predictor based on previous two values of RTT can be written as:

$$\hat{\Delta}(n+1) = (\Delta(n-1) - \Delta(n-2))\rho(n) + \Delta(n-2) \tag{7.8}$$

where

$$\rho(n) = \frac{t_n - t_{n-2}}{t_{n-1} - t_{n-2}} \tag{7.9}$$

It should be noted that this prediction is subject to upper and lower bounds of RTT. It can be seen that equally spaced RTT measurements are not required. A 1^{st}-order predictor is prone to noisy response so a simple low-pass filter is used to smoothen the time delay prediction. Estimating RTT by using a *ping* command can furnish intial values. A local estimator, however, is required to deliver RTT estimates at every controller calculation step.

7.2 Design of Passivity Controllers

As formulated in section 6.2, the effect of time-delay is canceled in the calculation of passivity controllers, so their values are computed as given in Tables 6.7 and 6.9, depending on the choice of techniques selected. The passivity controllers given in Table 6.9 that incorporate both the energy derivatives as well as parabolic power integration are used in the simulations given in this chapter.

In order to calculate the time delay in real-time based on the RTT-predictor given in section 7.1, we need to provide time references. These reference can be generated by clock signals in the simulation setup as shown in Fig 7.3 giving the details of a passive network $2 - port$ in the presence of time-varying delays. As the Kalman estimator requires the synchronized input, output pairs of force and velocity, so a circular buffer of several previous values of both variables needs to be maintained which is synchronized on the arrival of every time stamp.

The use of energy derivatives as explained in section 6.3 proves to be an invaluable tool in the stabilization of teleoperation with variable time delays. Because the time delay is considerably longer than the control loop frequency and is always varying, so inter-sample behavior is very important in the case of variable time delay. Energy derivative provides us with a look into this period and allows the passivity controller design algorithm to take care of extra dissipation should the current state approaches an active behavior. It has been observed that the system resolves to near instability in the absence of energy derivative approach.

Because the time stamps are used to calculate the RTT as well as to synchronize the input and output of the joint model of slave arm and the environment, so the

data in forward and backward channel needs to be joined, similar to packets in a packet switched network, in order to maintain coherency in the transmitted signals.

The stabilization approach proposed for variable time delays using the delay predictor and with the provision of time base in the form of time stamps, is shown in Fig. 7.1.

7.3 Simulation and Results

In order to simulate the system, we generate stochastic time delays as described in section 7.1. The parameters of the teleoperation system are kept same as in previous simulations for the sake of comparison.

Figs. 7.2-7.9 show the different variables of simulated system for β-distributed delays with a mean value of $415ms$. Variable time delays and a histogram of RTT are given in Figs. 7.6 and 7.7. The histogram shows beta distribution approaching a normal distribution. Good velocity tracking is observed in Fig. 7.2 as the continuous and discrete energies show a stable system in Figs. 7.3 and 7.4. A little noisy response close to $t = 9s$ is a result of higher variance in the FTT, shown as T_f in Fig. 7.6. Afterwards, the system becomes stable again as can be judged by diminishing outputs (in terms of slave velocity modifications) of passivity controller in Fig. 7.5.

Fig 7.8 shows the parameters of 3^{rd} order model of slave side teleoperation components. A comparison of estimated and actual $RTTs$ is shown in Fig. 7.9 which shows the filtered values of 1^{st}-order RTT predictor.

The given teleoperation model was simulated with different parameters for β-distribution and stable results were obtained for RTT delays with values less than $500ms$. Nevertheless, it should be noted that an increase in the variance of RTT directly influences the stabilization technique by causing errors in the prediction of slave energy. In such a case, one needs to subscribe to better RTT-estimation/prediction techniques using the knowledge of network traffic or other heuristic methods like Neural Networks trained on certain sections of network/internet during a variety of variable conditions.

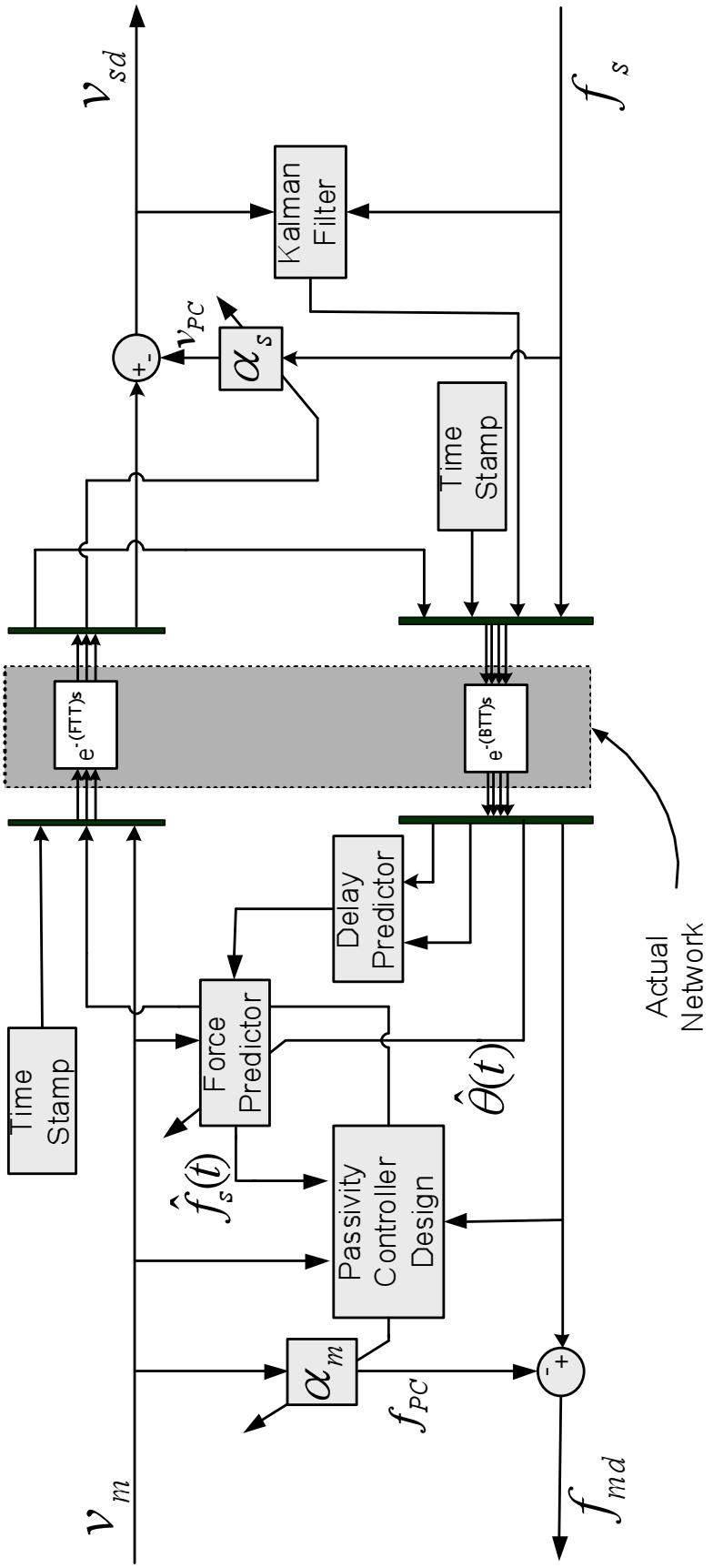

Figure 7.1: Passive network 2-port in the presence of variable time delays using Predictive Time Domain Passivity

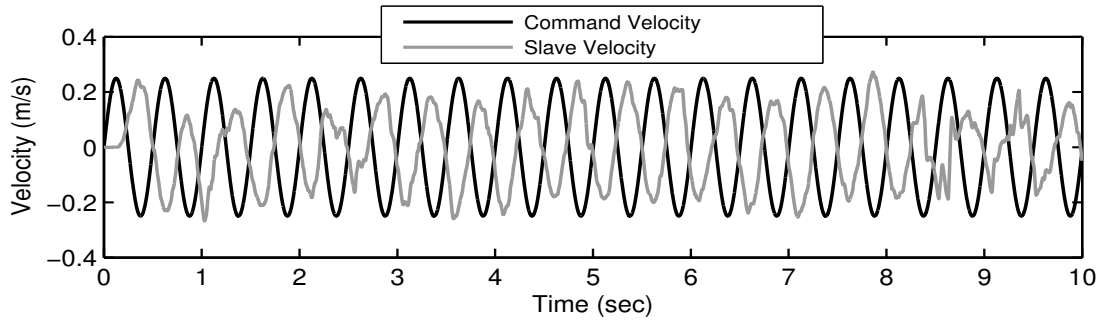

Figure 7.2: Time varying delay, command and slave velocities

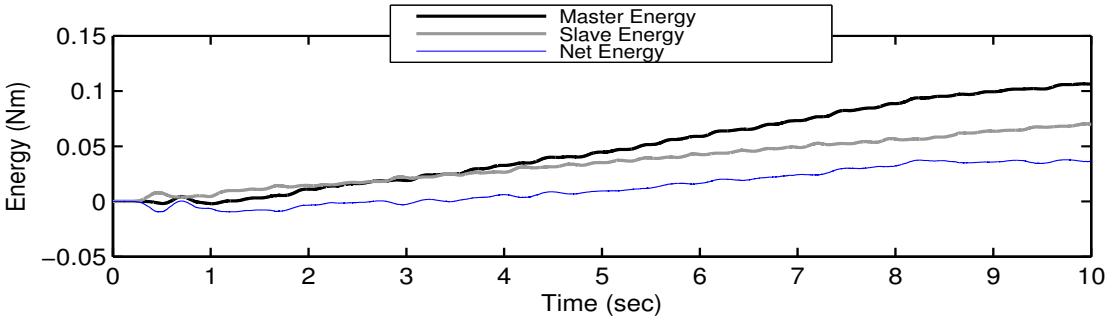

Figure 7.3: Time varying delay, continuous energies

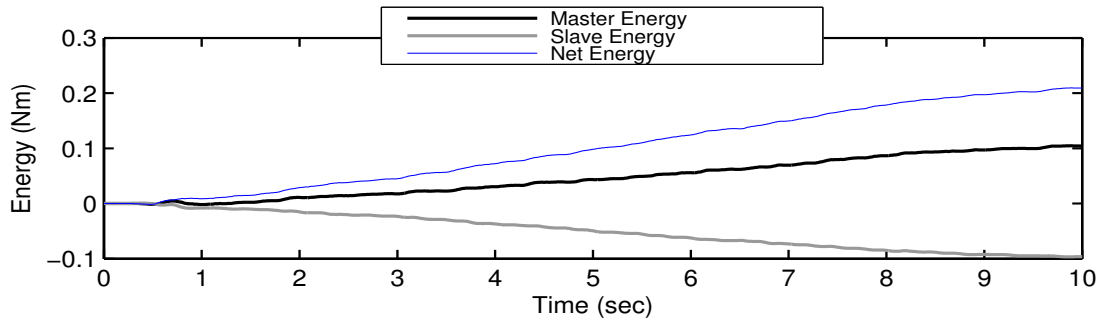

Figure 7.4: Time varying delay, discrete energies

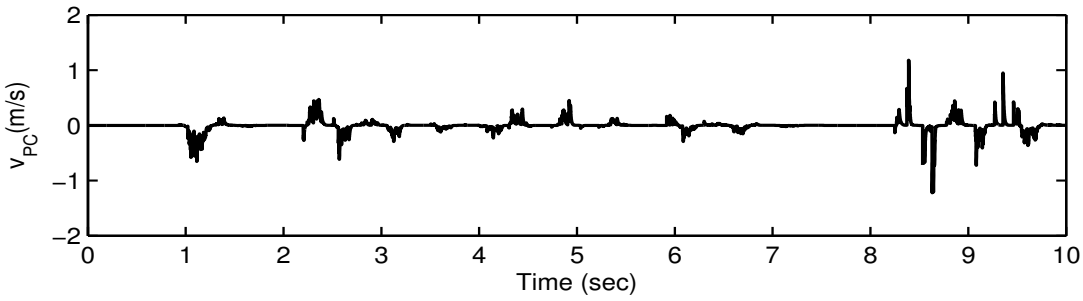

Figure 7.5: Time varying delay, v_{PC}

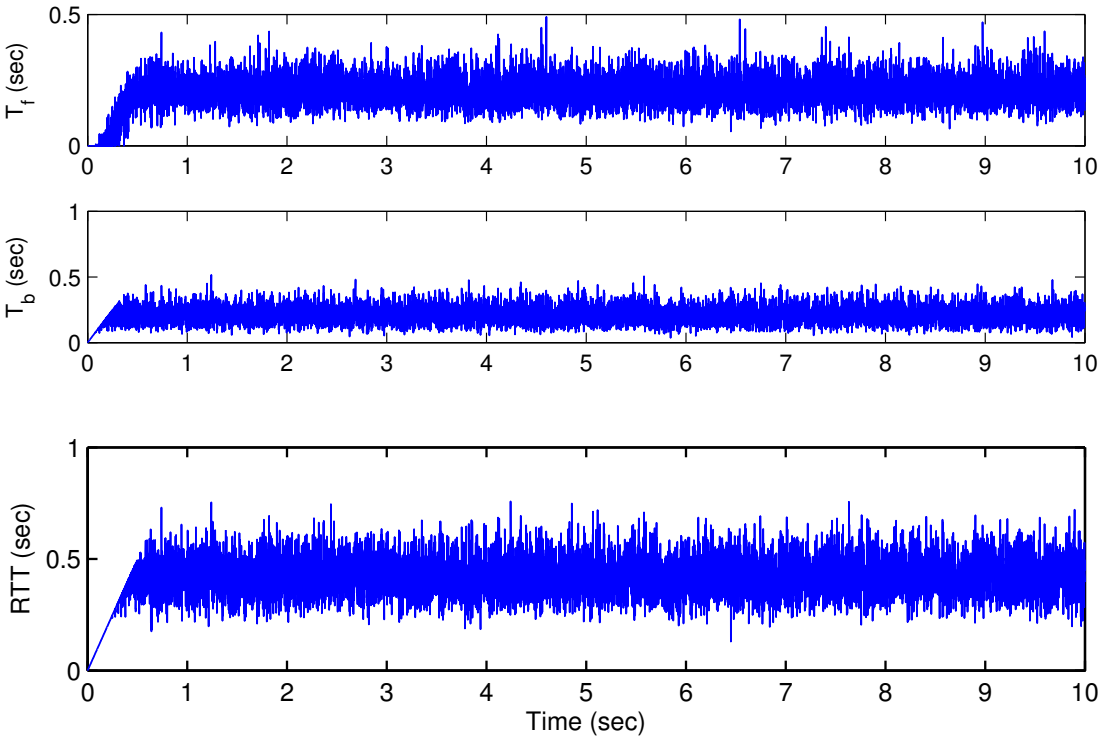

Figure 7.6: Delays in the system

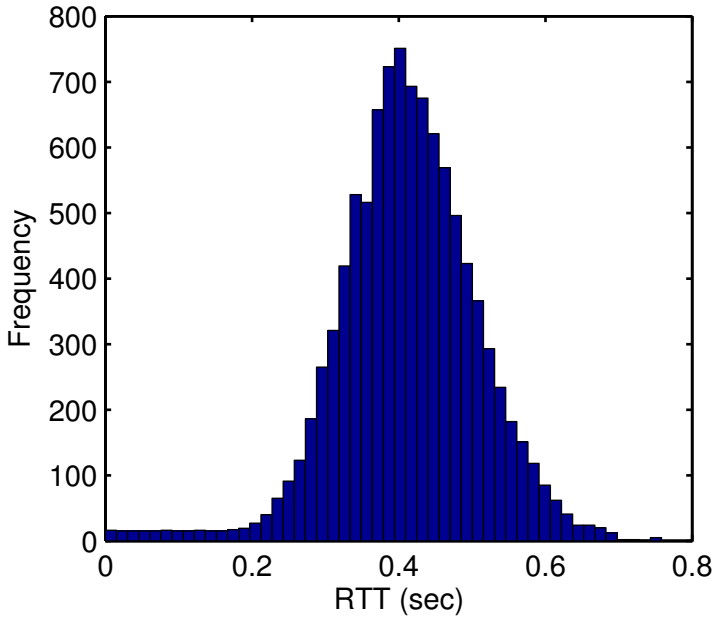

Figure 7.7: Histogram of RTTs

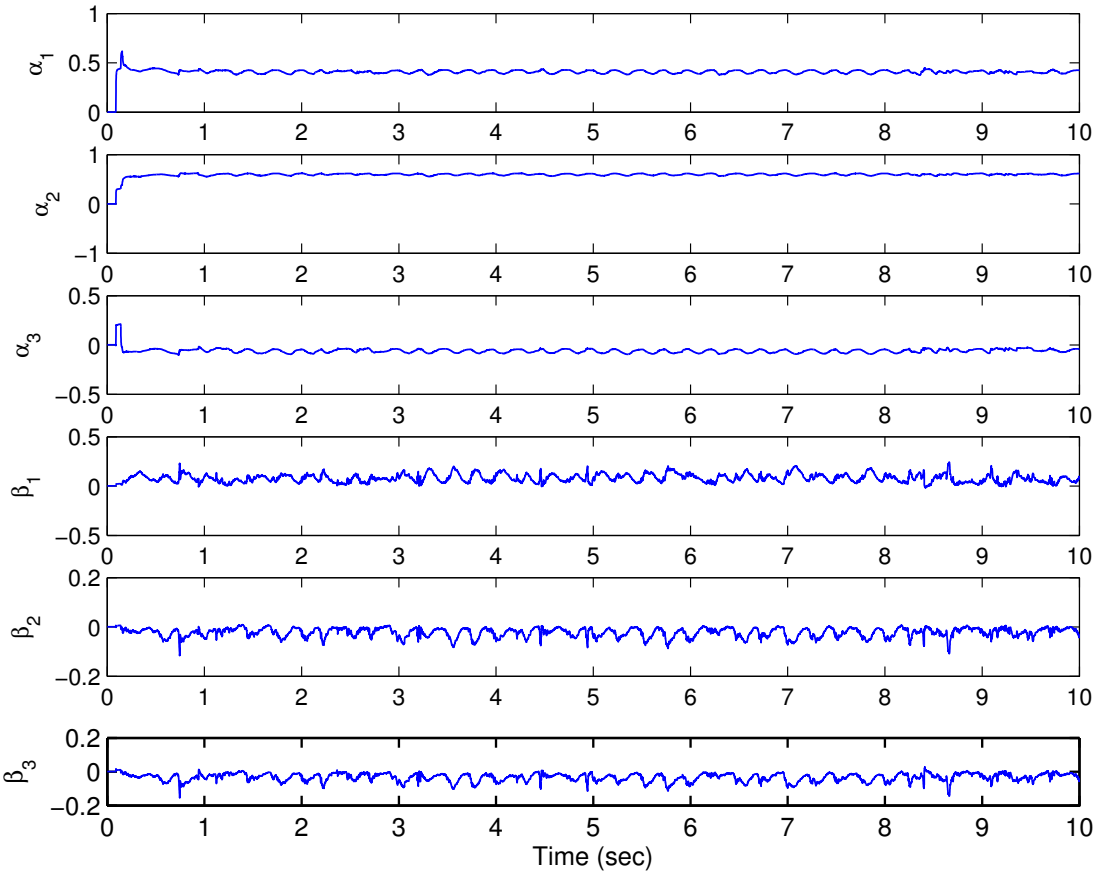

Figure 7.8: Time varying delay, online estimated parameters

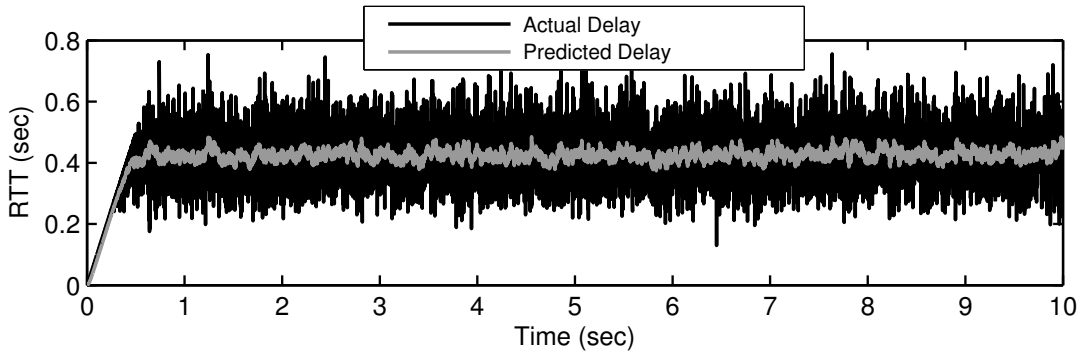

Figure 7.9: Actual vs. predicted RTTs

7.4 Implementation using TrueTime Network Simulator

TrueTime is a Matlab tool for real-time network simulation. A brief overview of TrueTime can be found in Appendix A. Here TrueTime is introduced into the teleoperation system in order to simulate the existing scheme over switched ethernet. By this way, one can investigate the effects of the variable time delay on the stability and performance of teleoperation system. The already discussed teleoperation model will continue to be used but with a TrueTime network in place of time-delay blocks. In addition, to facilitate data packeting several A/D and D/A blocks will be used to convert analog signals to and from the digital domain. Different components of this scheme are described in the following sections.

7.4.1 Communication Channel

The communication channel component of TrueTime network is the main interface between the master and the slave. It can serve as switched ethernet or wireless LAN, as well as can model itself for several other network types. The use of the internet for remote monitoring and control applications is attractive due to its ubiquity, cost and standardization of equipment and communication protocols. The 1994 Mercury Project at USC (University of Southern California) first enabled the control of a robotic manipulator using HTTP protocol. The main challenges, when for example the internet is used as a communication link, from a control theoretic point of view in the teleoperation system, are the time-varying delay, packet loss, or even the complete loss of the connection. If provided that no time delay via the network exists, the stability of the teleoperation system can be guaranteed. This problem is linked to delay-induced power generation in the communication channel which violates passivity arguments as has previously been described. Here the TrueTime simulator is used to simulate the communication channel. Fig. 7.10 shows the complete teleoperation system model with a TrueTime network added.

Inside the network block in Fig. 7.10, the TrueTime network is connected as shown in Fig. 7.11. Having a look at this figure also reveals that this block also provides us with the network schedule generated during the simulation.

This network connects nine nodes and is used as switched ethernet with a data rate of 100 Mbps and the minimum frame size of 100 bytes.

A/D Converter

Since the network receives a digital signal only, an A/D converter is required. This can be realized by the use of a TrueTime *computer node* as seen in Fig. 7.12.

List 7.1 is the initialization of the A/D converter for the forward channel. The A/D code function is shown in List 7.2.

Listing 7.1: The A/D Converter Initialization Function

```
function ADC_F_init
```

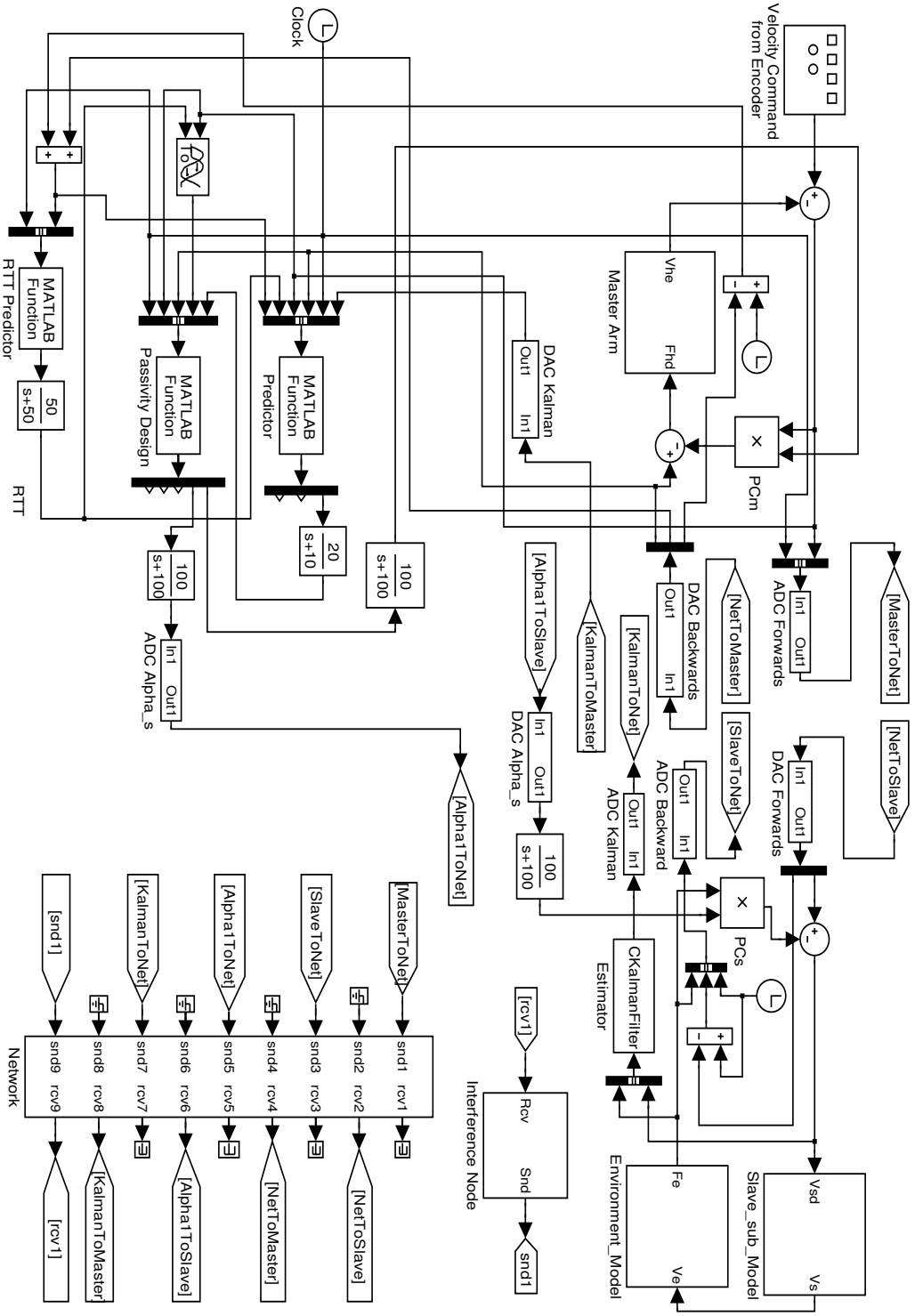

Figure 7.10: Main components of the simulation framework with TrueTime simulator

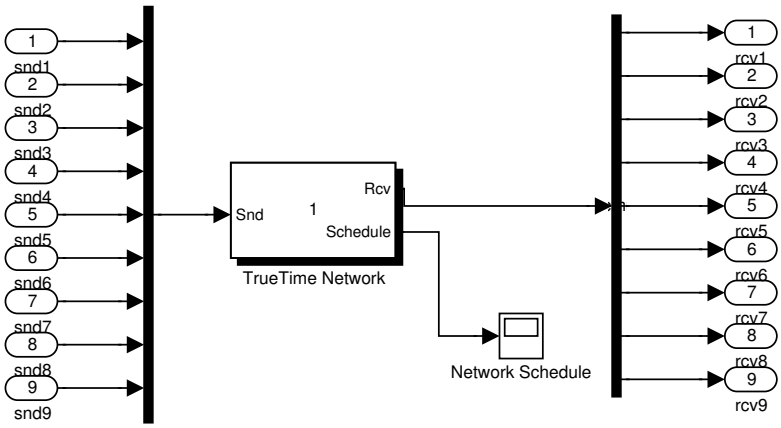

Figure 7.11: TrueTime Network Configuration

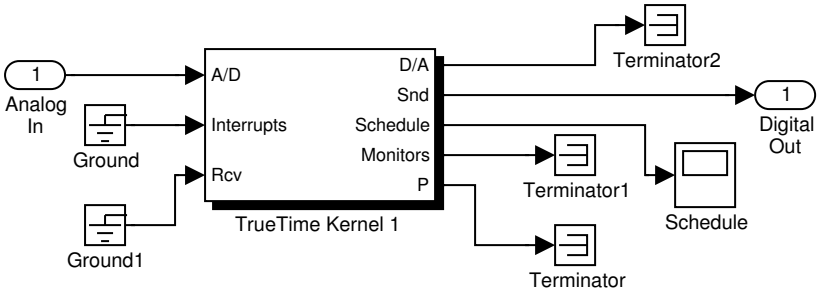

Figure 7.12: The A/D Converter

```
% Initialize ADC_Forward

% nbrOfInputs, nbrOfOutputs, fixed priority
ttInitKernel(2,0,'prioFP');
% Create ADC task
data.y = 0;
offset = 0;
period = 0.005;
prio = 1;
ttCreatePeriodicTask('ADC_F_task',offset,period,prio,'
    ADC_F_code',data);
% Initialize network
ttCreateInterruptHandler('nw_handler', prio, 'msgSndToNet_F');
% Node#1 in the network
ttInitNetwork(1, 'nw_handler');
```

Listing 7.2: The A/D Converter Code Function

```
function [exectime,data] = ADC_F_code(seg,data)

switch seg,
    case 1,
```

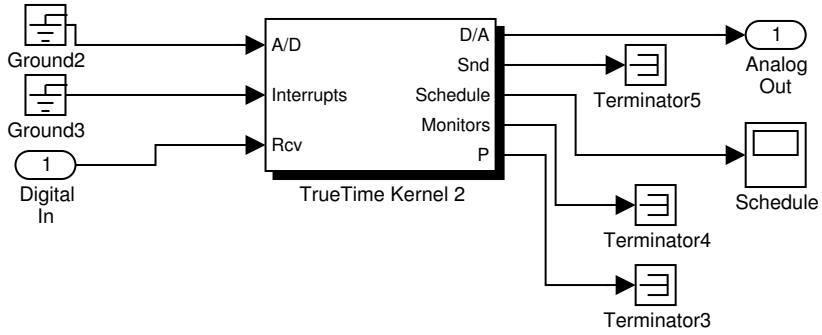

Figure 7.13: The D/A Converter

```
        for k=1:2
            data.y(k) = ttAnalogIn(k);
        end
            exectime = 0;
    case 2,
        % Send message to node 2
        ttSendMsg(2, data.y, 10);
        exectime = 0.0001;
    case 3,
        % finished
        exectime = -1;
end
```

D/A Converter

The D/A converter is a computer node that converts the digital received signal from the network to its analog version. Fig. 7.13 shows the D/A converter. List 7.3 is the initialization of the D/A converter in the forward channel and List 7.4 shows the D/A converter code function.

Listing 7.3: The D/A Converter Initialization Function

```
function DAC_F_init
% Receives messages from the Network and perform DAC
% Initialize DAC_Forward

% nbrOfInputs, nbrOfOutputs, fixed priority
ttInitKernel(0, 2, 'prioFP');

deadline = 1000;
prio = 1;
ttCreateTask('DAC_F_task', deadline, prio, 'DAC_F_code');

% Initialize network
ttCreateInterruptHandler('nw_handler', prio, 'msgRcvFromNet_F')
    ;
```

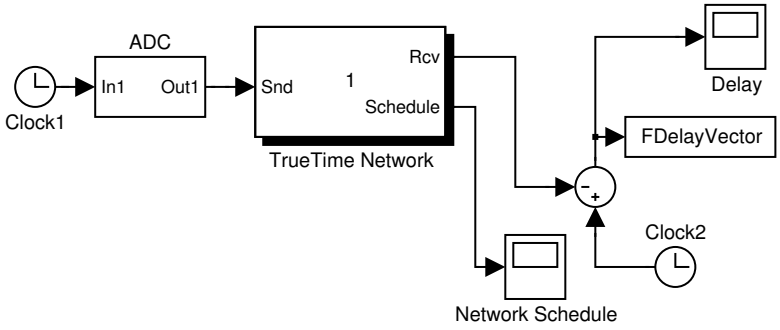

Figure 7.14: Time Delay Measurement

```
% Node#2 in the network
ttInitNetwork(2, 'nw_handler');
```

Listing 7.4: The D/A Converter Code Function

```
function [exectime,data] = DAC_F_code(seg,data)

switch seg,
    case 1,
        data.u = ttGetMsg;
        exectime =0.0001;
    case 2,
        for i=1:2
            ttAnalogOut(i,data.u(i));
        end
        exectime = 0;
    case 3,
        exectime = -1; %finished
end
```

Interference Node

This node has been added to increase the time delay in the communication channel to make it work in a more realistic way. The function of this node is to send junk data to occupy the channel bandwidth and thus increase the time delay and jitter in the transmitted signals.

Time Delay Measurement

A time stamp is used to measure the time delay within the communication network. Fig. 7.14 shows the method by which the time delay is measured.

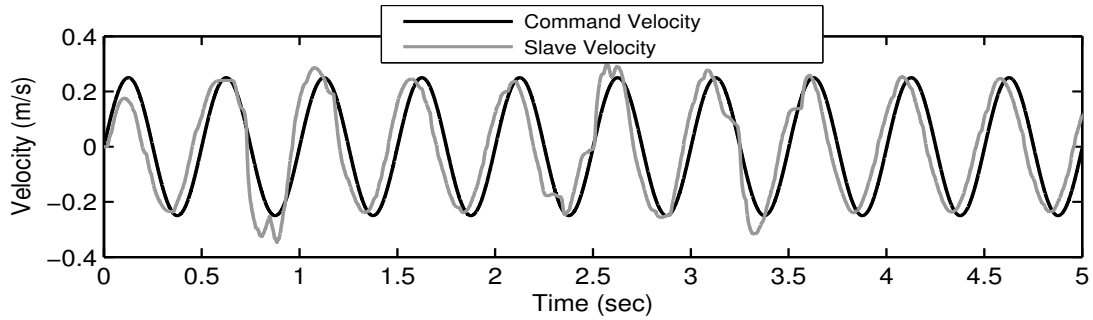

Figure 7.15: TrueTime network, interference 90% of channel bandwidth, command and slave velocities

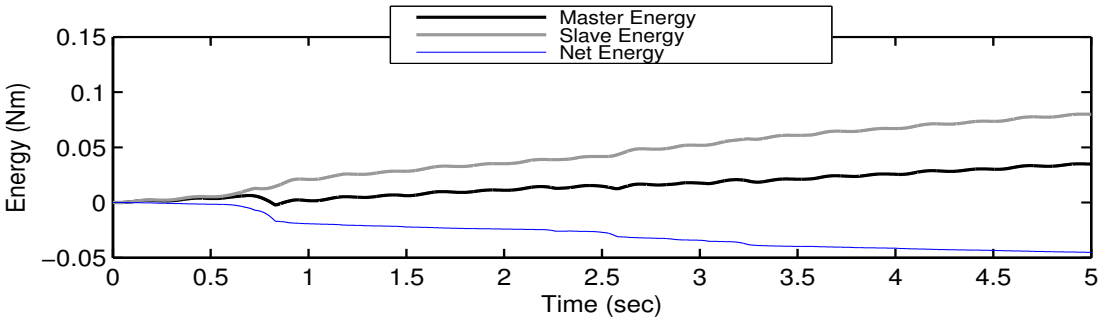

Figure 7.16: TrueTime network, interference 90% of channel bandwidth, continuous energies

7.4.2 Simulation Results

The main control loop, in all simulations, runs at a sampling rate of 10 ms in real-time sense. Kalman filter update rate is also fixed at 10 ms.

Following are the results where the interference node randomly transmits 100 bytes with a period of $5.5ms$ and consumes up to 90% bandwidth of the communication channel. The mean value of delay is $33.2ms$. It can be observed that even in the case of real-time simulations and real-time switched ethernet network block, the passivity controllers are stablizing the teleoperation as is shown in Figs. 7.15, 7.16, and 7.17.

Fig. 7.19 shows the slave passivity contribution to regulate the energy in terms of modifications in slave velocity.

$RTTs$ in the teleoperation framework are shown in Fig. 7.20 with a histogram of the same, following (exponential) beta distribution, in Fig. 7.21.

Fig. 7.22 shows the online identification of the joint model of slave robot and the environment. Scheduling of the network transmissions for various physical variables in the teleoperator is plotted in Fig. 7.23. Pre-emptions in the transfer of velocity and force information because of interference and estimator data can be easily spotted.

Now if the interference node is allowed to operate on its 100% requirement, i.e., about $145Kbps$, one observes a marginal degradation in the tracking because of increased pre-emptions, however, the system is still perfectly stable as shown in

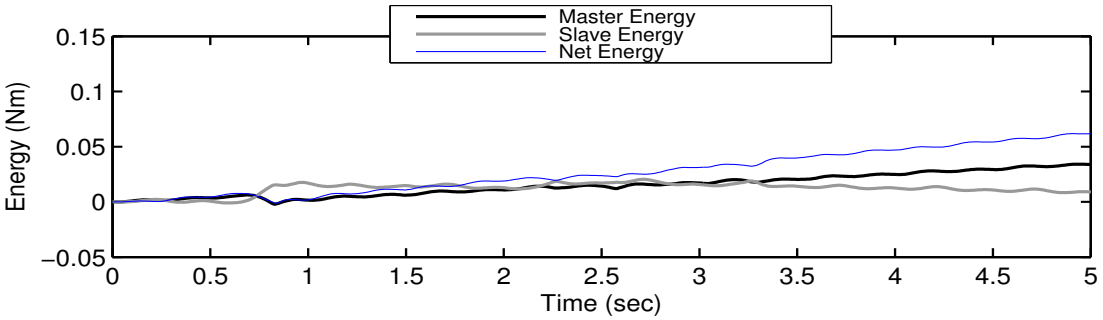

Figure 7.17: TrueTime network, interference 90% of channel bandwidth, discrete energies

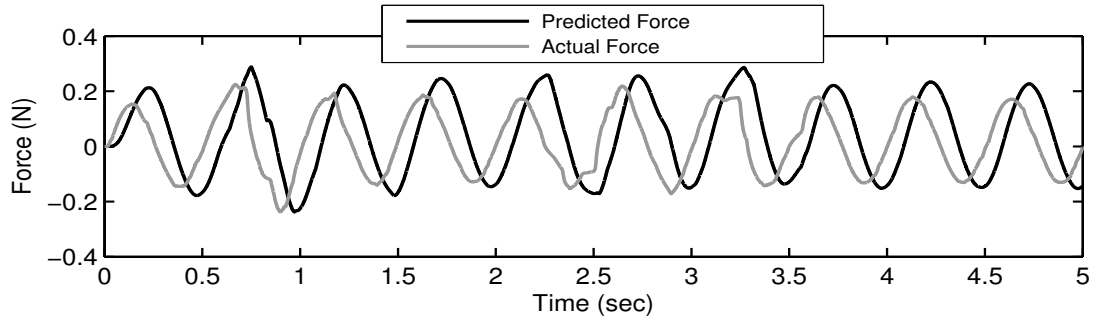

Figure 7.18: TrueTime network, interference 90% of channel bandwidth, predicted vs. actual force

Figs. 7.24-7.28. The mean value of delay in this case is $57ms$ close to almost all of real-life appliations involving switched networks.

If the interference load on the network is increased to 300 $Kbps$, with an interference load of 300 bytes every $8ms$, the delay increases to a mean value of $95ms$ and the tracking becomes a bit noisy though still the system is completely stable, refer to Figs. 7.29-7.33.

It is clear that the communication network introduces destabilizing effects into the bilateral teleoperation system, such as time-varying delay and loss of data. The appropriate reconstruction of lost- and tardy data due to time-varying delay

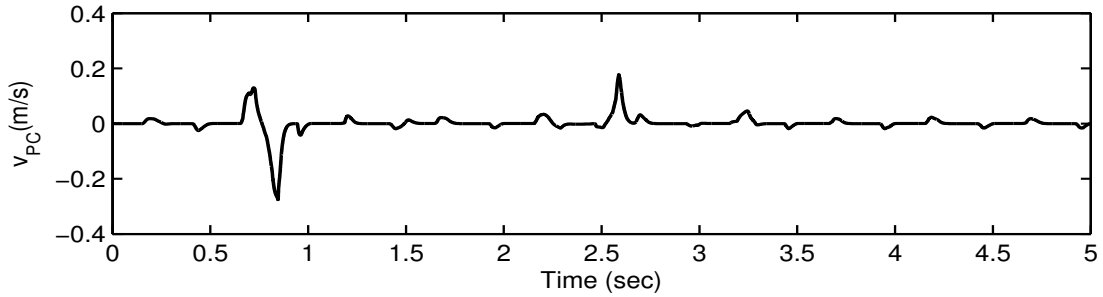

Figure 7.19: TrueTime network, interference 90% of channel bandwidth, v_{PC}

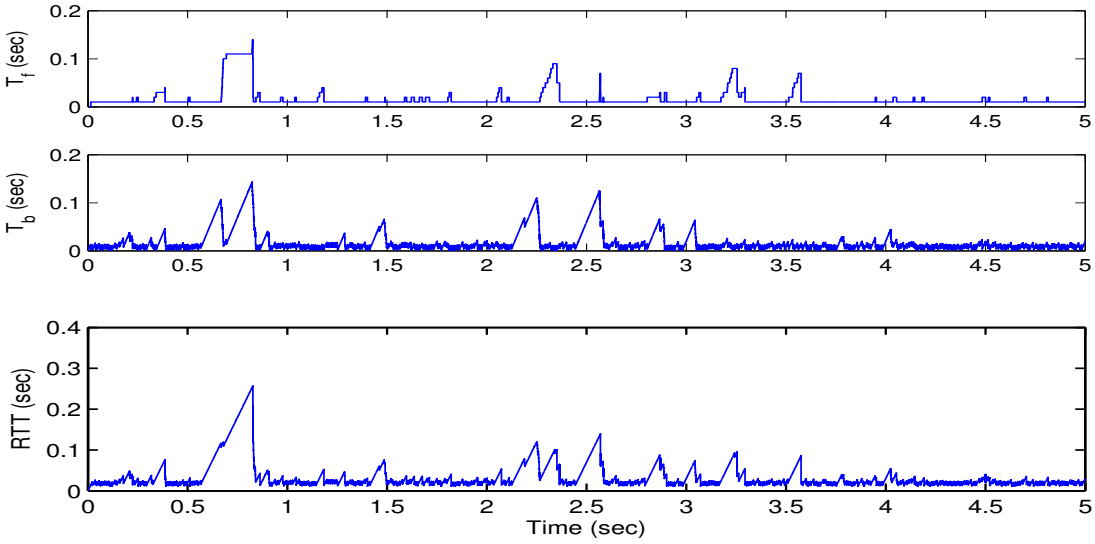

Figure 7.20: TrueTime network, interference 90% of channel bandwidth, Delays in the system

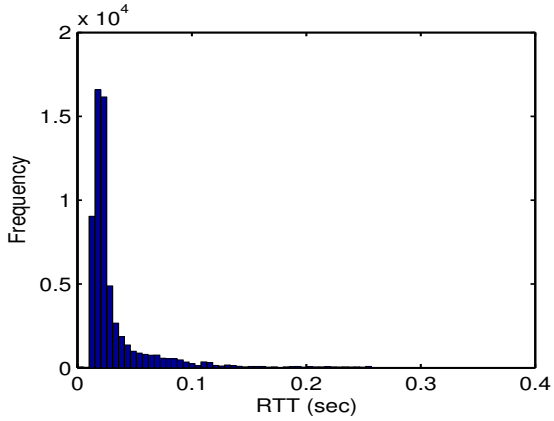

Figure 7.21: TrueTime network, interference 90% of channel bandwidth, histogram of RTTs

is identified as the main issue for the stability of teleoperation systems in communication network. The results shown indicate that the system stays stable even up to a network delay of $400ms$ as shown in Figs. 7.29 and 7.32.

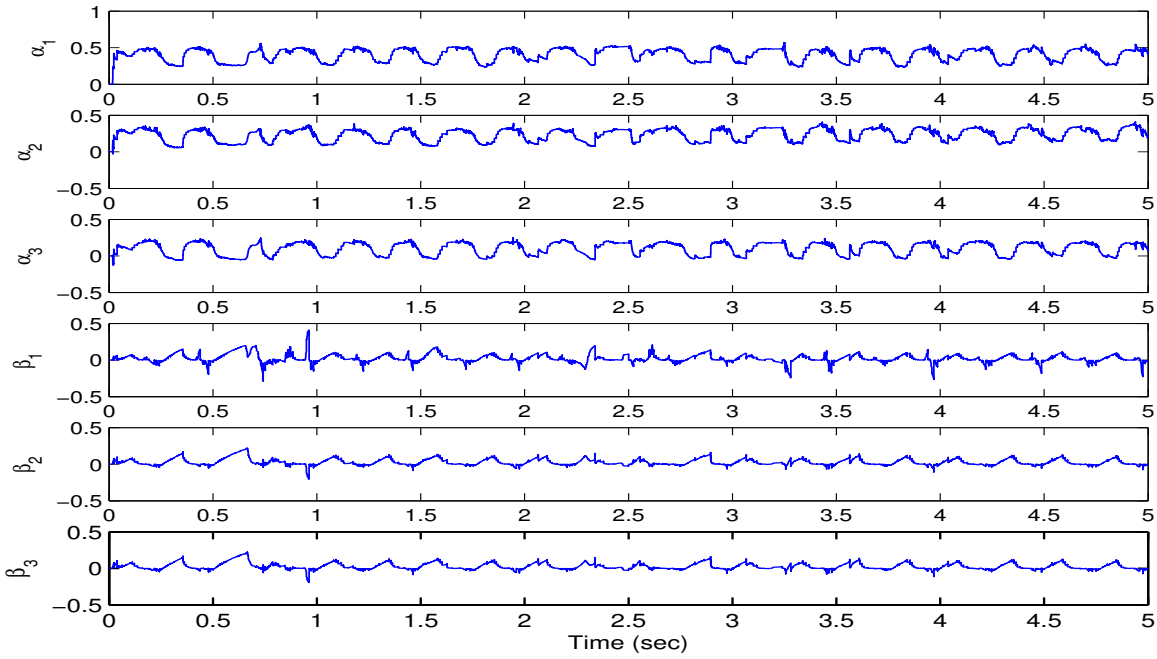

Figure 7.22: TrueTime network, interference 90% of channel bandwidth, online estimated parameters

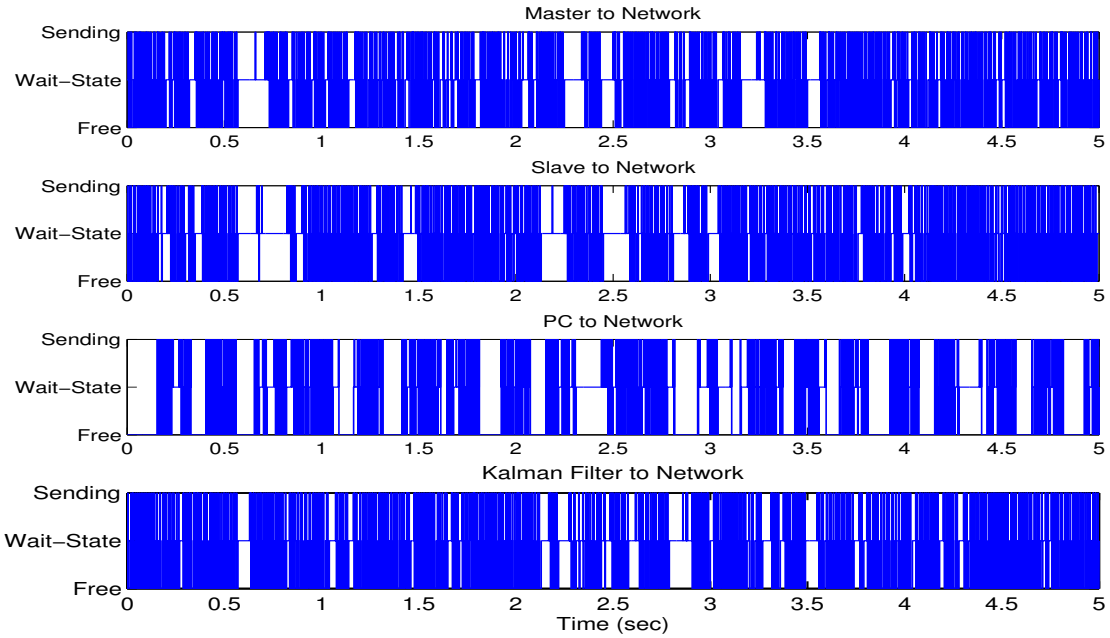

Figure 7.23: TrueTime network, interference 90% of channel bandwidth, network schedule

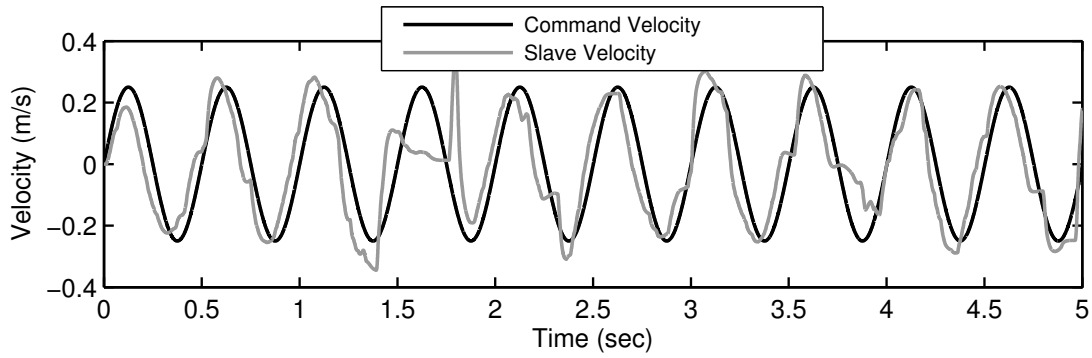

Figure 7.24: TrueTime network, interference: 100 bytes at every 5 ms, command and slave velocities

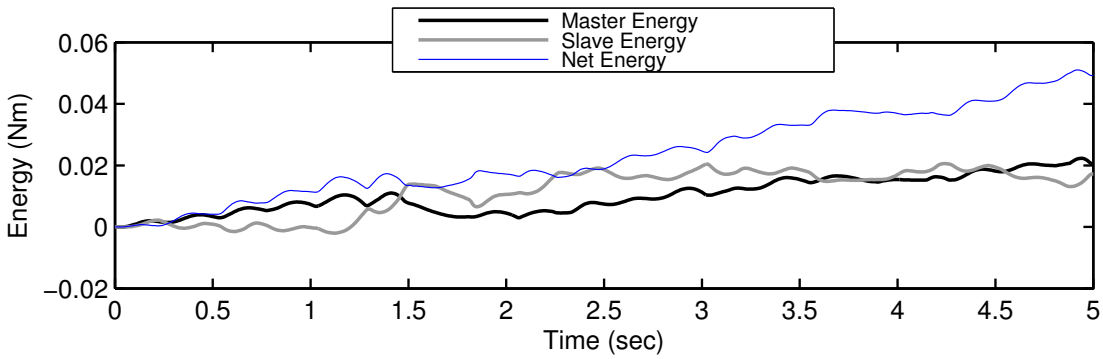

Figure 7.25: TrueTime network, interference: 100 bytes at every 5 ms, discrete energies

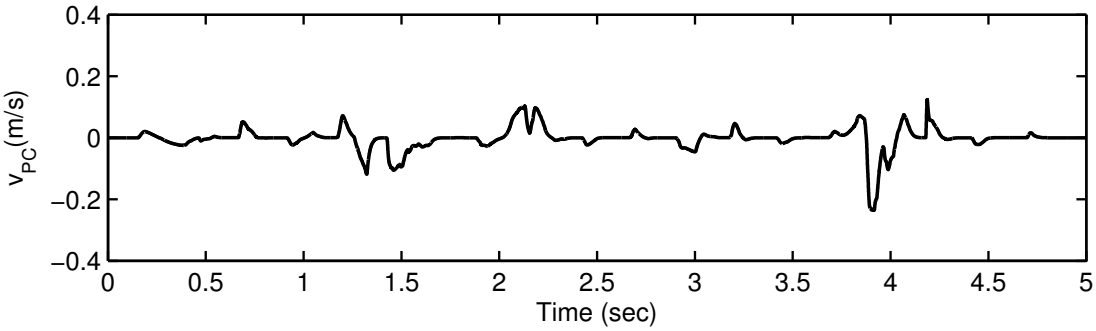

Figure 7.26: TrueTime network, interference: 100 bytes at every 5 ms, v_{PC}

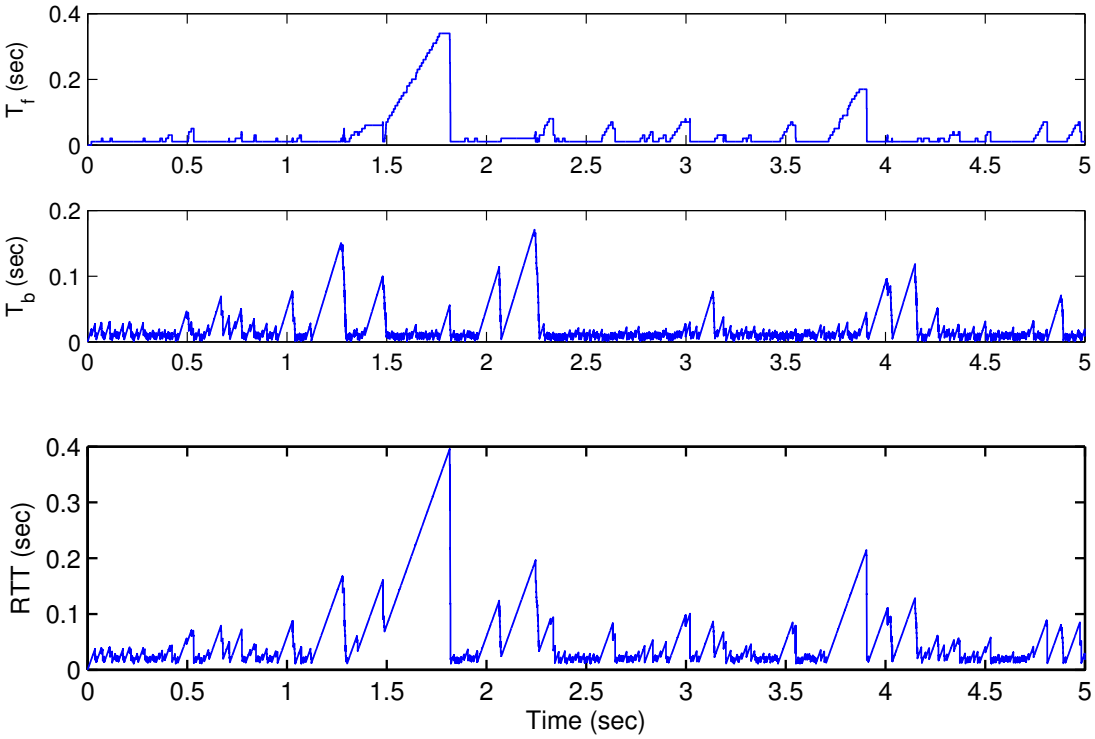

Figure 7.27: TrueTime network, interference: 100 bytes at every 5 ms, Delays in the system

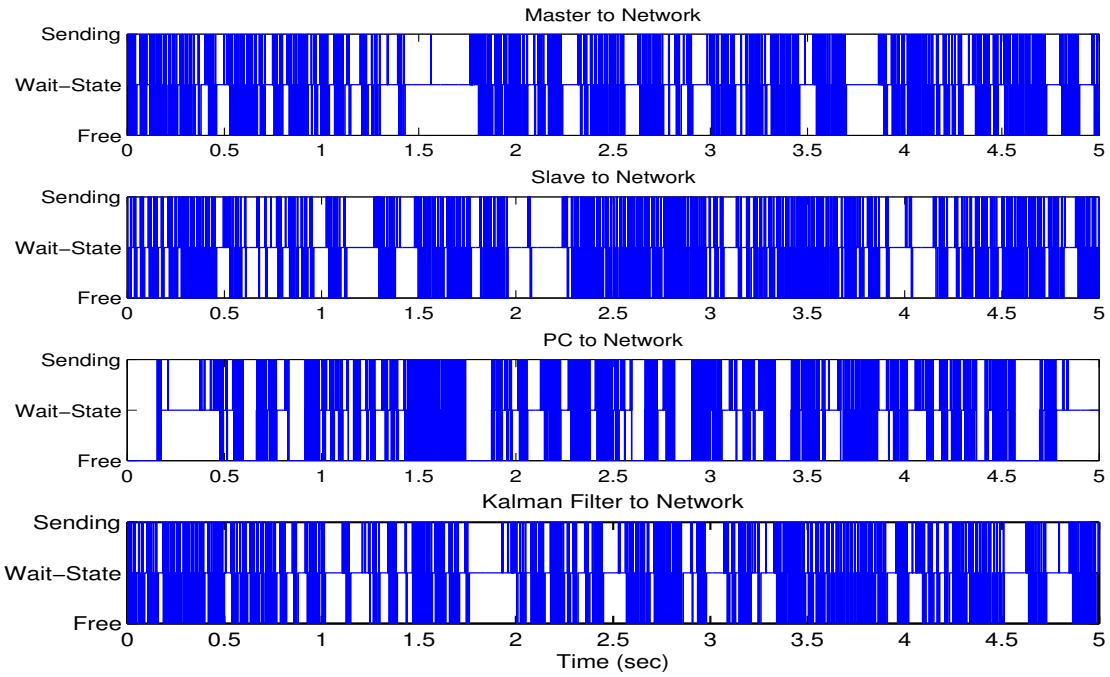

Figure 7.28: TrueTime network, interference: 100 bytes at every 5 ms, network schedule

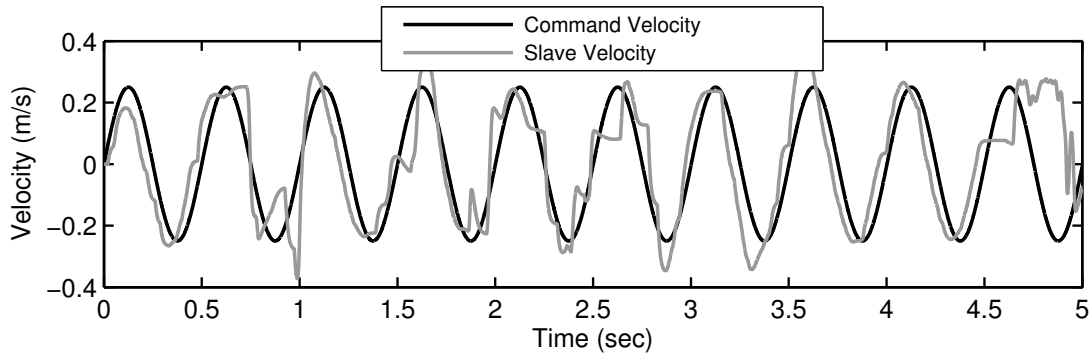

Figure 7.29: TrueTime network, interference: 300 bytes at every 8 ms, command and slave velocities

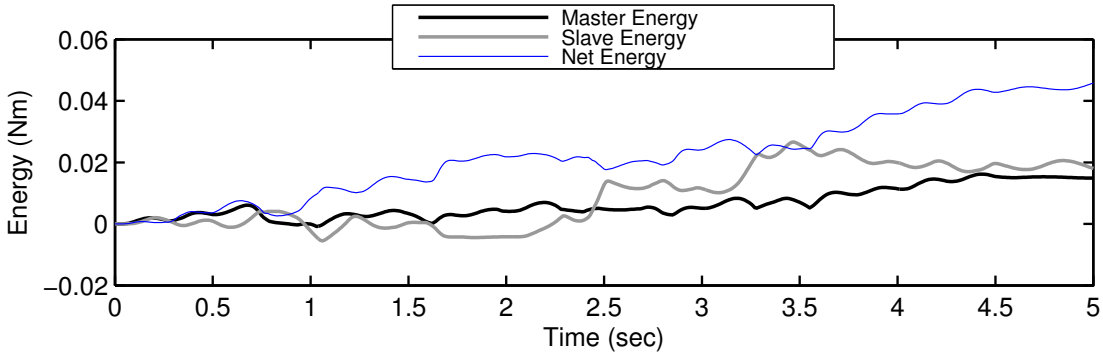

Figure 7.30: TrueTime network, interference: 300 bytes at every 8 ms, discrete energies

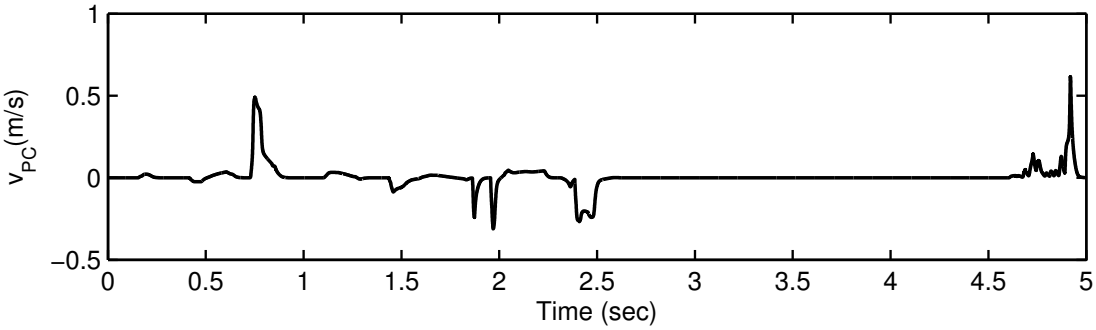

Figure 7.31: TrueTime network, interference: 300 bytes at every 8 ms, v_{PC}

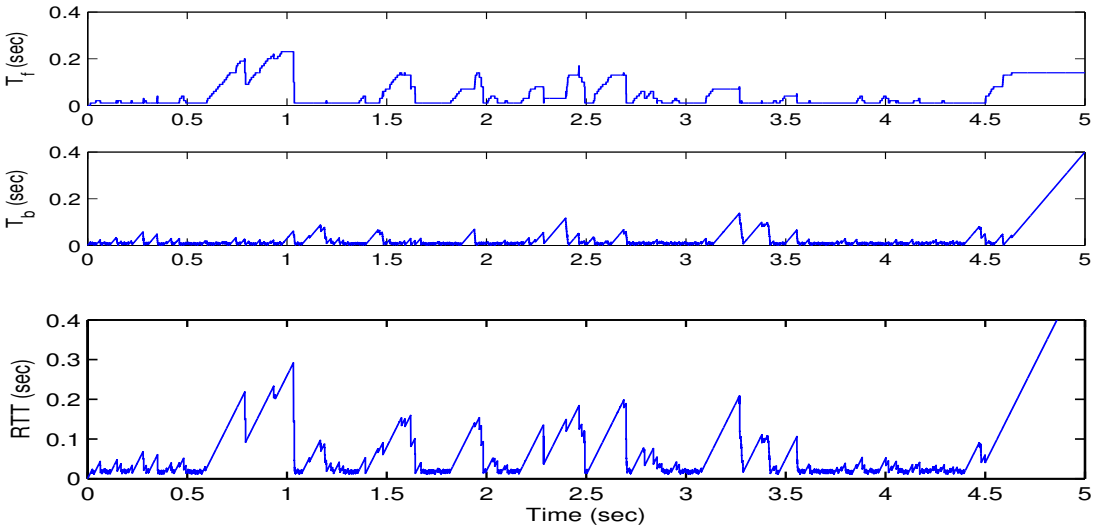

Figure 7.32: TrueTime network, interference: 300 bytes at every 8 ms, Delays in the system

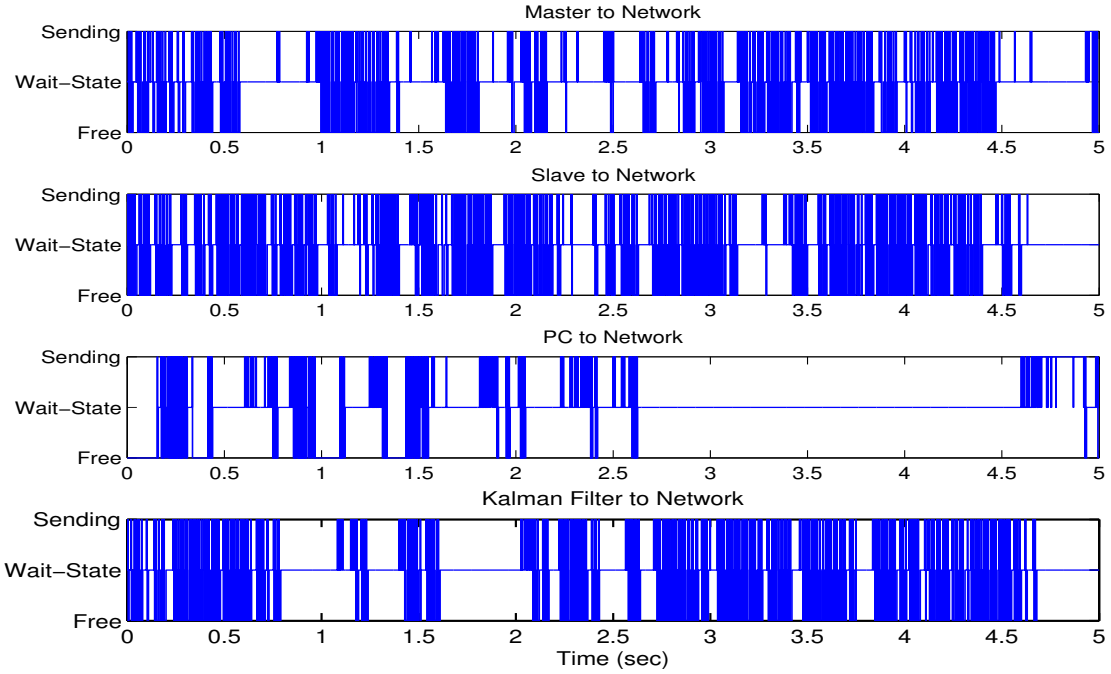

Figure 7.33: TrueTime network, interference: 300 bytes at every 8 ms, network schedule

Chapter 8

Discussion of Results and Conclusions

8.1 Discussion of Results

In this dissertation, application of Time Domain Passivity Control[44, 28, 31, 30, 32], to the stabilization of delayed teleoperation has been proposed, developed and evaluated. The proposed stabilization scheme is simulated in the following different scenarios:

1. Constant time delays in the communication channel
2. Variable environment parameters
3. Variable time delays using delay blocks
4. Variable time delays using real-time network simulation

In all of these cases, very good stabilization is achieved as described in chapters 6 and 7.

As is quite natural, the stabilization, in the case of constant time delays is close to optimal and allows for greater uncertainties in the rest of the teleoperation system as a manifestation of robustness.

The design of the stabilization scheme started with the derivation of master and slave passivity controllers as described in section 6.2.1. The results obtained with this approach, given in Fig. 6.3-6.8 exhibit the need for improvement and further development.

Consequently, section 6.3 explains how we can use real-time derivative of net energy to compensate for such dynamics of teleoperation system, that may lead to an active behavior and thus to the loss of passivity resulting in instability. The use of energy derivatives in the design of passivity controllers (see Table 6.7) resulted in a highly transparent system which dissipates energy more frequently with an added advantage that the energy dissipated at each sampling time becomes less spiky. In this step, the Kalman filter based parameter estimator of the joint model of slave arm and the environment is also moved to the slave side of the network

$2-port$ to allow for better dynamic estimation and other benefits as described in section 6.3.2. The results thus obtained showing highly improved velocity tracking and almost zero net discrete energy are given in section 6.3.3.

Furthermore, the stabilization approach is improved in the area of net discrete energy estimation. Previous sections dealt with energy estimation using rectangular summation of power values on master and slave ports of the network. Section 6.4.1 describes non-linear estimation of net energy using Simpson's rule of parabolic integration. Parabolic Power Integration not only provided more smooth and distributed energy dissipation as discussed in section 6.4.3, but also brought about a 10% increase in velocity tracking.

To highlight the dynamic nature of developed scheme, section 6.5 discusses the application to an environment with time-varying parameters. Owing to online estimation of environment parameters, the stabilization of such a setup with i) variable stiffness, ii) variable damping, and iii) both variable stiffness and damping, is achieved as the results in section 6.5.1 indicate.

In chapter 7, the stabilization of teleoperation using Time Domain Passivity Control in the presence of time varying delays of both stochastic and random (real-time network generated) nature is discussed. Stable tracking of velocity signal with a RTT of up to $700ms$ is obtained as shown by the results in section 7.3. There is, however, a slight loss of transparency when compared to the results in the case of constant time delay given in section 6.4.3.

In order to further evaluate the system in an almost real-life scenario, the stabilization approach is applied to TrueTime real-time network simulator based teleoperation system given in section 7.4. The results given show stable teleoperation in a switched ethernet in the presence of interfering traffic and $RTTs$ up to a value of $500ms$.

8.2 Contributions

During the course of PhD research, the author has significantly contributed to the scientific knowledge in the area of teleoperation. Some of the achievements made through this research are listed below:

1. Extension of Time Domain Passivity Control to the stabilization of a communication channel with simultaneous stabilization on impedance (force regulation) and admittance (velocity regulation) causalities.

2. Design and implementation of energy prediction concept in bilateral teleoperation.

3. Extension of Time Domain Passivity Control to the stabilization of teleoperation systems with constant time delays.

4. Extension of Time Domain Passivity Control to the stabilization of teleoperation systems with variable time delays.

The contributions to scientific work, including the off-campus research collaboration in teleoperation, has produced the following publications:

1. Asif Iqbal and Hubert Roth. Stabilization of Teleoperated Systems with Stochastic Time Delays using Time Domain Passivity, *Proc. of SICE-ICASE International Joint Conference 2006 (SICE-ICCAS 2006), BEXCO, Busan, Korea*, October 2006.

2. Asif Iqbal and Hubert Roth. Predictive Time Domain Passivity Control for Delayed Teleoperation using Energy Derivatives, *Proc. of The 9th International Conference on Control, Automation, Robotics and Vision, ICARCV 2006, Singapore*, December 2006.

3. Asif Iqbal and Hubert Roth. Time Domain Passivity Control for Delayed Teleoperation with Parabolic Power Integration, *Proc. of The 7th International Conference On Technical Informatics (CONTI'2006), TIMISOARA, ROMANIA*, June 2006.

4. Mayez A. Al-Mouhamed, Onur Toker, and Asif Iqbal. A Multi-Threaded Distributed Telerobotic Framework, *IEEE Transactions on Mechatronics*, November 2006.

5. Asif Iqbal, Otto Roesch, Hubert Roth, and Asad Rasool. Using Meta-Heuristics in the Control of a Non-Linear Input Delay Laboratory Helicopter System, *Proc. of The 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005, Seville, Spain*, December 2005.

6. Asif Iqbal, Hubert Roth and Moneeb Abu-Zaitoon. Stabilization of Delayed Teleoperation using Predictive Time-Domain Passivity Control, *Proc. of IASTED International Conference on Robotics and Application (RA 2005), Boston, USA*, October 2005.

7. Otto J. Roesch, Hubert Roth and Asif Iqbal. Extended Stability Margins on Controller Design for Nonlinear Input Delay Systems, *Proc. of The 16th IFAC World Congress, held in Prague, Czech Republic*, July 2005.

8. Mayez A. Al-Mouhamed, Onur Toker, Asif Iqbal, and Syed M.S. Islam. Evaluation of Real-Time Delays for Networked Telerobotics, *Proc. of The* 3^{rd} *International IEEE Conference on Industrial Informatics, held in Perth, Western Australia*, August 2005.

8.3 Conclusions

Stability in delayed teleoperation is a fundamental issue in present telerobotic systems because of commodity communication links like internet that inherit time delays on account of shared bandwidth in an effort to lower the price tag. This is still an open research area with almost non-existent solutions that address all

possible issues in one stabilization scheme. Time Domain Passivity Control has certain benefits over the rather popular approach based on wave variable concept. In this work, the first application of Time Domain Passivity theory to delayed teleoperation is presented. The stabilization results using the proposed technique are encouraging and offer quite transparent teleoperation both in the case of constant as well as variable time delays.

The concept of prediction of net energy across a communication channel, as proposed in this work, is novel and can certainly be improved. It needs mentioning that longer time delays, lower sampling frequency, and variable conditions of the environment are the factors affecting precise estimation of net energy. Therefore, the concept may find better audience in such teleoperation systems where the force reflections and the command signals are not of high frequency as well as the environment conditions do not change so rapidly. Such areas involve but are not limited to undersea surveillance and manipulation, space telemanipulation, slow moving mobile telerobotics, etc. With better energy estimation and prediction algorithms, this stabilization scheme can be valuable to other teleoperation scenarios as well.

As the basic idea behind the proposed approach is stabilization, so one may find that tracking is not up to the mark in certain cases. This is a realistic comment and can be addressed by augmenting the teleoperation system with a position control loop at a higher level, operating at lower frequency. This will also cater for the numerical drifts in quantization and data-packeting of control signals. The longer sampling times ensure that lesser energy will flow through the outer loop responsible for position control, thus minimizing the disturbance to main teleoperation loop operating on force-velocity architecture.

Higher fidelity can be achieved by separating the control and haptic threads in the computation of controllers on a micro-computer, as proposed in [58], and then using energy bounding algorithm on the force control loop to enable display of high resolution haptics.

Appendix A

TrueTime SIMULATOR

A.1 Introduction

TrueTime is a MATLAB/SIMULINK based simulator that facilitates co-simulation of controller task execution in real time kernels, network transmissions, and continuous plant dynamics. This section also describes the fundamental steps in the creation of a TrueTime simulation including how to write the code that is executed during simulation, how to configure the kernel and network blocks as well as what compilation must be performed to get an executable simulation. The code functions for the tasks and the initialization commands may be written either as C++ functions or as MATLAB m-files.

A.2 Simulation Using TrueTime

In TrueTime, computer and network blocks are introduced as desired. The computer blocks are event driven and execute user-defined tasks and interrupt handlers representing, e.g., I/O tasks, control algorithms, and network interfaces. The scheduling policy of the individual computer blocks is arbitrary and is decided by the user. Likewise, in the network, messages are sent and received according to a chosen network model.

The level of simulation detail is also chosen by the user; it is often neither necessary nor desirable to simulate code execution on instruction level or network transmissions on bit level. TrueTime allows the execution time of tasks and the transmission times of messages to be modeled as constant, random, or data-dependent. Furthermore, TrueTime allows simulation of context switching and task synchronization using events or monitors.

TrueTime can be used in several ways:

- to investigate the effects of timing nondeterminism, caused, for example, by preemption or transmission delays, on control performance
- to develop compensation schemes that adjust the controller dynamically based on measurements of actual timing variations

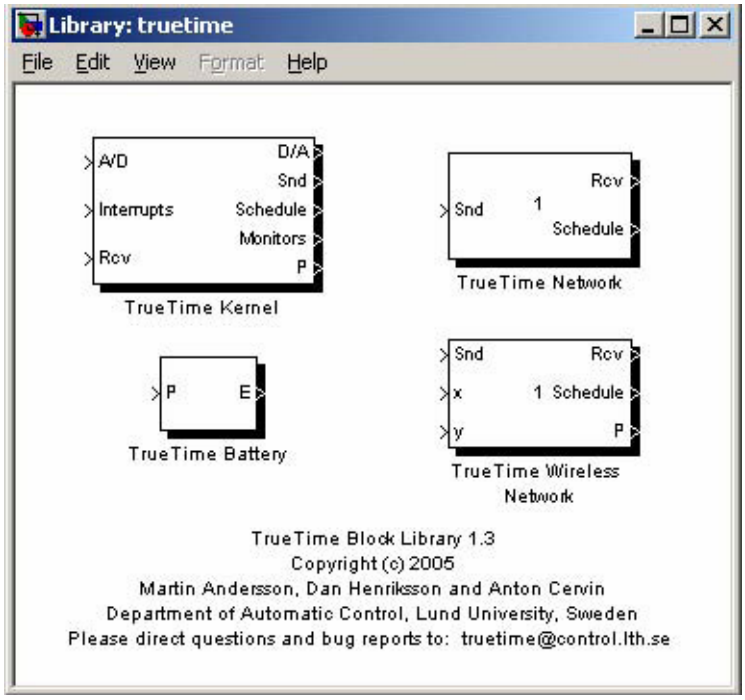

Figure A.1: The TrueTime library

- to experiment with new, more flexible approaches to dynamic scheduling, such as feedback scheduling of CPU time and communication bandwidth and quality-of-service (QoS)-based scheduling approaches
- to simulate event-driven control systems (e.g., engine controllers and distributed controllers).

A.3 Simulation Environment

The interfaces to the computer and network Simulink blocks are shown in Fig. A.1. Both blocks are event driven, with the execution determined by both internal and external events. Internal events are timely and correspond to events such as "a timer has expired," "a task has finished its execution," or "a message has completed its transmission." External events correspond to external interrupts, such as "a message arrived on the network" or "the crank angle passed 0°." The block inputs are assumed to be discrete-time signals, except for the signals connected to the A/D converters of the computer block, which may be continuous-time signals. All outputs are discrete-time signals. The schedule and monitors outputs display the allocation of common resources (CPU, monitors, network) during the simulation.

A.3.1 The Computer Block

The computer block S-function simulates a computer with a simple but flexible real-time kernel, A/D and D/A converters, a network interface, and external in-

terrupt channels. The execution of tasks and interrupt handlers is defined by user-written code functions. These functions can be written either in C++ (for speed) or as MATLAB m-files (for ease of use). Control algorithms may also be defined graphically using ordinary discrete Simulink block diagrams.

A.3.2 The Network Block

The TRUETIME network block simulates medium access and packet transmission in a local area network. When a node tries to transmit a message (using the primitive *ttSendMsg*), a triggering signal is sent to the network block on the corresponding input channel. When the simulated transmission of the message is finished, the network block sends a new triggering signal on the output channel corresponding to the receiving node. The transmitted message is put in a buffer at the receiving computer node. A message contains information about the sending and the receiving computer node, arbitrary user data (typically measurement signals or control signals), the length of the message, and optional real time attributes such as a priority or a deadline. Six models of networks are supported: CSMA/CD (e.g. Ethernet), CSMA/AMP (e.g. CAN), Round Robin (e.g. Token Bus), FDMA, TDMA (e.g. TTP), and Switched Ethernet. The propagation delay is ignored, since it is typically very small in a local area network. Only packet level simulation is supported. It is assumed that higher protocol levels in the kernel nodes have divided long messages into packets, etc. The network block is configured through the block mask dialog, see Fig. A.2. Some parameters can also be set on a per node basis with the command *ttSetNetworkParameter*. The following network parameters are common to all models:

- **Network number:** The number of the network block. The networks must be numbered from 1 and upwards. Wired and wireless networks are not allowed to use the same number.

- **Number of nodes:** The number of nodes that are connected to the network. This number will determine the size of the Snd, Rcv and Schedule input and outputs of the block.

- **Data rate (bits/s):** The speed of the network.

- **Minimum frame size (bytes):** A message or frame shorter than this will be padded to give the minimum length.

- **Preprocessing delay (s):** The time a message is delayed by the network interface on the sending end. This can be used to model, e.g., a slow serial connection between the computer and the network interface.

- **Postprocessing delay (s):** The time a message is delayed by the network interface on the receiving end.

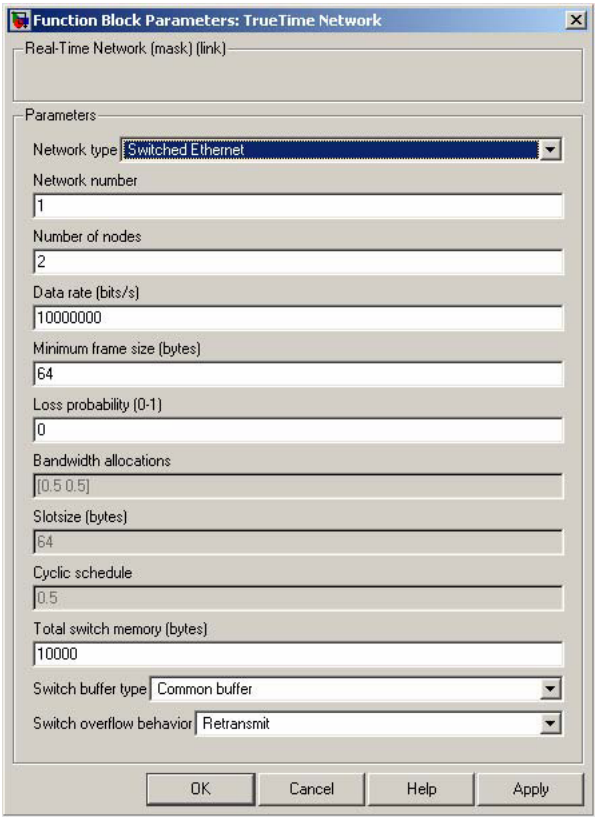

Figure A.2: TrueTime network block dialog

- **Loss probability (0-1):** The probability that a network message is lost during transmission. Lost messages will consume network bandwidth, but will never arrive at the destination.

Appendix B

DC Motor, Gear, and Encoder Parameters

B.1 Motor Data

Motor Type: RE30, Maxon, 60 Watt DC Motor.

B.1.1 Motor Parameters

Power rating	:	60 Watt
Nominal voltage	:	12 Volt
No load speed	:	7800 rpm
Stall torque	:	807 mNm
Armature resistance, R_a	:	0.207 Ω
Armature inductance, L_a	:	0.03 mH
Motor torque constant, K_t	:	13.9 mNm/A
Motor voltage constant, K_m	:	0.014 V.s/rad
Rotor inertia, J_m	:	$32.49 \times 10^{-7}\ Kgm^2$
Mechanical time constant, T_{th}	:	3 ms
Electrical time constant, τ_e	:	$\tau_e = \frac{L_a}{R_a}$
Motor efficiency, η_m	:	79.6 %

B.2 Gear Data

Gear Type: Planetary Gear head Maxon GP 32 C.

B.2.1 Gear Parameters

Maximum continuous torque	:	3 Nm
Reduction ratio	:	14:1
Mass moment of inertia	:	$0.8 \times 10^{-7}\ Kgm^2$
Gear efficiency, η_g	:	75 %
Number of stages	:	2
Gear head length, L_g	:	36.3 mm
Average backlash no load	:	0.8

B.3 Encoder Data

Encoder Type: Type *L* Digital MR Encoder with line drive

B.3.1 Encoder Parameters

Counts per turn	:	256
Number of channels	:	3
Max. Operating frequency	:	80 kHz
Supply voltage	:	5 ±5%
Output signals	:	TTL compatible
Moment inertia of code wheel	:	1.7×10^{-7} Kg
Output current per channel	:	max 5 mA

B.4 Beam Parameters

For the attached solid beam, the parameters are as follows:

Length of the beam, r	=	0.30 m
Mass of the beam, m	=	0.75 Kg

Moment of inertia of the beam is given as:

$$\begin{aligned} J_{beam} &= \frac{1}{3}mr^2 \\ &= \frac{1}{3}(0.75)(0.3)^2 \\ &= 0.3 \text{ Kg m}^2 \end{aligned} \tag{B.1}$$

The load moment of inertia is given as:

$$\begin{aligned} J_{load} &= J_{beam} + J_m \\ &= 0.3 + 32.49 \times 10^{-7} \\ &= 0.301 \text{ Kg m}^2 \end{aligned} \tag{B.2}$$

Bibliography

[1] Thomas B. Sheridan. Teleoperation, telepresence, and telerobotics: Research needs for space. *Human Factors in Automated and Robotic Space Systems: Proc. of a Symposium*, pages 279–291, 1987.

[2] Thomas B. Sheridan. Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, October 1993.

[3] William R. Ferrell. Delayed force feedback. *Human Factors*, 8:449–455, 1966.

[4] S. Hirche and M. Buss. Passive Position Controlled Telepresence System with Time Delay. 2003.

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

[6] S. Hirche and M. Buss. Haptic Internet-Telepresence. *Proc. of The Internet Challenge: Technology and Applications, Berlin, Germany*, pages 63–72, 2002.

[7] Alessandro Eusebi and Claudio Melchiorri. Force reflecting telemanipulators with time-delay: Stability analysis and control design. *IEEE Transactions on Robotics and Automation*, 14(4):635–640, 1998.

[8] Robert J. Anderson and Mark W. Spong. Asymptotic stability for force reflecting teleoperators with time delay. *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1618–1625, 1989.

[9] Mark W. Spong. Communication delay and control in telerobotics. *Journal of Japan Robotics Society*, 11(6):803–810, 1993.

[10] E. Fridman and U. Shaked. Delay-dependent stability and h_∞ control: constant and time-varying delays. *International Journal of Control*, 76(1):48–60, 2003.

[11] Richard J. Adams, Manuel R. Moreyra, and Blake Hannaford. Stability and performance of haptic displays: Theory and experiments. *Proc. ASME International Mechanical Engineering Congress and Exhibition*, pages 227–234, 1998.

[12] I. G. Polushin and H. J. Marquez. Stabilization of bilaterally controlled teleoperators with communcation delay: an *iss* approach. *Internation Journal of Control*, 76(8):858–870, 2003.

[13] Rogelio Lozano, Nikhil Chopra, and Mark W. Spong. Passivation of force reflecting bilateral teleoperators with time varying delay. *Proc. of Mechatronics'02, Entschede, Netherlands*, June 2002.

[14] Jose M. Azorin, Oscar Reinoso, Rafael Aracil, and Manuel Ferre. Control of teleoperators with communication time delay through state convergence. *Journal of Robotic Systems*, 21(4):167–182, March 2004.

[15] Gunter Niemeyer and Jean-Jacques E. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152–162, 1991.

[16] Silviu-Iulian Niculescu, Damia Taoutaou, and Rogelio Lozano. Bilateral teleoperation with communication delays. *International Journal of Robust and Nonlinear Control*, 13:873–883, 2003.

[17] Anas Fattouh and Olivier Sename. h_∞-based impedance control of teleoperation systems with time delay. *Proc. of IFAC workshop on Time-Delay Systems, INRIA, Rocquencourt, France*, September 2003.

[18] Guenter Niemeyer and Jean-Jacques E. Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, September 2004.

[19] Neville Hogan. Impedance control: An approach to manipulation, part i - theory. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–7, March 1985.

[20] Sandra Hirche and Martin Buss. Study of teleoperation using realtime communication network emulation. *Proc. of IEEE/ASME Internationa Conference on Advanced Intelligent Mechatronic AIM2003*, 2003.

[21] A. Frisoli, E. Sotgiu, C.A. Avizzano, D. Checcacci, and M. Bergamasco. Force-based impedance control of a haptic master system for teleoperation. *Sensor Review*, 24(1):42–50, 2004.

[22] G. Niemeyer and Jean-Jacques E. Slotine. Towards force-reflecting teleoperation over the internet. *Proc. of the 1998 IEEE International Conference on Robotics & Automation*, pages 1909–1915, 1998.

[23] Singha Leeraphan, Thavida Maneewan, and S. Laowattana. Stable adaptive bilateral control of tranleoperation through time-varying delay. *Proceeding of the International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland*, pages 2979–2984, September 30 - October 4 2002.

[24] Ned Mohan. *Electric Drives: An Integrative Approach*. MNPERE, 2004.

[25] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, Norway, 2002.

[26] Yasuyoshi Yokokohji and Tsuneo Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. *IEEE Transactions on Robotics and Automation*, 10(5):605–619, October 1994.

[27] Guenter Niemeyer and Jean-Jacques E. Slotine. Using wave variables for system analysis and robot control. *Proc. of the 1997 IEEE International Conference on Robotics & Automation*, pages 1619–1625, April 1997.

[28] Blake Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, February 2002.

[29] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stable teleoperation with time domain passivity control. *IEEE Transactions on Robotics and Automation*, 20(2):365–373, April 2004.

[30] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stability guaranteed control: Time domain passivity approach. *IEEE Transactions on Control Systems Technology*, 12:365–373, April 2004.

[31] Jee-Hwan Ryu, Carsten Preusche, Blake Hannaford, and Gerd Hirzinger. Time domain passivity control with reference energy following. *IEEE Transactions on Control Systems Technology*, In Press.

[32] Jee-Hwan Ryu, Carsten Preusche, Blake Hannaford, and Gerd Hirzinger. Time domain passivity control with reference energy following. *IEEE Transactions on Control System Technology*, 13(5):737–742, September 2005.

[33] Jaime Alvarez-Gallegos, Domingo Cortes, and M. W. Spong. A stable control scheme for teleoperators with time delay. *International Journal of Robotics and Automation*, 12(3):73–79, 1997.

[34] Liya Ni and David W.L. Wang. A gain-switching control scheme for position-error-based bilateral teleoperation: Contact stability analysis and controller design. *Internationl Journal of Robotics Research*, 23(3):255–274, 2004.

[35] Luis A. Montestruque and Pano Antsaklis. Stability of model-based networked control system with time-varying transmission times. *IEEE Transactions on Automatic Control*, 49(9):1562–1572, 2004.

[36] Imad Elhajj, Ning Xi, and Yun hui Liu. Real-time control of internet based teleoperation with force reflection. *Proc. of IEEE International Conference on Robotics and Automation*, pages 3284–3289, April 2000.

[37] M. C. Cavusoglu, A. Sherman, and F. Tendick. Bilateral controller design for telemanipulation in soft environments. *Proc. of IEEE International Conference on Robotics and Automation*, pages 1045–1052, May 2001.

[38] J.E. Colgate. Robust control of dynamically interacting systems. *ASME Journal of Dynamical Systems, Measurements and Control*, 116(3):419–428, 1994.

[39] Gene H. Hostetter, Clement J. Savant Jr., and Raymond T. Stefani. *Desing of Feedback Control Systems*. Holt, Rinehart and Wilson, 1982.

[40] William S. Levine, editor. *The Control Handbook*. CRC Press, 1996.

[41] Eduardo D. Sontag. On the input-to-state stability property. *European Journal of Control*, (1):24–36, 1995.

[42] El-Kebir Boukas and Zi-Kuan Liu. *Deterministic and Stochastic Time Delay Systems*. Birkhäuser, 2002.

[43] N. Hogan. Controlling impedance at the man/machine. *Proc. of the IEEE International Conference on Robotics and Automation, Arizona*, pages 1626–1631, 1989.

[44] Blake Hannaford and Jee-Hwan Ryu. Time domain passivity control of haptic interfaces. *Proc. of the 2001 IEEE International Conference on Robotics and Automation*, pages 1863–1869, May 2001.

[45] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1990.

[46] R. Sepulchre, M. Jankovic, and P.V. Kokotovic. *Constructive Nonlinear Control*. Springer, 1997.

[47] Henrik Flemmer. Aspects of using passivity in bilateral telemanipulation. *Technical Report Nr. TRITA - MMK 2004:16, Royal Insitute of Technology, Stockholm, Sweden*, 2004.

[48] Margaret L. McLaughlin, Joao P. Hespanha, and Gaurav S. Sukhatme. *Touch in Virtual Environments: Haptics and the Design of Interactive Systems*. Pearson Education, 2001.

[49] I.D. Landau, R. Lozano, and M.M'Saad. *Adaptive Control*. Springer, 1997.

[50] Lennart Ljung. *System Identification, Theory for the User*. Prentice Hall Information and System Sciences Series. Prentice Hall PTR, second edition, 1999.

[51] Carlo Benedetti, Matteo Franchini, and Paolo Fiorini. Stable tracking in variable time-delay teleoperation. *Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2252–2257, October 2001.

[52] Tissaphern Mirfakhrai and Shahram Payandeh. On using delay predictors in controlling force-reflecting teleoperation over the internet. *Robotica*, 23:809–813, 2005.

[53] Xiufen Ye, Max Q-H Meng, , Peter Xiaoping Liu, and Guobin Li. Statistical analysis and prediction of round trip delay for internet-based teleoperation. *Proc. of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2999–3003, oct 2002.

[54] Ming Yang, X. Rong Li, and Huimin Chen. Predicting internet end-to-end delay: An overview. *Proc. of INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 2815–2819, March 2005.

[55] Liming Wei and Deborah Estrin. The trade-offs of multicast tree and algorithms. *Proc. of International Conference on Computer Communications and Networks, ICCCN'94*, August 1994.

[56] N.M. Piratla, A.P. Jayasumana, and H. Smith. Overcoming the effects of correlation in packet delay measurements using inter-packet gaps. *Proc. of 12th IEEE International Conference on Networks (ICON 2004)*, pages 233–238, November 2004.

[57] Wendy L. Martinez and Angel R. Martinez. *Computational Statistics Handbook with Matlab*. Chapman and Hall/CRC, 2002.

[58] Jong-Phil Kim, Changhoon Seo, and Jeha Ryu. A multirate energy bounding algorithm for high fidelity stable haptic interaction control. *Proc. of The SICE-ICASE International Joint Conference 2006, Busan, Korea*, pages 215–220, October 2006.

[59] Thomas B. Sheridan. Human supervisory control of robot systems. *Proc. of IEEE International Conference of Robotics Automation, San Francisco*, April 1986.

[60] Martin Andersson, Dan Henriksson, Anton Cervin, and Karl-Erik Arzen. Simulation of wireless networked control systems. *Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*, December 2005.

[61] S.I. Niculescu and C. T. Abdallah. Delay effects on static output feedback stabilization. *Proc. IEEE Conference on Decision and Control*, pages 2811–2816, 2000.

[62] C. Abdallah, P. Dorate, and J. Benitez-Read. Delayed positive feedback can stabilize oscillatory systems. *Proc. of American Control Conference*, pages 3106–3107, 1993.

[63] Keqin Gu. An improved stability criterion for systems with distributed delays. *International Journal of Robust and Nonlinear Control*, (13):819–831, 2003.

[64] S. Evesque, A. M. Annaswamy, S. Niculescu, and A. P. Dowling. Adaptive control of a class of time-delay systems. *Transactions of ASME*, 125:186–193, 2003.

[65] Magdi S. Mahmoud. Uncertain jumping systems with strong and weak functional delays. *Automatica*, 40:501–510, 2004.

[66] Y. S. Lee, Y. S. Moon, W. H. Kwon, and P. G. Park. Delay-dependent robust h_∞ control for uncertain systems with a state-delay. *Automatica*, 40:65–72, 2004.

[67] Dong Yue. Robust stabilization of uncertain systems with unknown input delay. *Automatica*, 40:331–336, 2004.

[68] Chenggui Yuan and Xuerong Mao. Robust stability and controllability of stochastic differantial delay equations with markovian switching. *Automatica*, 40:343–354, 2004.

[69] Jean-Pierre Richard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39:1667–1694, 2003.

[70] N. Chopra, M. W. Spong, S. Hirche, and M. Buss. Bilateral Teleoperation over Internet: the Time Varying Delay Problem. *Proc. of the American Control Conference, Denver, CO*, 2003.

[71] Z. Hu, S. E. Salcudean, and P. D. Loewen. Robust controller design for teleoperation systems. *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 3:2127–2132, October 1995.

[72] Hu Shousong and Zhu Qixin. Stochastic optimal control and analysis of stability of networked control systems with long delays. *Automatica*, 39:1877–1884, 2003.

[73] Henrik Flemmer, Bengt Eriksson, and Jan Wikander. Passivity issues in bilateral teleoperation - a phase property. *Proc. of the 2003 Eurohaptics*, 1:228–241, July 2003 2002.

[74] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, pages 35–45, March 1960.

[75] Shahin Sirouspour and Ali Shahdi. Discrete-time linear quadratic gaussian control for teleoperation under communication time delay. *The International Journal of Robotics Research*, 25(2):187–201, February 2006.

[76] T. B. Sheridan. *Telerobotics, Automation and Human Supervisory Control*. MIT Press, 1992.

[77] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips. Stability of netowrked control systems. *IEEE Control Systems Magazine*, (1):84–99, February 2001.

[78] Asif Iqbal, Hubert Roth, and Moneeb Abu-Zaitoon. Stabilization of delayed teleoperation using predictive time-domain passivity control. *Proc. of the IASTED International Conference on Robotics and Applications RA2005*, pages 20–25, October 2005.

[79] Asif Iqbal and Hubert Roth. Predictive time domain passivity control for delayed teleoperation using energy derivatives. *Proc. of 9th International Conference on Control, Automation, Robotics and Vision, ICARCV 2006*, December 2006. Accepted for publication.

[80] H. Ohsaki, M. Murata, and H. Miyahara. Modeling end-to-end packet delay dynamics of the Internet using system identification. *17th International Teletraffic Congress*, December 2001.

[81] A.G. Parlos. Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors. *Proc. of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN'02*, pages 2460–2465, 2002.

[82] Tissaphern Mirfakhrai and Shahram Payandeh. A model for time-delays for teleoperaion over the internet. *Proc. of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 236–241, July 2001.

[83] P.F. Hokayem and M.W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 2006. to appear in Dec. issue.

[84] Asif Iqbal and Hubert Roth. Time domain passivity control for delayed teleoperation with parabolic power integration. *Proc. of THE 7th International Conference On Technical Informatics, CONTI 2006*, 1:85–90, June 2006.

Index