DD
2010
16

High Performance Haptic Teleoperation and Collaboration over Networks

August 2010

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering



Keio University
Graduate School of Science and Technology
School of Integrated Design Engineering

Tumerdem, Ugur

Acknowledgments

This dissertation is the outcome of the research I conducted at Ohnishi Laboratory, Keio University, since September 2005. I am indebted to all the past and current Ohnishi lab members that have assisted me in my endeavour. I would like to thank my supervisor, Professor Kouhei Ohnishi, for his continuous support, encouragement and guidance throughout this period. I am also grateful that he was always understanding and gracious. It is thanks to him that I have been able to complete this dissertation. I would also like to thank the members of my thesis committee; Associate Professor Takahiro Yakoh, Associate Professor Hiroaki Nishi and Professor Hiromitsu Ohmori, for the time they have dedicated and the feedback they have provided for the improvement of this dissertation. I would like to express my gratitude to Professor Toshiyuki Murakami and Assistant Professor Seiichiro Katsura who have provided me with valuable feedback and ideas throughout my research. I am also grateful to Dr. Tomoyuki Shimono at Yokohama National University, Assistant Professor Toshiaki Tsuji at Saitama University and Dr. Kenji Natori at Aoyama Gakuin University for their friendship and support. I would like to express my appreciation of the Global COE program of Keio University that has supported me and my research, and all the members of the program that have been so kind and helpful. I would also like to thank my friends Baris Yalcin, Cihan Acar and Caglar Kizilirmak with whom I shared so much during this period. Without their friendship it would have been impossible to come all this way.

Finally, I would like to thank my family, especially my mother, for believing in me and always supporting me in so many ways, even when I was so far away. They have always been the greatest source of strength and inspiration for me, and without them, I would not have been anywhere close to this point where I am today.

UGUR TUMERDEM

Keio University
August 2010

Contents

Acknowledgments ii

List of Figures vii

Chapter 1 Introduction 1

1.1 Networked motion control 2

1.1.1 Motion Control for Robotics 2

1.2 Networked Control Systems 3

1.2.1 Networked Motion Control 4

1.3 Haptic Teleoperation 5

1.3.1 Bilateral Teleoperation 6

1.3.2 Multilateral Teleoperation 8

1.4 Organization of the chapters 9

Chapter 2 Motion Control with Disturbance Observers 12

2.1 Disturbance Observers 13

2.2 Acceleration control 15

2.3 Position control 16

2.4 Force control 17

2.4.1 Reaction force observer 17

Chapter 3 Acceleration Consensus 20

3.1 Consensus Algorithms 21

3.1.1 Information Graphs 21

3.1.2 Consensus algorithms 23

3.1.3 Consensus Filters 25

3.1.4 Switching 27

3.2 Acceleration Consensus 28

3.2.1 Acceleration consensus with common inputs 32

3.2.2 Null space controller 34

3.2.3 Acceleration consensus under switching 36

3.2.4 Robust Acceleration Consensus 37

Chapter 4 Multilateral Teleoperation through Haptic Consensus 46

4.1 Multilateral Teleoperation 47

4.1.1 Haptic consensus with consensus filters 50

4.1.2 Haptic consensus on a flexible network 53

4.1.3 Haptic consensus with consensus filters 54

Chapter 5 Asymmetric Multilateral Teleoperation 58

5.1 Scaled Consensus 59

5.2 Scaled Acceleration Consensus 62

5.3 Asymmetric Multilateral Teleoperation 63

5.3.1 Position Scaling 65

5.3.2 Force Scaling 68

5.3.3 Multi Robot Micro-Macro Teleoperation 72

Chapter 6 Stable Bilateral Teleoperation under Time Delay 75

6.1 Four channel architecture and damping injection method 78

6.1.1 Four channel teleoperation 78

6.1.2 Augmented 4 channel architecture with damping injection . . 82
6.2 Stability analysis 83
6.2.1 L_2 Stability and small gain theorem 83
6.2.2 L_2 stability of four channel teleoperation 85
6.3 Transparency analysis 93
6.4 Experiment Results 100
6.5 Bilateral Teleoperation under Varying Time Delay 106

Chapter 7 Stable Multilateral Teleoperation under Time Delay 112
7.1 Multilateral teleoperation with damping injection 113
7.2 Small gain theorem and L_2 stability of multilateral teleoperation . . 117
7.2.1 L_2 stability of multilateral teleoperation 118
7.3 Experiment Results 126
7.4 Multilateral Teleoperation under Varying Time Delay 127

Chapter 8 Conclusions 133

Bibliography 136

Achievements 144

List of Figures

1.1 Thesis Layout 11

2.1 Actual robot model 13
2.2 Nominal robot model 14
2.3 Disturbance Observer 15
2.4 Acceleration tracking robot model 16
2.5 Position control using acceleration controller 17
2.6 Reaction force observer(RFOB) 19
2.7 Force control using acceleration controller 19

3.1 Information Graphs 22
3.2 Simulation of position consensus of four robots with different initial positions 24
3.3 Simulation of velocity consensus of four robots with different initial velocities 25
3.4 Acceleration consensus algorithm 29
3.5 Simulation results for acceleration consensus of four robots 31
3.6 Simulation results for acceleration consensus of four robots with a common acceleration input 33

3.7 Simulation results for acceleration consensus algorithm with formation center control 35
3.8 Simulation results for acceleration consensus under switching 36
3.9 Simulation results for robust acceleration consensus for 3 robots . . . 42
3.10 Simulation results for robust acceleration consensus with common input for 3 robots 43
3.11 Simulation results for robust acceleration consensus for 3 robots with sinusoidal input 44
3.12 Simulation results for robust acceleration consensus for 3 robots with 200ms time delay on each link 45

4.1 An acceleration consensus algorithm for haptic teleoperation 49
4.2 Haptic consensus with force consensus filters 52
4.3 Switching Network Structure 56
4.4 Consensus filter tracking in switching network structure 56
4.5 Haptic consensus on a 10kHz switching network with a switching consensus filter 56
4.6 Consensus filter tracking in switching network structure 57
4.7 Haptic consensus on a 10Hz switching network with a switching consensus filter 57

5.1 Weighted information graph 60
5.2 Position response of position scaled robots 67
5.3 Force response of position scaled robots 67
5.4 Position response of position and force scaled robots 70
5.5 Force response of position and force scaled robots 70
5.6 Scaled haptic consensus with force consensus filters 71
5.7 The experiment setup for micro-macro multilateral teleoperation . . 73

5.8 Position response of robots 74
5.9 Force response of robots 74

6.1 Four channel teleoperation architecture with disturbance and reaction force observer 81
6.2 Input output connected systems 84
6.3 Input output transformation of the four channel teleoperation architecture 87
6.4 The Minimum and the bandwith of transmitted impedances 95
6.5 Hybrid Matrix Responses for no delay 96
6.6 Hybrid Matrix Responses for 0.4s RT Delay 98
6.7 Hybrid Matrix Responses for 1s RT Delay 99
6.8 Hybrid Matrix Responses for 0.4s RT Delay with $C_f = b$ 100
6.9 Hybrid Matrix Responses for various architectures with damping injection under 0.4s RT delay 101
6.10 Bilateral teleoperation experiment setup consisting of two linear motors 102
6.11 Unstable four channel teleoperation with roundtrip time delay 400 ms 103
6.12 Experiment results with one way delay of 200ms from slave to master 104
6.13 Experiment results with 0.4 second roundtrip time delay 105
6.14 Experiment results with 1 second roundtrip time delay 106
6.15 Experiment results with one-way delay 100 + 100sin5000t[ms] variance 109
6.16 Experiment results with one-way delay 200 + 100sint[ms] variance . 110
6.17 Experiment results with one-way delay 200 + 100sin2000t[ms] variance 111

7.1 Strongly connected graph structure 114
7.2 Single robot block diagram and communication for multilateral teleoperation 116

7.3 Input output connected systems 118
7.4 Input output transformation of the multilateral teleoperation architecture 119
7.5 Multilateral experiment setup 127
7.6 Performance of damping injection based Multilateral teleoperation with no time delay 128
7.7 Unstable experiment results with one-way 100ms time delay 129
7.8 Performance of damping injection based Multilateral teleoperation with 200 ms delay between each robot 130
7.9 Stable experiment results with 0.5 second time delay between each robot 131
7.10 Stable experiment results with 200+100sin(2000(t))[ms] time delay between each robot 132

Chapter 1

Introduction

With the advents in wireless communication technology and rapid increases in computation power, it has become possible to perform many tasks ubiquitously, without the need to be actually present where the task is being performed. A typical example is that of teleconference, which involves the transmission of video and audio data on the network, enabling the tele-conferee to be virtually present in a remote environment and communicate and collaborate with other people as in real life. The goal of this dissertation is to present new methods to increase the immersion of the users in this virtual telepresence environment by introducing one more sense that we commonly make use of in our daily lives, that of touch. By using robots as communication devices, it is possible to realize remote sensing and manipulation of unknown environments. The robots can actually be utilized as the extension of the human arm and hand in a remote environment. Furthermore, by making use of multiple robots which are connected through networks, it is possible to realize collaborative telemanipulation by multiple human operators who share the sensation of touch. Such telepresence scenarios including the sense of touch or haptics, are already becoming reality. One of the application areas of such technologies is minimally invasive robotic surgery, where a surgeon can make operations remotely

with the use of telerobots. In such operations the haptic feedback would be vital as surgery requires precaution and the surgeon would need to feel the vital body parts, not to inflict any harm. The use of multiple robots in that case could be useful in surgical training or collaboration for increased safety. In this research, a framework for controlling multiple robots for high performance haptic teleoperation and collaboration over complex communication networks is proposed. Even video calls can be problematic when networks are not functioning effectively, introducing delay, loss of quality, and sometimes total loss of communication. In the case of haptic communication with telerobots, this would create an even bigger problem because of the dynamic nature of the robots that are making use of the network. In this dissertation a method is proposed which provides secure telepresence even when the communication network among the robots has a dynamically changing topology, circumventing the communication blackouts on the network. Also a method is proposed for safe haptic interaction by two or more operators on networks that introduce large time delays in the transmission of haptic data. These methods approach the problem from the perspective of motion control and networked control, or what is more specifically called in this dissertation as Networked Motion Control.

1.1 Networked motion control

1.1.1 Motion Control for Robotics

Motion control is a key research field that has been instrumental in creating robotic systems that can perform interactively with their environment. Robots are frequently used in industry for tasks such as positioning. This requires the accurate and precise control of the robot positions. Also some tasks such as assembly, milling or polishing require the robots to control the end effector forces as well. Furthermore, the robots have already started to find more applications in everyday life

among humans. And for the robots to safely interact with humans, it is desirable that the forces applied by the robots on their environment should be controlled. Therefore ideal motion control techniques should enable not only the position and speed control of robots, but also force control. With motion control it is possible to create robotic systems which are capable of assisting humans. Motion control techniques for robotics has been extensively researched during the 80's, and various methods such as computed torque controllers, feedback linearization methods and inverse dynamics methods have been proposed[1]. Acceleration control together with the use of disturbance observers[2][3] is a very simple and reliable method for attaining various motion control goals. Through acceleration control, it is possible to realize position, speed, force and compliance control with the robots.

1.2 Networked Control Systems

Research in networked control systems was cited as one of the major future directions in control in a panel discussing the future of the field[4]. And indeed, it has been receiving an immense amount of attention from control and robotics researchers around the world, and is currently seen as one of the most important research areas in robotics[5]. A special issue on Networked Control Systems of the IEEE Transactions on Automatic Control[6] gives the definition: "Networked control systems are control systems comprised of the system to be controlled and of actuators, sensors, and controllers, the operation of which is coordinated via a communication network. These systems are typically spatially distributed, may operate in an asynchronous manner, but have their operation coordinated to achieve desired overall objectives." The recent convergance of communication, computation and control due to the availability of faster and cheaper communication and computation technologies has broadened the horizons of the field of control and has given birth to the new era of networked control systems. Again it is argued in [6] that in this new era

emphasis in the analysis of control systems is shifting to "the explicit consideration of the interconnections; the network now must be considered explicitly as it affects significantly the dynamic behavior of the control system." Therefore the presence of the network should be taken into consideration when designing a control system with certain stability, performance and the reliability requirements. One of the recent developments enabling the analysis of the effects of network connection on the control system has been the use of graphs[7][8] and graph Laplacians[9]. Graphs and Laplacians have been extensively used in networked control systems and sensor networks with various network topologies running consensus algorithms[10]-[18]. Furthermore, effects of the network problems such as communication losses and time delays have been analyzed through the use of these methods[19][20]. It turns out that such methods are also useful in building networked motion controllers.

1.2.1 Networked Motion Control

A new generation of systems that are emerging as a by-product of the convergence in communication, computation and control technologies are the cyber-physical systems. These systems are designed as a network of interacting elements with physical input and output instead of standalone devices[21]. Motion controlled robotic systems are a good example of systems with physical inputs and physical outputs. They can create physical outputs such as force, and can react to environmental forces-inputs as well. The goal of this research is to extend the reach of the motion control system across the boundaries of the network such that the communication network acts as a backbone, making the overall robotic network a single control system. The kinds of networked control systems capable of realizing various motion control tasks will be called networked motion control systems. With networked motion control it is possible to realize various tasks by the coordinated motion of the robots. By using ideas from motion control and propogating force and position data of robots on the

network, it is possible achieve a coordinated networked robotic system, which can interact with humans and even permit humans to interact with each other. This can be realized through the integration of well known motion control techniques such as acceleration control with the networked control techniques such as consensus algorithms. In this dissertation this integration is realized through acceleration consensus algorithms and the networked motion control techniques are applied to a specific class of cyber physical systems; haptic teleoperation systems.

1.3 Haptic Teleoperation

Teleoperation means operating from distance and teleoperation systems are used for the manipulation of objects over a distance and can be realized with networked robots. These systems can be thought of as extensions of the humans body. Conventional teleoperation systems make use of two robots which are generally referred to as master and slave. The master robot is typically controlled by an operator and the slave robot copies the motion of the master. The word haptic on the other hand, comes from haptikos in Greek pertaining to the sense of touch. Haptics or haptics technologies refer to a very wide range of technologies that enable humans to perceive virtual or remote environments through the sense of touch. In the context of teleoperation, or haptic teleoperation, haptic refers to the perception of the environment the slave is in contact with, by the human through the use of force feedback. If the force at the slave side is reproduced with high quality at the master side this will be called high fidelity or transparent haptic teleoperation. In teleoperation literature a more specific name for haptic teleoperation using two robots is bilateral teleoperation but there are also teleoperation architectures involving multiple robots such as multilateral teleoperation which will also be introduced in this dissertation.

1.3.1 Bilateral Teleoperation

The first master-slave teleoperators were actually built completely mechanically in as early as the 1940s by Goertz. Later, electromechanical components took the place of the mechanical ones and teleoperation systems started to gain popularity. In the 1960's, the importance of the issue of time delay between the master and slave systems started to become apparent[22]. Until 1988, the problem of instability due to time delays in bilateral teleoperation systems was not solved, and this prevented a breakthrough in the field. In 1988, scattering variables, commonly used in electrical circuits was introduced into bilateral teleoperation, and succesful bilateral teleoperation under time delay was realized[23]. This was followed by a plethora research concetrating on the passivity approach presented in this result, and was further developed into the wave variable method[24]. Meanwhile, another important issue in bilateral teleoperation was recognized; the issue of performance. It was shown in [25] that, the performance goal of a teleoperation system should be transparency, meaning that the robotic system between the user and the environment should be transparent to the user, permitting the operator to feel and manipulate the environment perfectly. For quantifying and measuring transparency, hybrid matrices, also used in electrical circuits, were proposed. Following the introduction of hybrid matrices, it became easier for researchers to develop controllers with high performance, and in 1992, Yokokohji[26] and Lawrence[27], proposed four channel controllers which make use of all the force and position data for perfectly transparent teleoperation systems. Meanwhile it was shown in [26] that, transparency and the stability are two conflicting goals. In order to guarantee stability there should be a sacrifice in the transparency of the system. Indeed, the passivity based stability guaranteeing approaches of [23] and [24] could not satisfy the transparency goals given in [25], and the transparency based approaches of [26] and [27] made the system unstable in the presence of time delay. Until recently, these two ap-

proaches have created two different schools of thinking for bilateral teleoperation, where one aims to guarantee stability of nonlinear systems, such as [28] and [29], by using damping injection, without much regard for transparency. The other seeks to achieve highly transparent systems with linearized robot models like in [30] and [31]. In [30], a highly transparent system with acceleration controllers and disturbance observers was realized without the use of force sensors and in [31] three-channel transparent systems under delay were proposed without guaranteeing the stability for all cases. Other approaches to find the holy grail, or transparency under time delay were also attempted using linearized models, and disturbance observers[32][33], which provide better transparency under time delay without guaranteeing the stability. In this research both approaches have been utilized, while accepting that there should be some trade-off in performance and stability due to delay, both high transparency and stability is aimed. It is shown by using damping injection methods in four-channel based linearized systems, that it is possible to realize a delay independent stable, and high transparency system with good position tracking and high frequency force feedback. Therefore the system that is proposed in this dissertation can provide high performance teleoperation even on unreliable networks. One of the main applications of high transparency bilateral systems would be in minimally invasive robotic surgery[34], which requires accuracy, precision as well as safety. Therefore, proposed methods under time delay would satisfy the necessary safety and performance requirements by such operations. Another issue in precision robotic surgery though, is the scaling of motion of the surgeon for increased precision, and considerable effort has been given to realize what is called scaled teleoperation or micro-macro teleoperation[35]. In this research such systems, made up of not only two robots but multiple robots on networks with changing topologies, are developed using the acceleration consensus framework.

1.3.2 Multilateral Teleoperation

The survey paper on teleoperation [22] mentions that research of n-to-m master slave systems will be the next step to take in the field. It also mentions that for these kind of systems, there are issues that should be considered; such as communication(fully connected-relay based) and control(centralized-decentralized) architectures to be used. This research goal obviously coincides with the recent interest in networked control systems composed of multiple robots. The advancing technology has permitted us to create networked multiple telerobot systems which can connect multiple operators. First results regarding this kind of multi-operator multi-robot teleoperation were presented in [36]-[40], with the extension of the four channel architecture based on acceleration controllers, to multi robots. This kind of teleoperation is called multilateral teleoperation and enables the reflection of forces among all the robots, and perfect position tracking. A multilateral system can be used for haptic training of operators or haptic collaborative teleoperation. Some results for cooperative multi-robot teleoperation under time delay without considering transparency were presented in [41][42]. In this research, acceleration consensus algorithms and force consensus filters are proposed to realize multilateral teleoperation on networks with dynamically changing topologies, without losing transparency. The proposed method provides an effective decentralized solution to the control and communication problems mentioned in [23]. Due to the flexibility of the consensus algorithms, it is possible to employ this method under various network topologies with strict constraints. Also, as mentioned, consensus algorithms can be utilized to realize asymmetric multilateral teleoperation which can enable the operators to have various levels of dominance of the overall system, or control smaller-bigger robots. Furthermore, methods to realize multilateral teleoperation with high transparency under time delay have not been developed until now. In this dissertation, the proposed damping injection based four channel teleoperation architecture is extended

to the multilateral case with the help of acceleration control and a high performance multilateral teleoperation system which is robust against time delays is designed.

1.4 Organization of the chapters

The organization and contents of this dissertation can be seen on figure 1.1. In chapter 2 robust acceleration control using disturbance observers, is introduced. Reaction force observer which is a specialized disturbance observer used to sense forces applied on robots by operators and environment is also introduced and its application to force control is shown. In chapter 3, firstly information graphs and consensus algorithms based on graph Laplacians are described. The properties of first and second order consensus algorithms as well as the high pass consensus filters for averaging signals is discussed. The stability of consensus algorithms under switching topology is also mentioned. Following this an acceleration consensus algorithm is synthesized, based on the 2nd order consensus algorithms and consensus filters as well as the robust acceleration controllers presented in chapter 2. The behavior of the acceleration consensus algorithms under local and common inputs is analyzed and a null space controller based on the properties of graphs is introduced. Robustness of acceleration consensus algorithms under switching network structure shown by simulations. Furthermore, an acceleration consensus algorithm is introduced using local velocity feedback which can perform robustly under time delay. In chapter 4, it is shown that haptic teleoperation can be considered as a consensus problem. Acceleration consensus is used for realizing multilateral teleoperation. Force consensus filters are introduced for estimating the average of the forces on a network. Haptic consensus on networks with switching network topologies is shown, demonstrating the robustness of the algorithm. Experimental results for haptic consensus in a multilateral teleoperation system, consisting of three 1-DOF linear robots connected with various network structures as well as switching

topology are shown and the validity of the proposed framework is established. In chapter 5, the haptic consensus algorithm for scaled teleoperation and micro-macro teleoperation with multiple robots, which is called asymmetric multilateral teleoperation, is proposed. In this chapter, first a new scaled consensus algorithm-filter that can enable the nodes to track functions of inputs, based on the weights of weighted graphs is introduced. By making use of these consensus algorithms, scaled haptic consensus algorithms are obtained which enable the scaling of the inputs of operators and the robots sizes, meanwhile guaranteeing stable and transparent multilateral teleoperation. Experiment result are provided that confirm the validity of the proposed method. In chapter 6, the focus shifts to the problem of time delay on the networks, and the issue of stability and transparency in bilateral teleoperation. It is shown that it is possible to stabilize and have high performance teleoperation system using the four channel architecture by augmenting it with local damping injection. This idea is the same with the robust acceleration consensus algorithm introduced in chapter 3. Delay independent L_2 stability of the proposed system is shown and transparency analysis is also provided. Furthermore, it is shown with experiments using two 1-DOF robots that the proposed method has good performance under constant and varying time delay scenarios. In chapter 7, a form of the acceleration consensus algorithm proposed in chapter 3 along with damping injection is employed, to extend the ideas in chapter 6 to multilateral teleoperation. Again, delay independent L_2 stability of multilateral teleoperation based on robust acceleration consensus is shown, and the sufficient conditions to guarantee the stability are provided. Furthermore the validity of the proposed method under constant and varying time delays is shown through experiments. In chapter 8, the results are summarized and the dissertation is concluded.

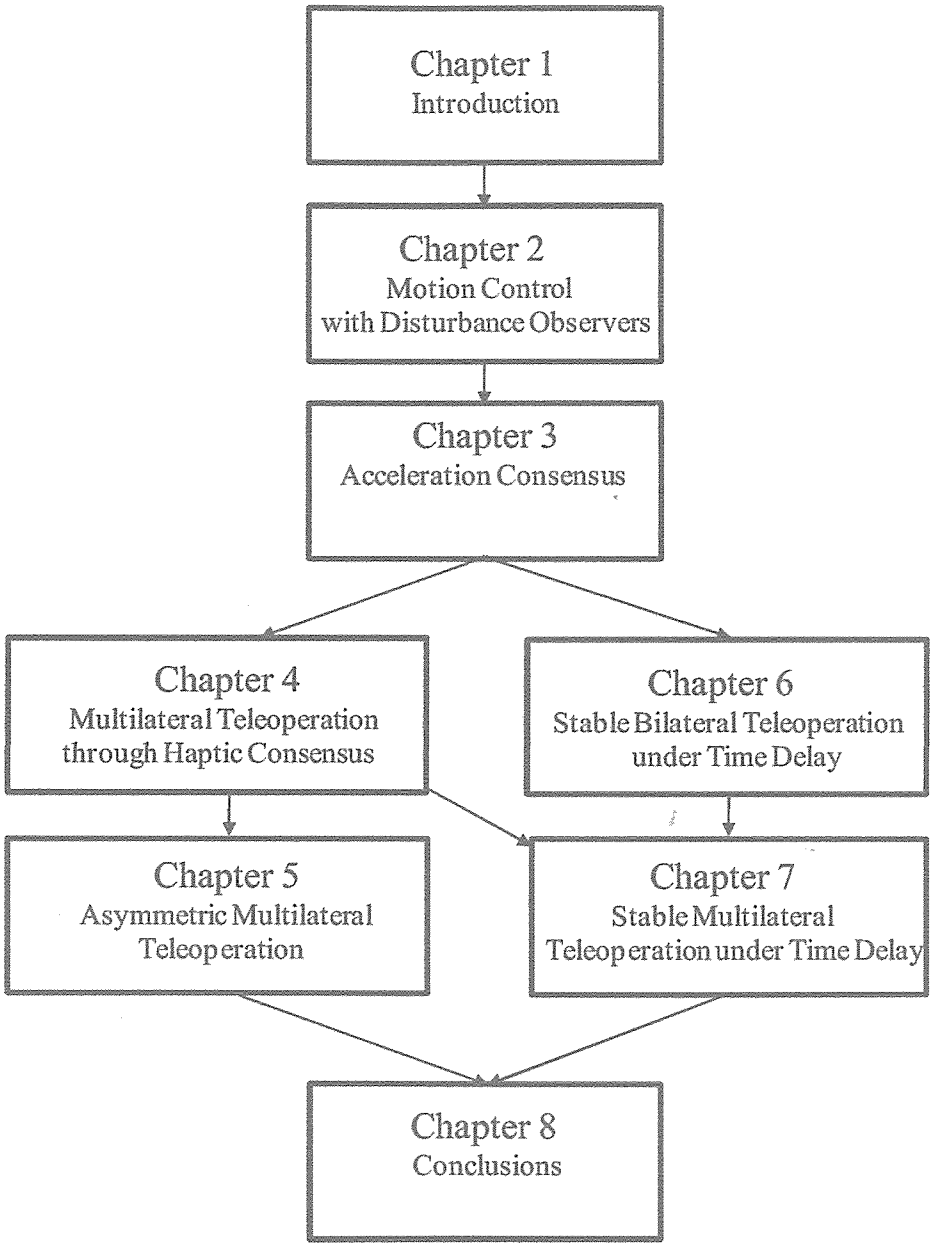


Figure 1.1: Thesis Layout

Chapter 2

Motion Control with Disturbance Observers

Motion control is an important discipline which has been extensively researched and utilized in developing complex systems such as robotic systems and automation equipment used in various environments; especially industry. The motion of the robots can be controlled using various actuators, however in this research the focus will be the control of robotic systems actuated by electric motors such as direct drive motors. Direct drive motors are very practical for precise motion control, due to the fact that they directly move their payload without any transmission mechanisms, which reduce losses due to friction, backlash and the like tremendously. Also, direct drive motors can produce high torques which can enable fast motion control. However there are certain problems with direct drive motors; they are more sensitive to variations in the load inertia, motor properties, etc... These variations are disturbances and must be eliminated for precise motion control. Ohnishi et al. proposed the disturbance observer[2] in order to estimate and supress the disturbances created by these variations. Once the disturbances are cancelled, motion control using electric motors can be realized through acceleration control[3]. Using

acceleration control, it is possible to realize position and force control which are two important goals of motion control. In this research networked motion control through networked acceleration control is realized. Therefore, first it necessary to explain how acceleration control with disturbance observers can be realized.

2.1 Disturbance Observers

In this chapter, robot modeling is based on the disturbance observers[2][3][43]. In this framework the robot is modeled in joint space as a load connected to the motor, and forces due to interaction between the joints such as coriolis or gravity as well as some nonlinearities such as friction are considered as disturbances acting on the motor. In this research 1DOF robots composed of linear direct drive motors are used and the manipulator is a simple load connected to the shaft of these motors. This means that there will be negligible interactive disturbance on the system. The model of the 1DOF robot can be seen in figure 2.1. In the figure, I_a^{ref} is the input current, $x, \dot{x}$ are the position, velocity responses of the motor respectively, M is the motor inertia, K_t is the actual motor constant, F_{int} is the interactive disturbance force including the coupling force from the manipulator, gravitation force, etc. F_{ext} is the external disturbance input to the system such as reactive force during contact. There are also friction forces acting on the system: f is the coulomb friction and $D\dot{x}$ is the viscous friction where D is the inherent damping in the robot. This research

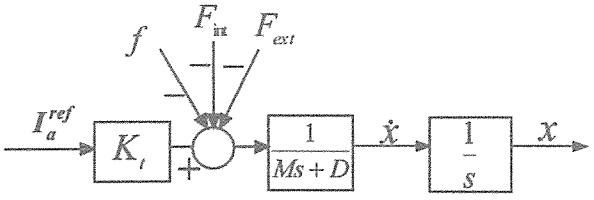


Figure 2.1: Actual robot model

make use of disturbance observers to compensate the effects of the disturbance acting on the robot. A disturbance observer makes use of a nominal model of a system as in figure 2.2. In the figure M_n is the nominal motor-load inertia, and K_{tn}

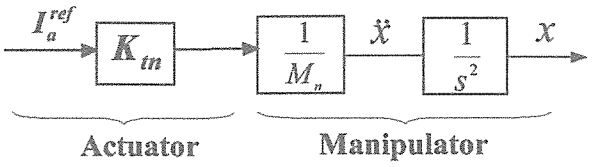


Figure 2.2: Nominal robot model

is the nominal motor(thrust) constant and $\ddot{x}$ the acceleration. The motor and the model are given the same control inputs such as current, and the difference between the response(e.g velocity) of the robot and the model is filtered. This difference is the disturbance acting on the system which was unaccounted for, in the model. Disturbance observer, which can be seen in figure 2.3, is a low pass filter with the cutoff frequency g_{dis}, which can estimate the total disturbance applied on the system by comparing the input and the system response. The total disturbance force acting on such a system can be written with various components[43] as:

$$\begin{aligned} F_{dis} &= F_l + \Delta M\ddot{x} - \Delta K_t I_a^{ref} \\ &= F_{int} + F_{ext} + f + D\dot{x} \\ &+ (M - M_n)\ddot{x} + (K_{tn} - K_t)I_a^{ref} \end{aligned} \tag{2.1}$$

where $(M - M_n)\ddot{x}$ shows the disturbance force created by the variation in the mass, and $(K_{tn} - K_t)I_a^{ref}$ is the disturbance force created by the force pulsation due to flux distribution variation of the motor. The disturbance estimate can be obtained

by the disturbance observer as:

$$\hat{F}_{dis} = \frac{g_{dis}}{s+g_{dis}}(I_a^{ref}K_{tn} + g_{dis}M_n\dot{x}) - g_{dis}M_n\dot{x} \tag{2.2}$$

$$\hat{F}_{dis} = \frac{g_{dis}}{s+g_{dis}}F_{dis} \tag{2.3}$$

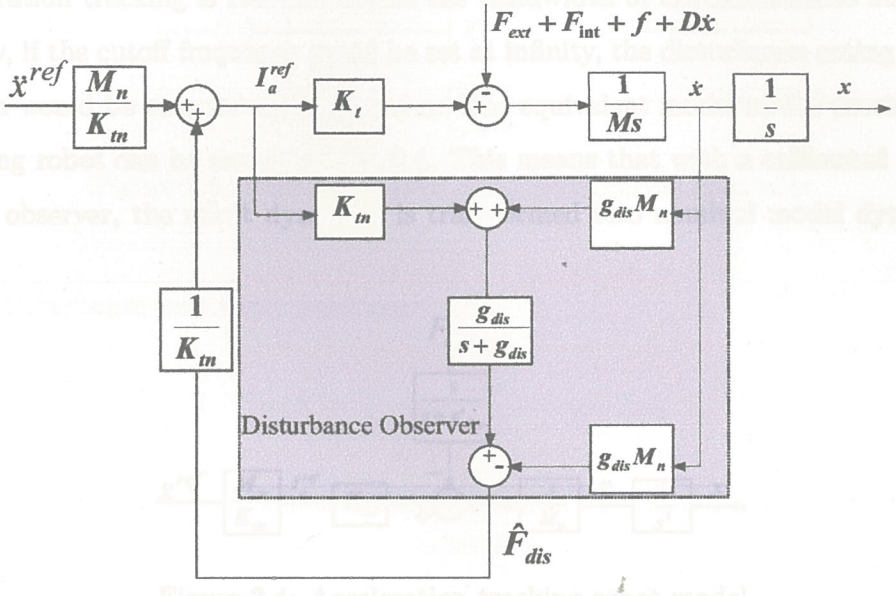


Figure 2.3: Disturbance Observer

2.2 Acceleration control

When this disturbance estimate is fedback to the system for compensation, it is possible to obtain acceleration tracking with the error:

$$\ddot{x}^{ref} - \ddot{x} = \frac{G(s)}{M_n}F_{dis} \tag{2.4}$$

Here, $G(s)$ is the disturbance sensitivity function[35]:

$$G(s) = \frac{s}{s + g'_{dis}} \tag{2.5}$$

where $g'_{dis} = \frac{K_t}{K_{tn}} g_{dis}$ if complete disturbance compensation is realised from the velocity measurement. The fact that $G(s)$ becomes a highpass filter means that the acceleration tracking is realized within the bandwidth of the disturbance observer. Ideally, if the cutoff frequency could be set at infinity, the disturbance acting on the system would be zero for all frequencies. The equivalent model of the acceleration tracking robot can be seen in figure 2.4. This means that with a calibrated disturbance observer, the robot dynamics is transformed into nominal model dynamics.

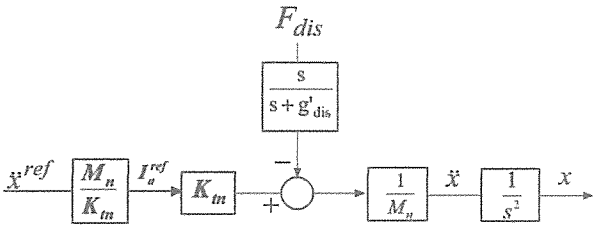


Figure 2.4: Acceleration tracking robot model

2.3 Position control

It is also possible to realize position control with an acceleration controller as can be seen on figure 2.5: Where the acceleration command value $\ddot{x}^{cmd}$ is determined from the trajectory generation algorithm and is the second derivative of the position trajectory. It is possible to write the position error with this acceleration controller

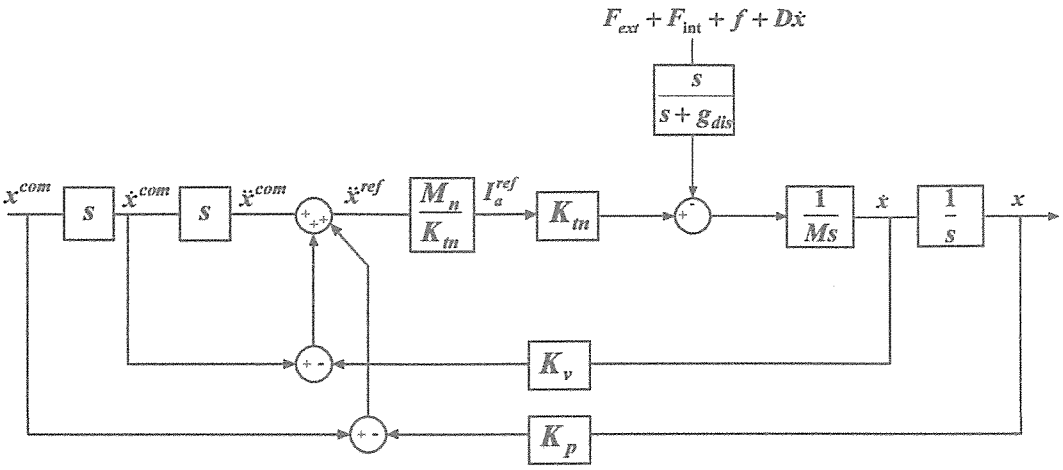


Figure 2.5: Position control using acceleration controller

using disturbance sensitivity functions:

$$X^{cmd} - X = \frac{G(s)}{M_n(s^2 + sK_v + K_p)} F_{dis} \tag{2.6}$$

Again if the disturbance compensation is succesful, the position error becomes zero.

2.4 Force control

Force control can also be realized using acceleration controllers. However first the force applied on the system should be measured. This can be realized by force reaction observers.

2.4.1 Reaction force observer

The force input to the system by an operator or the environment is a part of the disturbance estimated by disturbance observers and can be more precisely determined with a reaction force observer(RFOB)[43], shown in figure 2.6 and described

by the following equation:

$$\begin{aligned}\hat{F}_{ext} &= \frac{g_{reac}}{s+g_{reac}}(I_a^{ref}K_{tn} + g_{reac}M_n\dot{x} \\ &\quad -F_{int} - f - D\dot{x}) - g_{reac}M_n\dot{x} \end{aligned} \tag{2.7}$$

where g_{reac} is the cutoff frequency of the reaction force observer and the rest of the terms are the same as a disturbance observer. Basically, reaction force observer estimates the external force term F_{ext} of the disturbance. Assuming that the motor and load models are calibrated, by using constant velocity tests, it is possible to obtain the friction terms and interactive disturbance terms such as gravity acting on the robot during motion[43]. By substracting these values from disturbance estimate, it is possible to extract the external force which can be an environmental or operator force. It is possible to determine the bandwidth of the force sensing by choosing the g_{reac}. It has been shown in [44] that using RFOBs, it is possible to realize higher bandwith force sensing compared to force sensors. For force control, the estimated force from the reaction force observer becomes the acceleration command value of the acceleration controller. The force controller structure can be seen in figure 2.7.

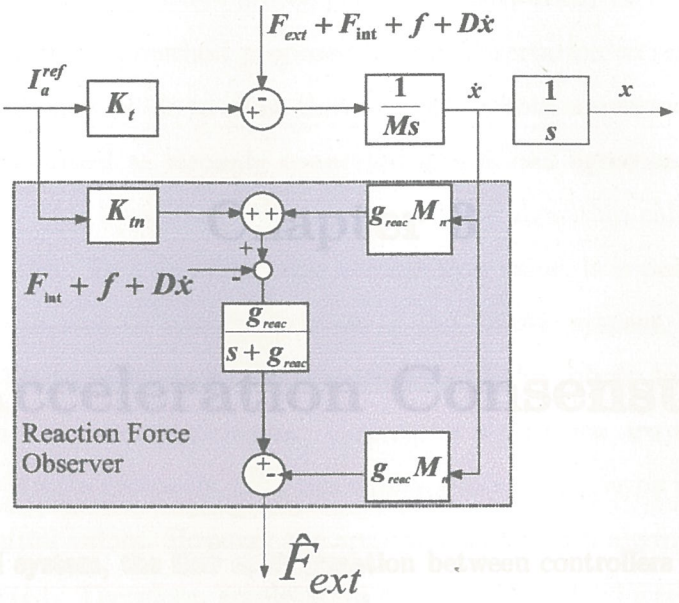


Figure 2.6: Reaction force observer(RFOB)

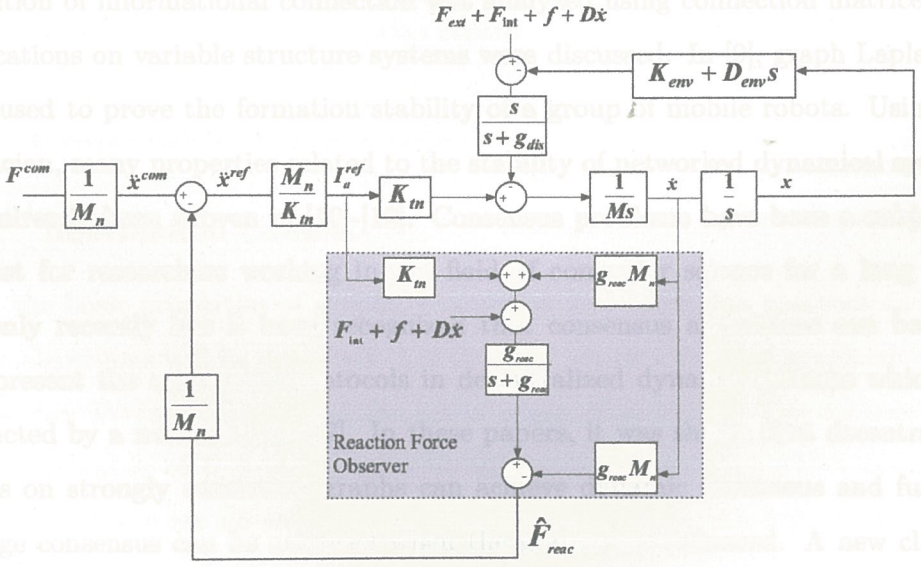


Figure 2.7: Force control using acceleration controller

Chapter 3

Acceleration Consensus

In a networked system, the flow of information between controllers such as system states is of great importance in realizing stable coordinated motion. Recently, directed graphs have turned out to be rather useful for the representation of network topology and information flow in dynamical systems [7]-[19]. In [7][8], dynamic transition of informational connection was analyzed using connection matrices and applications on variable structure systems were discussed. In [9], graph Laplacians were used to prove the formation stability of a group of mobile robots. Using the Laplacian, many properties related to the stability of networked dynamical systems have already been proven in [10]-[12]. Consensus problems have been a subject of interest for researchers working in the field of computer science for a long time, but only recently has it been recognized that consensus algorithms can be used to represent the agreement protocols in decentralized dynamic systems which are connected by a network [10]-[17]. In these papers, it was shown that decentralized agents on strongly connected graphs can achieve dynamic consensus and further, average consensus can be achieved when the graphs are balanced. A new class of distributed consensus algorithms have been proposed in [13]-[17] which can filter sensor measurements and agree on the average of the measurements. In this research

consensus algorihtms and filters are employed for networked motion control. Acceleration consensus is a method proposed in this dissertation to realize networked motion control. It is possible to show that multiple robots connected by a network, which can be described as strongly connected graph, can agree on a common acceleration value. This is realized by a novel consensus algorithm called acceleration consensus algorithm. By controlling this acceleration value, it is possible to control the overall coordinated or common motion of the robotic system. What is more, it is also possible to control the relative motions of the robots by the Laplacian dynamics as in consensus algorithms. Consensus algorithms are very interesting that, with very simple protocols, they can enable robots to agree on a meeting point based on the initial values. However the applications for such algorithms in motion control are limited. Therefore, acceleration consensus is introduced and it enables advanced motion control tasks with a network of robots. Acceleration consensus literally refers to the agreement of robots on their acceleration values. As acceleration control is the key technology for advanced motion control, acceleration consensus will be employed for networked motion control.

3.1 Consensus Algorithms

3.1.1 Information Graphs

First, the basic properties of graphs, connection and Laplacian matrices and consensus algorithms will be described. Let $\mathcal{G}$=$(\mathcal{V}, \mathcal{E}, \mathcal{A})$ be a directed graph with a set of n nodes; $\mathcal{V}$=$\{v_1, \ldots, v_n\}$, and the set of edges; $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The connection or adjacency matrix of $\mathcal{G}$ is $\mathcal{A} = [a_{ij}]$ with adjacency elements a_{ij} which are always positive. If there are no weights, the elements are binary. An edge is defined as e_{ij}=(v_i, v_j). If $e_{ij} \in \mathcal{E}$ then $a_{ij} > 0$ otherwise $a_{ij} = 0$. The neighborhood of a node v_i is N_i=$\{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. The degree matrix Δ=$[\delta_{ij}]$ is diagonal, with diago-

nal elements δ_{ii}=deg$_{out}(v_i)$ and all other elements 0. For instance, the information graphs seen in figure 3.1 have the connection and degree matrices :

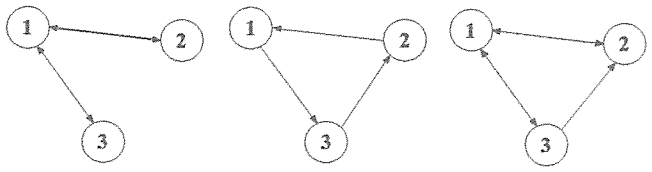


Figure 3.1: Information Graphs

$$A_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; A_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix},$$

$$\Delta_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; \Delta_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; \Delta_3 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

and

$$L_1 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \; L_2 = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \; L_3 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & -1 & 2 \end{bmatrix}$$

It is possible to obtain the Laplacian matrices L by substracting connection matrices from the degree matrices; $L = \Delta - A$. All these graphs are strongly connected which means that there is a path from any arbitrary node to the other. Also the first two graphs are balanced; every node has an outdegree equal to the indegree. However the third graph is not balanced because the second node has one more input than the output. It will be shown that properties such as being strongly connected and

balanced are extremely important in consensus algorithms.

3.1.2 Consensus algorithms

A Laplacian consensus algorithm is represented by a differential equation:

$$\dot{x}_i = \sum_{v_j \in N_i} a_{ij}(x_j - x_i) \tag{3.1}$$

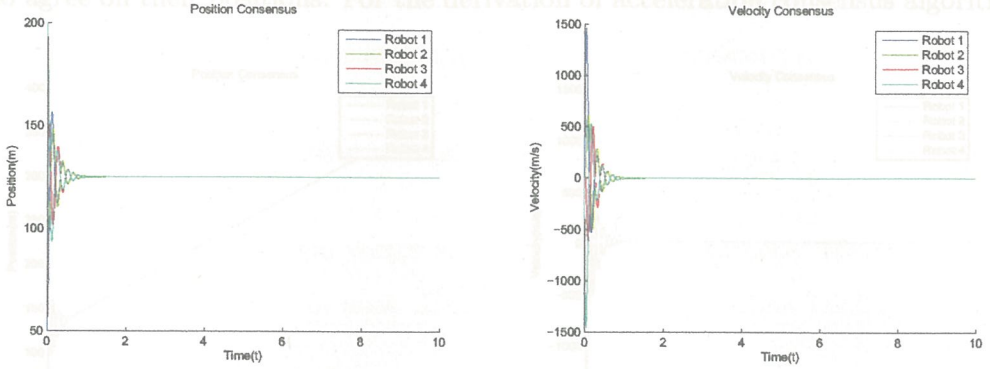
$$\dot{x} = -Lx \tag{3.2}$$

$$x(0) = z_0 \tag{3.3}$$

Where x_i are the values of each node. It was demonstrated in [11] that above defined algorithm globally asymptotically solves a consensus problem, given that the information graph A is strongly connected. If A is strongly connected, then $\text{rank}(L) = n - 1$ and one eigenvalue of L is always zero. Because L is positive semi-definite, all the nonzero eigenvalues of -L are negative and this quarantees that the system is stable. Also, in equilibrium all the nodes converge to the same value which is a right eigenvector of L corresponding to the 0 eigenvalue. This is called the consensus value. However in order to reach average consensus the graph must be balanced meaning that $\deg_{out}(v_i)$=$\deg_{in}(v_i)$. If the graph is balanced then every node converges to the average of their initial value:

$$x(i,t) \rightarrow \frac{1}{N} \sum_{i \in G} x(i,0) \tag{3.4}$$

Figure 3.2 shows the simulation results for this 1st order consensus algoritm, it can be seen that the positions of 4 robots converge to the average of their initial positions and their velocities converge to 0. If the consensus algorithm is second



(a) position consensus (b) velocity consensus

Figure 3.2: Simulation of position consensus of four robots with different initial positions

order as shown in [18]:

$$\ddot{x}_i = \sum_{v_j \in N_i} K_{p_{ij}}(x_j - x_i) + \sum_{v_j \in N_i} K_{v_{ij}}(\dot{x}_j - \dot{x}_i) \tag{3.5}$$

$$\ddot{x} = -K_p L x - K_v L \dot{x} \tag{3.6}$$

$$\dot{x}(0) = \dot{z}(0) \tag{3.7}$$

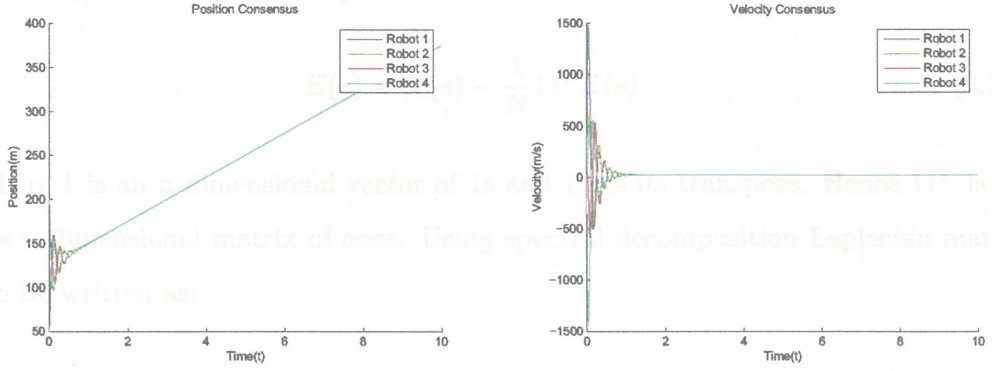
Such that the x represents position, $\dot{x}$ the velocity and $\ddot{x}$ the acceleration, it can be shown that if the initial velocities are different, the nodes will converge to the average of the velocities as well as having the same positions:

$$\dot{x}(i,t) \rightarrow \frac{1}{N} \sum_{i \in N_i} \dot{x}(i,0) \tag{3.8}$$

$$x_1 = x_2 = \ldots = x_N \tag{3.9}$$

Figure 3.3 shows the simulation results for the second order consensus algorithm. It can be seen that when the initial velocities are different, robots agree on the average of their initial velocities and move with the same speed. Furthermore, they

also agree on their positions. For the derivation of acceleration consensus algorithm



(a) position consensus (b) velocity consensus

Figure 3.3: Simulation of velocity consensus of four robots with different initial velocities

properties of the second order consensus algorithms will be used.

3.1.3 Consensus Filters

Consensus filters are basically modified consensus algorithms. It was shown that consensus filters can average the input signals applied to every node on a connected graph. A high-pass consensus filter was proposed by Spanos in [15][16], with the addition of local input signals $z_i(t)$ to the basic consensus algorithm:

$$\dot{x} = -Lx + \dot{z} \tag{3.10}$$

$$x(0) = z(0) \tag{3.11}$$

Note that the derivatives of the input are added to the system. The MIMO transfer function of this form can be written as:

$$H_{xz} = s(sI + L)^{-1} \tag{3.12}$$

If an error function as the difference between the states $x_i(s)$ and the average of the inputs $z_i(s)$ is assumed, it is possible to write an error transfer function as:

$$E(s) = X(s) - \frac{1}{N}11^T Z(s) \tag{3.13}$$

Where 1 is an n dimensional vector of 1s and 1^T is its transpose. Hence 11^T is an $n \times n$ dimensional matrix of ones. Using spectral decomposition Laplacian matrix can be written as:

$$L = \sum_i \lambda_i P_i \tag{3.14}$$

where λ_i are the eigenvalues of the Laplacian and P_i are the orthogonal projections on the orthogonal eigenspaces. The Laplacian has the following properties if the graph structure is connected:

$$(1)\ \lambda_1 = 0 \tag{3.15}$$

$$(2)\ P_1 = \frac{1}{N}11^T \tag{3.16}$$

$$(3)\ \lambda_i > 0 \text{ for all } i > 1 \tag{3.17}$$

So the error function can be rewritten as:

$$\begin{aligned} H_{eu} &= \frac{1}{N}11^T - \left(\frac{1}{N}11^T + \sum_{i>1}\frac{s}{s+\lambda_i}P_i\right) \\ &= -\sum_{i>1}\frac{s}{s+\lambda_i}P_i \end{aligned} \tag{3.18}$$

Because all the eigenvalues are positive, it is guaranteed that the error will converge to 0. This means that the filter states track the average of the inputs:

$$x(i,t) \rightarrow \frac{1}{N}11^T z \tag{3.19}$$

It should be noted that in this derivation, the derivatives of the signal are added as the inputs, such that the filter states can track the original input. So, this means that the inputs are first filtered by some predictor, such that the consensus filter can estimate the average of the inputs. This filter is very useful for the distributed averaging of signals on a network. Basically, the inputs are applied locally and filter states are transmitted on the network according to the graph topology. If the graph is connected, every node will agree on the average of their inputs, without actually having to know the inputs of every other node.

3.1.4 Switching

It has been shown in previous works[11] that consensus algorithms are stable under switching conditions, given that they are strongly connected and balanced at every time. First a disagreement function is defined as the difference between every state and the average of the states:

$$\delta = x - \frac{1}{N} 1 1^T x \tag{3.20}$$

$$\dot{\delta} = -L\delta(t) \tag{3.21}$$

where λ_2 is the second smallest eigenvalue, as the smallest eigenvalue is always zero. If the network topology is switching such that $G(t) = (V, E(t))$ has a time varying edge structure, $\|\delta\|^2$ is a good Lyapunov function candidate:

$$\phi = \|\delta\|^2 \tag{3.22}$$

$$\dot{\phi} = -2\delta^T L(G(t))\delta \leq -2\lambda_2 \|\delta\|^2 = -2\lambda_2 \phi \tag{3.23}$$

This means that the Lyapunov function exponentially vanishes with a speed of at least λ_2, which changes according to every edge structure. However given that the switching graphs are strongly connected and balanced, it is guaranteed that the λ_2

is always positive, which means that the error will converge to zero under switching topology.

3.2 Acceleration Consensus

For acceleration consensus, a new algorithm which is similar to the PD acceleration controller that is typically used in bilateral teleoperation [30] is defined. A similar 2nd order consensus algorithm was proposed in [18] to represent the flocking dynamics of inertial agents which we introduced previously. However that algorithm enables robots to agree on their initial velocities only. For the acceleration consensus servo inputs u are introduced to this 2nd order algorithm, so that the algorithm behaves like a consensus filter. This algorithm is called the acceleration consensus algorithm. By using the input, the common acceleration that every robot will agree on any strongly connected network can be determined. Agreement on this common acceleration is called acceleration consensus. Exploiting acceleration consensus, it becomes possible to realize very complicated motion control tasks. A block diagram portraying the acceleration consensus algorithm can be seen in figure 3.4. It can be seen that the position and velocity references are determined by the network connection; and is the sum of the position and velocity values of the neighboring nodes. Then every node computes a local control based on the feedback of its own states multiplied by its degree number. This local control represents the Laplacian dynamics. Furthermore it can be seen in the figure that acceleration commands are added locally as u_i. These acceleration commands are employed for acceleration consensus. First, the following servo consensus algorithm where K_p and K_v are positive graph weights and u_i are the local servo inputs, is defined:

$$\ddot{x}_i = \sum_{j \in N_i} K_p(x_j - x_i) + \sum_{j \in N_i} K_v(\dot{x}_j - \dot{x}_i) + u_i \tag{3.24}$$

$$\ddot{x} = -K_p L x - K_v L \dot{x} + u \tag{3.25}$$

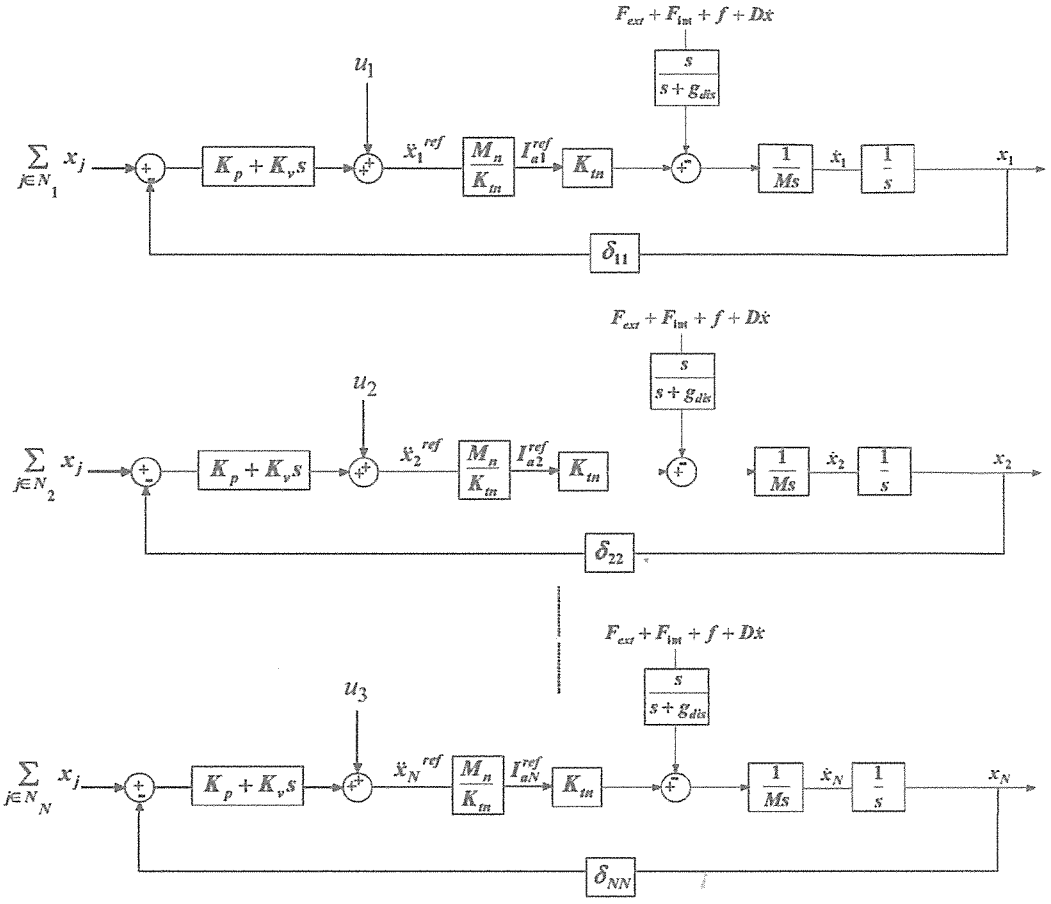


Figure 3.4: Acceleration consensus algorithm

By manipulating this equation and taking the Laplace transform the transfer function is obtained where $X(s) = H_{xu}U(s)$:

$$U(s) = (s^2 I + sK_v L + K_p L)X(s) \tag{3.26}$$

$$H_{xu} = (s^2 I + sK_v L + K_p L)^{-1} \tag{3.27}$$

In order to get the transfer function to the acceleration response, it is differentiated twice so that:

$$H_{au} = s^2(s^2 I + sK_v L + K_p L)^{-1} \tag{3.28}$$

If the servo reference is defined as the average of all inputs; $\frac{1}{N}11^T U$, where 11^T is a matrix of all ones, the error transfer function is obtained:

$$E(s) = \frac{1}{N}11^T U(s) - s^2 X(s) \tag{3.29}$$

$$H_{eu} = \frac{1}{N}11^T - H_{au} \tag{3.30}$$

Using spectral decomposition the Laplacian matrix can be written as:

$$L = \sum_i \lambda_i P_i \tag{3.31}$$

where λ_i are the eigenvalues of the Laplacian and P_i are the orthogonal projections on the orthogonal eigenspaces. The Laplacian has the following properties:

$$(1)\ \lambda_1 = 0 \tag{3.32}$$

$$(2)\ P_1 = \frac{1}{N}11^T \tag{3.33}$$

$$(3)\ \lambda_i > 0 \text{ for all } i > 1 \tag{3.34}$$

Hence the error transfer function can be written as :

$$\begin{aligned} H_{eu} &= \frac{1}{N}11^T - \left(\frac{1}{N}11^T + \sum_{i>1} \frac{s^2}{s^2 + sK_v\lambda_i + K_p\lambda_i} P_i\right) \\ &= -\sum_{i>1} \frac{s^2}{s^2 + sK_v\lambda_i + K_p\lambda_i} P_i \end{aligned} \tag{3.35}$$

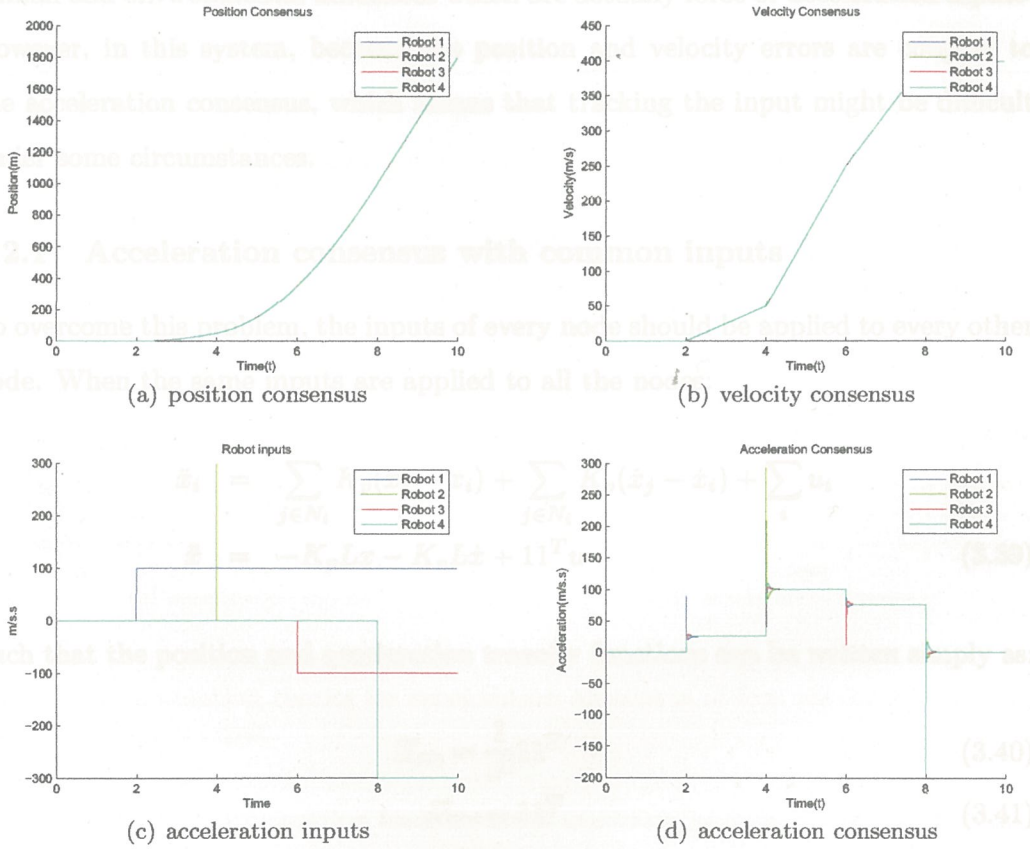
Because λ_i, K_p and K_v are all positive, using final value theorem it can be shown that $e(t) \to 0$ as $t \to \infty$. Hence it can be concluded that the acceleration of the systems will converge to the average of the inputs as $t \to \infty$:

$$\ddot{x} = \frac{1}{N} 1 1^T U \tag{3.36}$$

$$\dot{x}_i = \dot{x}_j = \ldots = \dot{x}_n \tag{3.37}$$

$$x_i = x_j = \ldots = x_n \tag{3.38}$$

This result shows that for any arbitrary input, the acceleration of each robot will



(a) position consensus

(b) velocity consensus

(c) acceleration inputs

(d) acceleration consensus

Figure 3.5: Simulation results for acceleration consensus of four robots

reach a consensus which is the average of the inputs applied to every node. It also guarantees that velocity and position consensus is reached. Thus the systems will identically track the average of any acceleration input. The simulation results for acceleration consensus of 4 robots with different acceleration inputs can be seen in figure 3.5. The robot accelerations eventually converge to the average of the inputs, however there are spikes when the signals are initially applied and there is some tracking error because the acceleration consensus is coupled to the position and velocity errors. his makes acceleration consensus applicable to haptic teleoperation, because the goal is to make sure that robots have the same motions under various human and environmental influences which are actually force or acceleration inputs. However, in this system, because the position and velocity errors are coupled to the acceleration consensus, which means that tracking the input might be difficult under some circumstances.

3.2.1 Acceleration consensus with common inputs

To overcome this problem, the inputs of every node should be applied to every other node. When the same inputs are applied to all the nodes:

$$\begin{aligned} \ddot{x}_i &= \sum_{j \in N_i} K_p(x_j - x_i) + \sum_{j \in N_i} K_v(\dot{x}_j - \dot{x}_i) + \sum_i u_i \\ \ddot{x} &= -K_p L x - K_v L \dot{x} + 1 1^T u \end{aligned} \tag{3.39}$$

Such that the position and acceleration transfer functions can be written simply as:

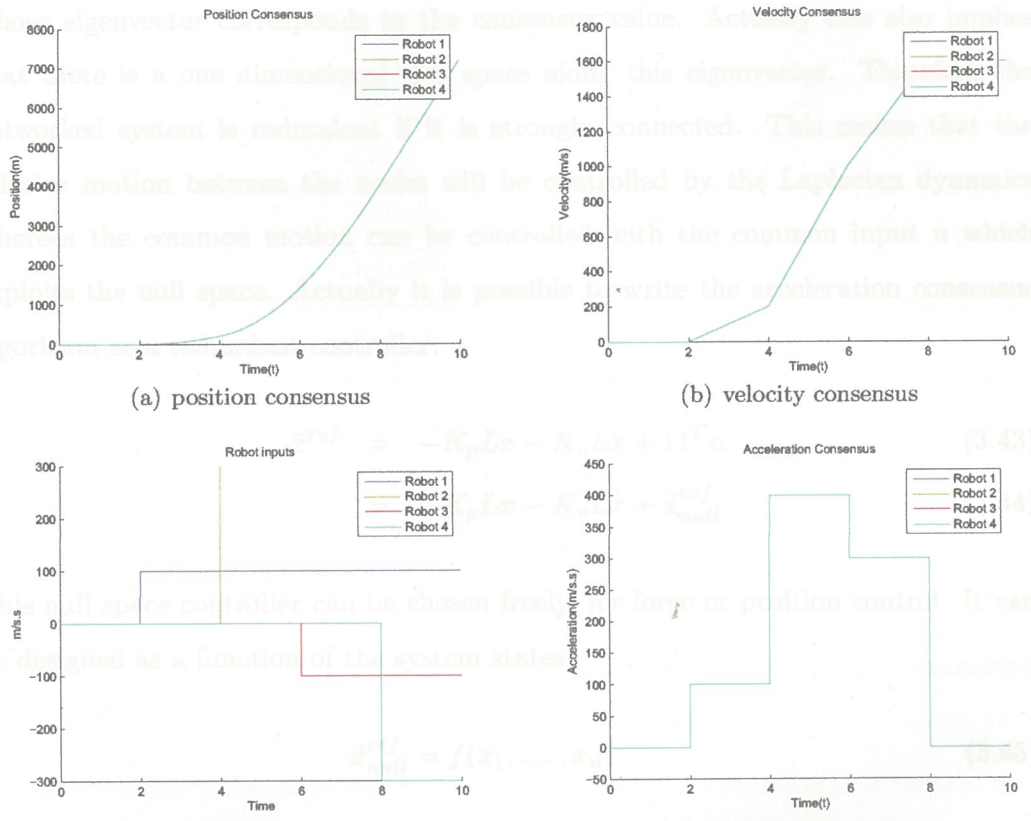
$$H_{xu} = \frac{1}{s^2} 1 1^T \tag{3.40}$$

$$H_{au} = 1 1^T \tag{3.41}$$

which implies that the input tracking error immediately becomes zero:

$$H_{cu} = N\frac{1}{N}11^T - 11^T \tag{3.42}$$

This is in fact a very important result for in research which is related to haptics. In



(a) position consensus

(b) velocity consensus

(c) acceleration inputs

(d) acceleration consensus

Figure 3.6: Simulation results for acceleration consensus of four robots with a common acceleration input

haptics, a common acceleration feedforward is necessary because it can take too long for the systems to reach acceleration consensus if the inputs are applied locally. A common input enables fast tracking of the signals as well as decreasing the relative

errors as can be seen in figure 3.6

3.2.2 Null space controller

Furthermore it is also possible to design a controller for the acceleration consensus value. It should be noted that the connected graph has always one eigenvalue 0 whose eigenvector corresponds to the consensus value. Actually this also implies that there is a one dimensional null space along this eigenvector. Therefore the networked system is redundant if it is strongly connected. This means that the relative motion between the nodes will be controlled by the Laplacian dynamics whereas the common motion can be controlled with the common input u which exploits the null space. Actually it is possible to write the acceleration consensus algorithm as a redundant controller:

$$\ddot{x}^{ref} = -K_p L x - K_v L \dot{x} + 1 1^T u \tag{3.43}$$

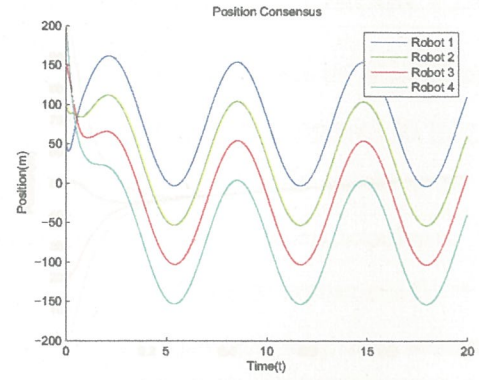
$$= -K_p L x - K_v L \dot{x} + \ddot{x}^{ref}_{null} \tag{3.44}$$

This null space controller can be chosen freely, for force or position control. It can be designed as a function of the system states:

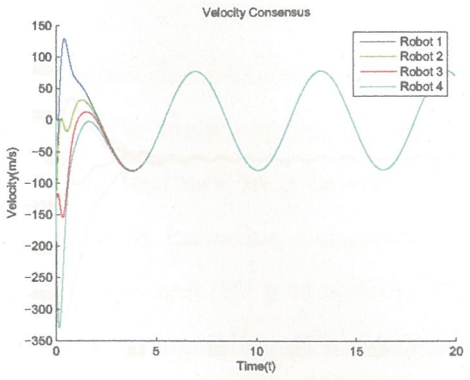
$$\ddot{x}^{ref}_{null} = f(x_1, \ldots, x_n) \tag{3.45}$$

For instance one of the robot positions can be controlled and all the other robots will follow that robot. Or the center of mass of the robots can be controlled, such that the overall motion of these robots, as well as their relative motion can be controlled. This can be realized with the controller:

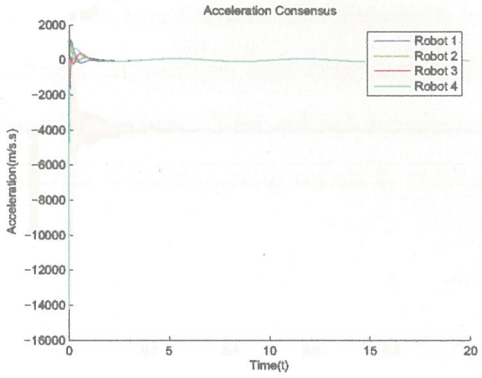
$$\ddot{x}^{ref}_{null} = -K_p(x^{com} - 1 1^T x) - K_v(\dot{x}^{com} - 1 1^T \dot{x}) \tag{3.46}$$



(a) position consensus



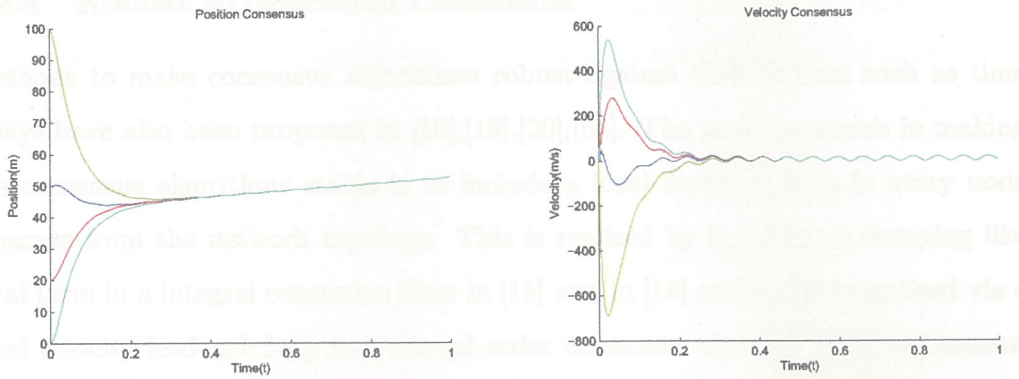
(b) velocity consensus



(c) acceleration inputs

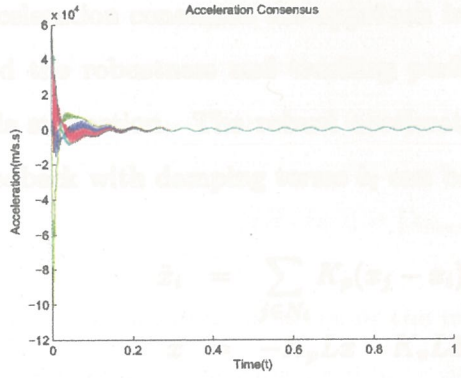
Figure 3.7: Simulation results for acceleration consensus algorithm with formation center control

Figure 3.7 shows the simulation results for control of the formation of 4 robots. The relative positions and velocities are controlled by the laplacian dynamics whereas the center of the formation is controlled by the null space controller which is given a sinusoid reference. It is possible to realize a wide range of networked motion goals using the null space and consensus algorithms.



(a) position consensus

(b) velocity consensus



(c) acceleration consensus

Figure 3.8: Simulation results for acceleration consensus under switching

3.2.3 Acceleration consensus under switching

It was also shown previously that the consensus algorithms are robust to switching in the topology of the network, hence the acceleration consensus algorithms are also robust to switching. Simulation results for acceleration consensus under a high frequency switching network can be seen in figure 3.8.

3.2.4 Robust Acceleration Consensus

Methods to make consensus algorithms robust against disturbances such as time delays have also been proposed in [15],[19],[20],[54]. The main approach in making the consensus algorithms stable is to include a local feedback loop in every node separate from the network topology. This is realized by including a damping like local term in a integral consensus filter in [15] and in [19] and [20] it is realized via a local velocity feedback loop in a second order consensus algorithm. The consensus conditions and robustness requirements have been generalized in [54]. For robust acceleration consensus the approach in [19],[20] will be utilized in this dissertation and the robustness and tracking performance of this algorithm will be shown in this subsection. The robust acceleration consensus algorithm with local velocity feedback with damping terms b_i can be written simply as:

$$\begin{aligned} \ddot{x}_i &= \sum_{j\in N_i} K_p(x_j - x_i) + \sum_{j\in N_i} K_v(\dot{x}_j - \dot{x}_i) - b\dot{x}_i + u_i \\ \ddot{x} &= -K_pLx - K_vL\dot{x} - b\dot{x} + u \end{aligned} \tag{3.47}$$

Before showing the tracking results for this algorithm its robustness will be shown. The difference of the conventional acceleration consensus algorithm and the robust consensus algorithm can be more clearly seen when the local dynamics of the robots are separated from the network dynamics as in [19]. The local robot dynamics for the acceleration consensus algorithm can be written as :

$$H_i(s) = \frac{K_p + K_v(s)}{s^2 + \delta_{ii}K_v(s) + \delta_{ii}K_p(s)} \tag{3.48}$$

and the overall group dynamics can be written as:

$$x(s) = H(s)x(s) + H_I(s)u \tag{3.49}$$

where $H_I(s)$ is the vector transfer function obtained by stacking the individual H_i terms and $H(s)$ is the special adjacency matrix which has the adjacency matrix with weights $H_i(s)$

$$H(s) = \begin{cases} a_{ij}(s)H_i(s) & if\ (v_i, v_j) \in \mathcal{E} \\ 0 & otherwise \end{cases}$$

where the edge weights are such that they satisfy $|a_{ij}(s)| = 1$. For instance if there are delay elements on every link these values would be $a_{ij}(s) = e^{-T_{ij}s}$ and again it is obvious that $|a_{ij}(s)| = |e^{-T_{ij}s}| = 1$. The stability of the system would than be dependent on the spectral radius of H and the stability of input u. The stability condition can be written as[19][64]:

$$\rho(H(jw)) = |\lambda_{max}(H(jw))| < 1\,, \quad \forall w \tag{3.50}$$

where ρ is the spectral radius or the maximum value of the maximum eigenvalue of H for all w. In addition to this, u should be a stable input and will be assumed so. According to Gersgorin's theorem, which states that all the eigenvalues of H are located in or on the union of following disks in the complex plane C:

$$D_i(s) := \{\lambda \in C: \ |\lambda| \leq \sum_{j \in N_i} |a_{ij}(s)H_i(s)|\} \tag{3.51}$$

Since $|a_{ij}(s)| = 1$ and $\sum_{j \in N_i} a_{ij} = \delta_{ii}$, this condition can simply be rewritten as:

$$\begin{aligned} \rho(H(jw)) &\leq \delta_{ii}\,|H_i(s)| \\ \delta_{ii}\,|H_i(s)| &< 1\,, \quad \forall w \\ |H_i(s)| &< \frac{1}{\delta_{ii}}\,, \quad \forall w \end{aligned} \tag{3.52}$$

since the absolute value of complex numbers is the square root of the complex number multiplied with its complex conjugate it is possible to write the absolute value of H_i as:

$$\begin{aligned} |H_i(jw)| &= \sqrt{H_i(-jw)H_i(jw)} \\ &= \sqrt{\frac{(K_p)^2 + (K_v)^2 w^2}{(\delta_{ii}K_p)^2 - 2(\delta_{ii}K_p)w^2 + (\delta_{ii}K_v)^2 w^2 + w^4}} \end{aligned} \tag{3.53}$$

It can simply be seen that for $w^2 < 2(\delta_{ii}K_p)$ the absolute value will be greater than $\frac{1}{\delta_{ii}}$ which makes the system unstable. On the other hand, when local velocity feedback is applied with damping b the local dynamics can be written as:

$$H_i(s) = \frac{K_p + K_v(s)}{s^2 + \delta_{ii}K_v(s) + b(s) + \delta_{ii}K_p(s)} \tag{3.54}$$

which in turn yields:

$$|H_i| = \sqrt{\frac{(K_p)^2 + (K_v)^2 w^2}{(\delta_{ii}K_p)^2 - 2(\delta_{ii}K_p)w^2 + (\delta_{ii}K_v + b)^2 w^2 + w^4}}$$

and if the damping terms are chosen as:

$$b \geq \sqrt{2\delta_{ii}K_p} \tag{3.55}$$

we will obtain

$$|H_i| = \sqrt{\frac{(K_p)^2 + (K_v)^2 w^2}{(\delta_{ii}K_p)^2 + (\delta_{ii}K_v)^2 w^2 + 2(\delta_{ii}K_v b)^2 w^2 + w^4}} < \frac{1}{\delta_{ii}}, \quad \forall w > 0$$

and because the value will be equal to 1 for $w = 0$ the system will be marginally stable. Thus, it is clear that introduction of local velocity feedback will provide stability even in the presence of time delay, independent of the size of the delay. This is why, in chapters 6 and 7 a specialized version of this algorithm will be used for

teleoperation under time delay. Next the tracking results for the robust acceleration algorithm is shown. For this analysis no delay on the network is assumed. The transfer function can be obtained as:

$$U(s) = (s^2 I + sbI + sK_v L + K_p L)X(s) \tag{3.56}$$

$$H_{xu} = (s^2 I + sbI + sK_v L + K_p L)^{-1} \tag{3.57}$$

This transfer function can be rewritten using the spectral properties of the Laplacian:

$$\begin{aligned} H_{xu} &= \frac{1}{(s^2+bs)N}11^T + \sum_{i>1}\frac{1}{s^2+sb+K_v\lambda_i+K_p\lambda_i}P_i \\ H_{vu} &= \frac{1}{(s+b)N}11^T + \sum_{i>1}\frac{s}{s^2+sb+sK_v\lambda_i+K_p\lambda_i}P_i \\ H_{au} &= \frac{s}{(s+b)N}11^T + \sum_{i>1}\frac{s^2}{s^2+sb+sK_v\lambda_i+K_p\lambda_i}P_i \end{aligned} \tag{3.58}$$

based on these transfer functions it can be seen that the acceleration tracking is realized through a high pass filter, therefore the reference is defined as a highpass version of the average of the inputs and the error becomes:

$$E(s) = \frac{s}{(s+b)N}11^T U(s) - s^2 X(s) \tag{3.59}$$

$$H_{eu} = \frac{s}{(s+b)N}11^T - H_{au} \tag{3.60}$$

$$= -\sum_{i>1}\frac{s^2}{s^2+sb+sK_v\lambda_i+K_p\lambda_i}P_i \tag{3.61}$$

and similarly for previous cases the error will go to zero except for the . It can be seen that now accelerations will be tracked based on the cut-off frequency of the filter which is determined by the damping term b. Furthermore this also implies that the velocities will be tracked by some scaled low pass filter such that the tracking

results are as time goes to infinity :

$$\ddot{x} \rightarrow \frac{s}{(s+b)N} 11^T U \tag{3.62}$$

$$\dot{x} \rightarrow \frac{1}{(s+b)N} 11^T U \tag{3.63}$$

$$x_i \rightarrow x_j \rightarrow \ldots \rightarrow x_n \tag{3.64}$$

Figure 3.9 shows the responses of three robots connected with a robust consensus algorithm. It can be seen that for a constant input the robot accelerations will immediately go to zero and the robot velocities will track the $\frac{1}{bN} 11^T U$ where $b = 200$, as the input frequency is 0. Furthermore the positions of the robots will track each other with some error due to local damping. If the servo reference is defined as the sum of all the inputs $11^T u$ the following consensus algorithm is obtained:

$$\begin{aligned} \ddot{x}_i &= \sum_{j \in N_i} K_p(x_j - x_i) + \sum_{j \in N_i} K_v(\dot{x}_j - \dot{x}_i) - b\dot{x}_i + \sum_i u_i \\ \ddot{x} &= -K_p L x - K_v L \dot{x} - b\dot{x} + 11^T u \end{aligned} \tag{3.65}$$

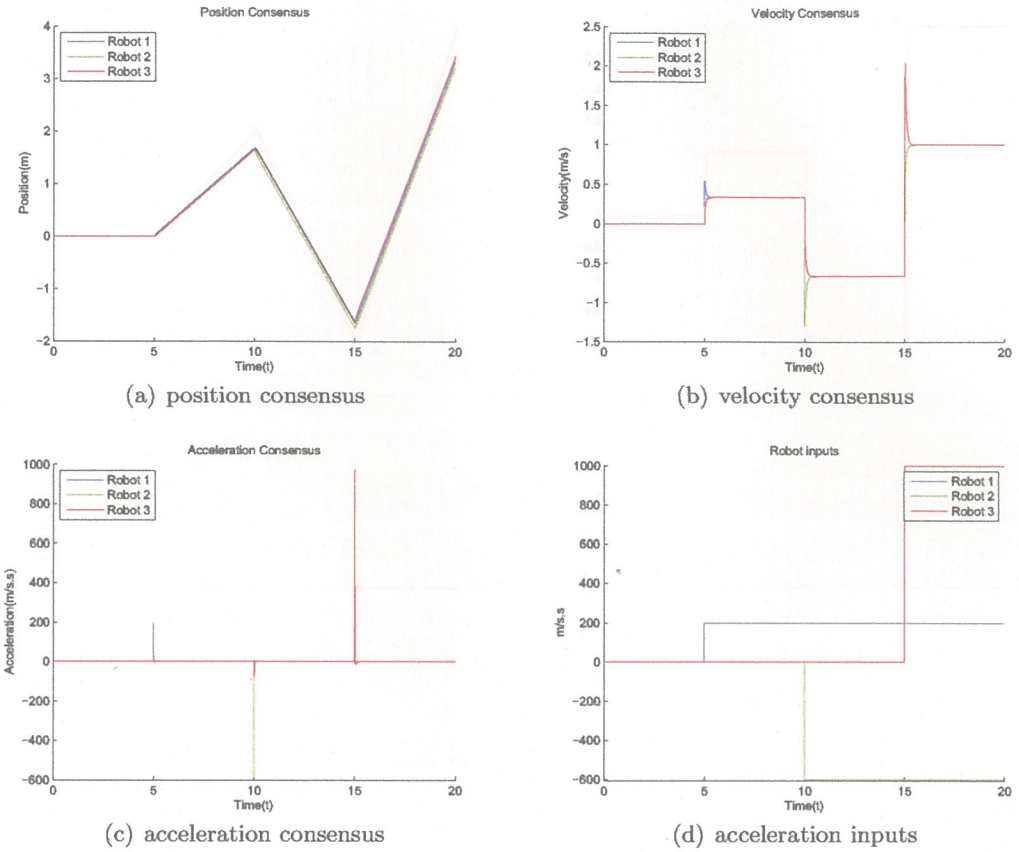
This would not change the stability analysis as long as the inputs are stable but immediate tracking results obtained:

$$\ddot{x} = \frac{s}{(s+b)} 11^T U \tag{3.66}$$

$$\dot{x} = \frac{1}{(s+b)} 11^T U \tag{3.67}$$

$$x = \frac{s^2}{(s^2+bs)} 11^T U \tag{3.68}$$

Figure 3.10 shows simulations results for the case where common inputs are applied to all the robots. It indeed verifies that perfect tracking is realized, with the similar filtered acceleration and velocity outputs. Because step inputs were used in these simulations the effect of the filtering may not be visible, therefore figure 3.11 shows



(a) position consensus

(b) velocity consensus

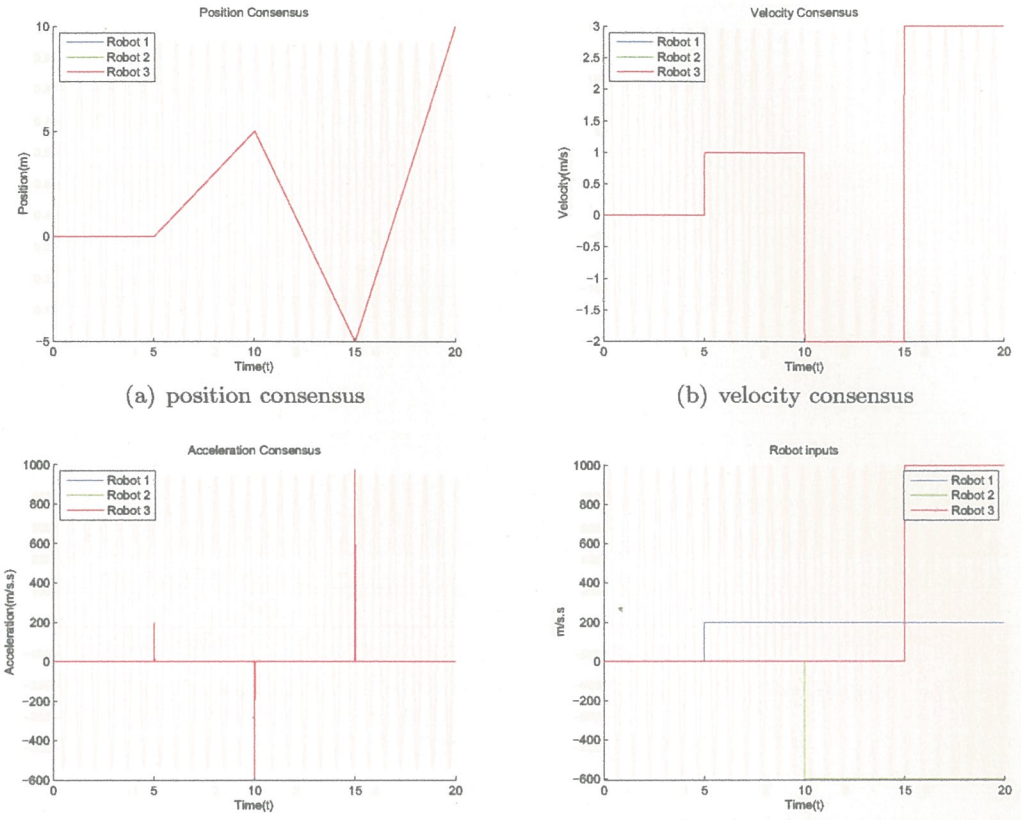
(c) acceleration consensus

(d) acceleration inputs

Figure 3.9: Simulation results for robust acceleration consensus for 3 robots

the simulation results when sinusoidal inputs with 10hz frequency were utilized. In this case the value of the velocity response is slightly less than $\frac{1}{b}11^T U$ because of the filtering. If the frequency of the signal is higher than damping value then the signals would be cut-off. On the other hand, the stability analysis that was provided showed the robustness of this algorithm under time delay. When there is time delay on all the channels the consensus algorithm can be written as:

$$\ddot{x}_i = \sum_{j \in N_i} K_p(x_j(e^{-T_{ij}s}) - x_i) + \sum_{j \in N_i} K_v(\dot{x}_j(e^{-T_{ij}s}) - \dot{x}_i) - b\dot{x}_i + \sum_i u_i$$



(a) position consensus

(b) velocity consensus

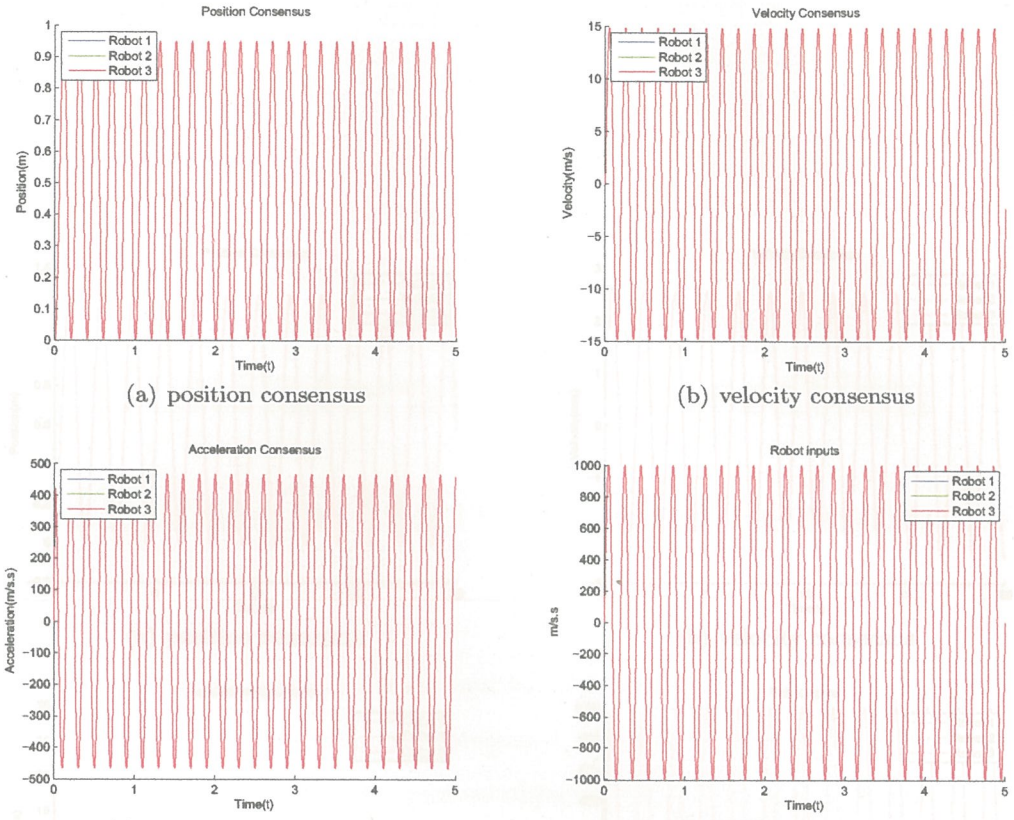
(c) acceleration consensus

(d) acceleration inputs

Figure 3.10: Simulation results for robust acceleration consensus with common input for 3 robots

$$\ddot{x} = -K_p L(s)x - K_v L(s)\dot{x} - b\dot{x} + 11^T u \tag{3.69}$$

In the presence of time delay it is difficult to obtain the tracking results, but simulation results show that with the assumption of a common input that is unaffected from time delay acceleration tracking would be realized. The tracked common value depends on the size of time delay between the channels. Figure 3.12 shows the simulation results when the delay among each link $T_{ij} = 0.2$s. These results actually show that the use of acceleration consensus algorithms in haptic teleoperation



(a) position consensus

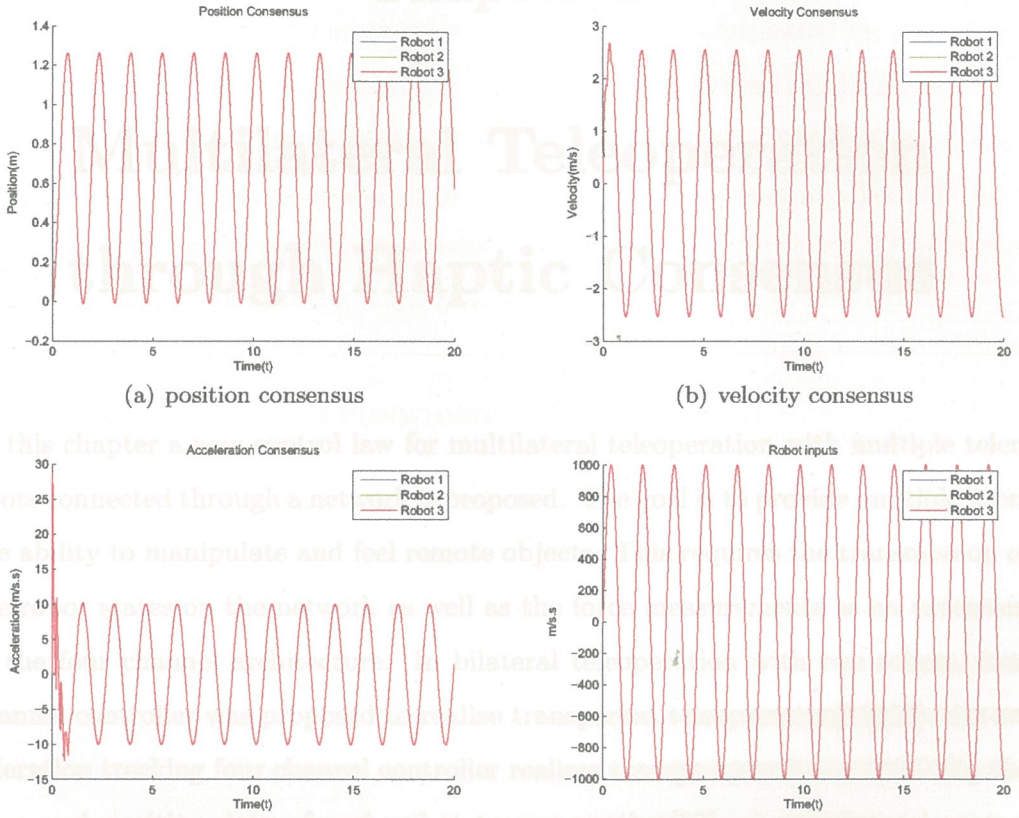
(b) velocity consensus

(c) acceleration consensus

(d) acceleration inputs

Figure 3.11: Simulation results for robust acceleration consensus for 3 robots with sinusoidal input

especially under time delay could be very effective.



(a) position consensus

(b) velocity consensus

(c) acceleration consensus

(d) acceleration inputs

Figure 3.12: Simulation results for robust acceleration consensus for 3 robots with 200ms time delay on each link

Chapter 4

Multilateral Teleoperation through Haptic Consensus

In this chapter a new control law for multilateral teleoperation with multiple telerobots connected through a network is proposed. The goal is to provide multiple users the ability to manipulate and feel remote objects. This requires the transmission of the robot states on the network as well as the force measurements as an extension of the four channel architecture. In bilateral teleoperation with two robots, four channel controller was proposed to realize transparent teleoperation[26][27]. An acceleration tracking four channel controller realizes transparency by transmitting the force and position data of each robot to one another[30]. A multilateral control strategy was also demonstrated using acceleration control; which provides shared telepresence to multiple teleoperators by feedforwarding the sum of all force inputs on the robots[36]-[40]. This strategy requires the transmission of all force and position data among the robots. It was previously shown that consensus algorithms enable nodes on a network to agree on a common value depending on the initial conditions, by transmitting the states according to the network topology represented by a graph and its Laplacian matrix. It was also shown that consensus filters are mod-

ified consensus algorithms which enable the distributed averaging of local inputs by the nodes on the network. The algorithm shown in this chapter, also called the haptic consensus algorithm, is an acceleration consensus algorithm which is a synthesis of acceleration control and consensus filters. A completely distributed control law that can operate on any strongly connected and balanced graph is proposed, where the sum of the forces acting on the teleoperation system is estimated via distributed force consensus filters. Furthermore, it is shown that the system can be stable under network topologies under dynamically switching topologies. This kind of an algorithm would be extremely useful where there are communication blackouts in some communication links between the robots, and could guarantee stable teleoperation as long as the overall graph is connected.

4.1 Multilateral Teleoperation

In multilateral haptic teleoperation, multiple teleoperators are used and the forces felt at each robot must be transmitted to the others, and the robots must have the same motion under all circumstances. For instance all the robots should copy the motion of the master robot(s) which is/are controlled by an operator/operators. And if any of the slaves contact any object, the impact force should be transmitted to the operator/s. This means that at steady state; the sum of the forces felt at every robot must equal zero, and the states of each robot such as position, velocity and acceleration must be equal:

$$F_1 + F_2 + \ldots + F_N = 0 \tag{4.1}$$

$$x_1 = x_2 = \ldots = x_N \tag{4.2}$$

For these goals N decentralized acceleration controllers can be designed as:

$$\begin{aligned}
\ddot{x}_1^{ref} &= \ddot{x}_1^{cmd} + K_p(x_1^{cmd} - \delta_{11}x_1) + K_v(\dot{x}_1^{cmd} - \delta_{11}\dot{x}_1) \\
\ddot{x}_2^{ref} &= \ddot{x}_2^{cmd} + K_p(x_2^{cmd} - \delta_{22}x_2) + K_v(\dot{x}_2^{cmd} - \delta_{22}\dot{x}_2) \\
&\vdots \\
\ddot{x}_N^{ref} &= \ddot{x}_N^{cmd} + K_p(x_N^{cmd} - \delta_{nn}x_N) + K_v(\dot{x}_N^{cmd} - \delta_{nn}\dot{x}_N)
\end{aligned}$$

where δ_{ii} are the diagonal entries of the outdegree matrix Δ and the command values for the position and velocity are defined by the connection matrix $\mathcal{A}$:

$$x^{cmd} = \mathcal{A}x, \;\; \dot{x}^{cmd} = \mathcal{A}\dot{x} \tag{4.3}$$

and all the acceleration feedforwards are the same:

$$\ddot{x}_1^{cmd} = \ddot{x}_2^{cmd} = \ldots = \ddot{x}_N^{cmd} = F_1 + F_2 + \ldots + F_N \tag{4.4}$$

If the the same acceleration feedforward is given to all the robots, assuming identical nominal inertias M_n: a multilateral acceleration control law can be written using the Laplacian matrix as follows:

$$\ddot{x}^{ref} = -K_pLx - K_vL\dot{x} + \frac{1}{M_n}(F_1 + F_2 + \ldots + F_N) \tag{4.5}$$

Using the disturbance observers and sensitivity functions from section 3, this algorithm can be written as:

$$\begin{aligned}
-\frac{1}{M_n}G(s)F_{dis} + \ddot{x} = &-K_pLx - K_vL\dot{x} \\
&+\frac{1}{M_n}(F_1 + F_2 + \ldots + F_N)
\end{aligned} \tag{4.6}$$

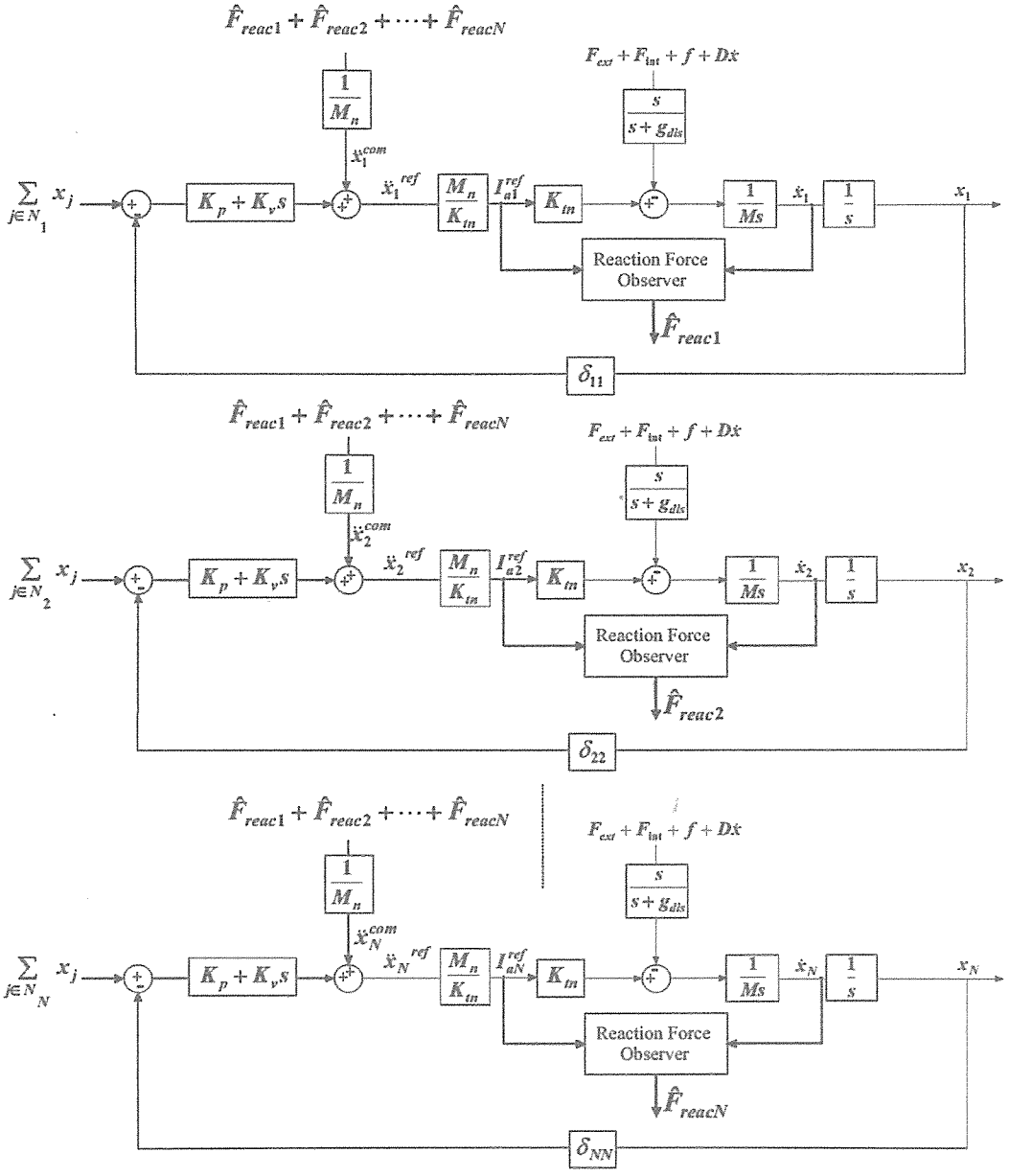


Figure 4.1: An acceleration consensus algorithm for haptic teleoperation

where $u = F_1 + F_2 + \ldots + F_N$ This structure can be seen in figure 4.1. Using disturbance observers, systems can be transformed into nominal models. If the disturbance estimation and compensation is successful, the sensitivity function $G(s)$ becomes zero, hence the multilateral dynamics is transformed into the consensus dynamics. Stability of accelaration consensus algorithm also implies the overall stability of the multilateral teleoperation system. Because the inputs to all the robots are the same;$u_1 = u_2 = \ldots = u_N = \frac{1}{M_n}(F_1 + F_2 + \ldots + F_n)$, the systems will reach acceleration consensus immediately as shown in chapter 3. The acceleration consensus value will be the average of the inputs to the systems. So at steady state they will be:

$$\ddot{x}_m = \ldots = \ddot{x}_N = \frac{1}{M_n}(F_1 + F_2 + \ldots + F_N) \tag{4.7}$$

By achieving acceleration consensus, it is also guaranteed that the robots will track each others' velocities and positions:

$$\dot{x}_1 = \dot{x}_2 = \ldots = \dot{x}_N \tag{4.8}$$

$$x_1 = x_2 = \ldots = x_N \tag{4.9}$$

Thus the haptic systems will behave exactly the same under various environmental conditions; by achieving acceleration, velocity and position consensus; and as well as having the sum of their forces 0. So haptic consensus can also be realized in multilateral teloperation. What is more multilateral teleoperation systems let more than one operator feel an object, thus it becomes possible to realize consensus among operators by the sensation of touch.

4.1.1 Haptic consensus with consensus filters

The consensus filters were introduced in chapter 3. By using a consensus algorithm and a consensus filter it is possible to realize haptic consensus which makes sure that

the position values are always the same and the accelerations of the systems track a common force reference. The common force reference was previously defined as the sum of the forces $F_1 + F_2 + \ldots + F_N$. However if it is desired that the robots transmit all their data according to the network topology, the force data should also be sent like the position and velocity data, according to the graph of the network. However when the force data are transmitted according to the graph it is not possible to realize haptic consensus; the system would be unstable. Therefore, there are two solutions to this problem, either there should be a different network dedicated for the force data, and every robot should send its force measurement to every other robot; which means that every node on the network graph should be connected. This is not very practical, because it limits the flexibility of the network. The other solution is to use consensus filters which were introduced in chapter 3. As we know, consensus filters can estimate the average of the inputs on every node, just by communicating their states according to the network graph. If the network is strongly connected and balance it was shown that the information being propogated on the network will converge to the averages of the input. Therefore, consensus filters provide a good solution to this problem; they can estimate the average of all the force inputs in the teleoperation network. By multiplying this average with the number of robots the sum of the force measurement of every robot can easily be estimated. By employing consensus filters haptic consensus on any strongly connected and balanced network can be realized, transmitting both the force and position data according to the graph. Here is how the force consensus filter works; forces measured at every node locally can be used as the inputs to the consensus filter:

$$\dot{u}_i = \sum_{j \in N_i} a_{ij}(u_j - u_i) + F_i \tag{4.10}$$

It is known that the derivative of consensus filter states u_i estimate the average of the inputs which are reaction forces measured locally at every node. As this is a

high pass consensus filter defined in chapter 3, at steady state the derivative of the filters converge to:

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_N \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} \tag{4.11}$$

Thus if the average is multiplied with the number of robots N the sum of the

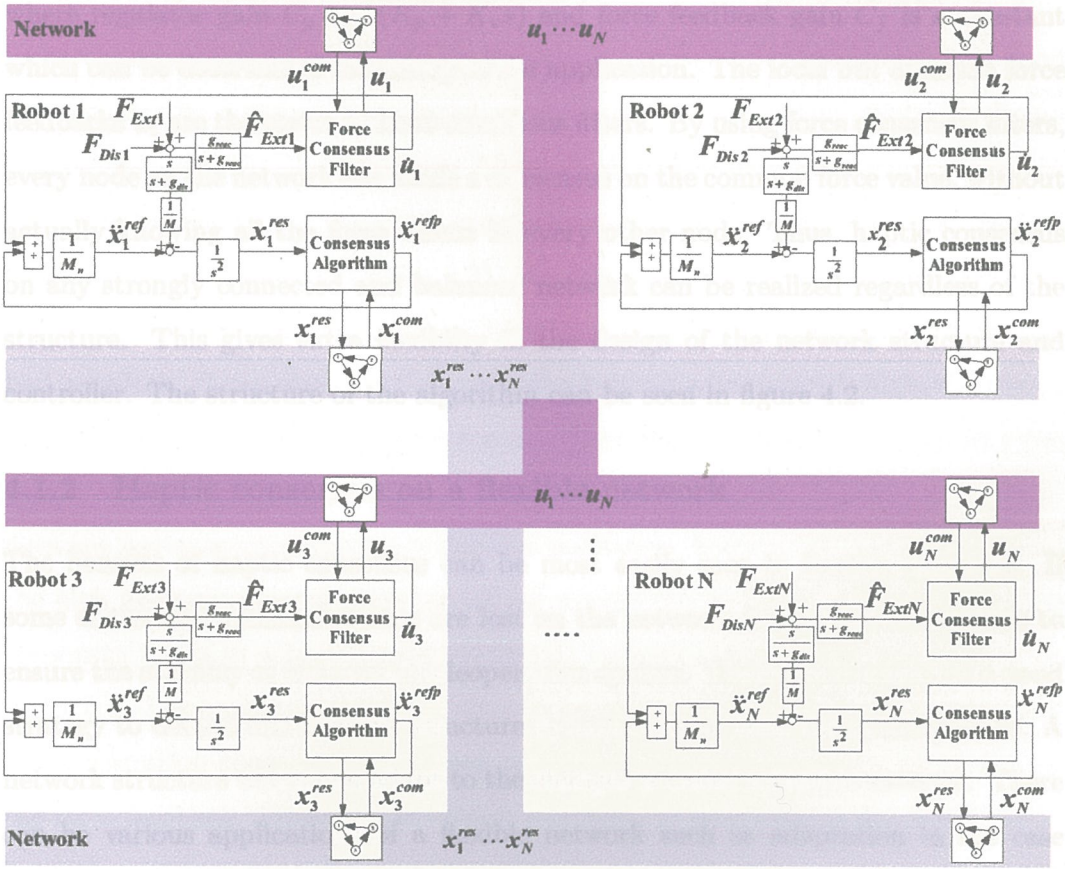


Figure 4.2: Haptic consensus with force consensus filters

forces can be obtained. For this the only condition is that the network is strongly connected and balanced. Then it is possible to explicitly write the haptic consensus algorithm for multilateral teleoperation system with N robots, using force consensus filters, as:

$$\ddot{x}_i^{ref} = C_p(s) \sum_{j \in N_i} (x_j - x_i) + C_f N \dot{u}_i \tag{4.12}$$

$$\dot{u}_i = \sum_{j \in N_i} a_{ij}(u_j - u_i) + F_i \tag{4.13}$$

where regulator gain $C_p = \frac{1}{2}(K_p + K_v s)$ and force feedback gain C_f is a constant which can be determined according to the application. The local but common force feedbacks u_i are the states of force consensus filters. By using force consensus filters, every node on the network can reach a consensus on the common force value, without actually knowing all the force values in every other node. Thus, haptic consensus on any strongly connected and balanced network can be realized regardless of the structure. This gives extra flexibilty in the design of the network structure and controller. The structure of the algorithm can be seen in figure 4.2.

4.1.2 Haptic consensus on a flexible network

The benefits of haptic consensus can be most easily seen in flexible networks. If some of the communication lines are lost on the network, it may not be possible to ensure the stability of the overall teleoperation system. Therefore it might be a good strategy to change the network structure dynamically to keep the system stable. A network structure which can adapt to the changes is called a flexible network. There can be various applications of a flexible network such as adaptation in the case of communication blackouts, finding the optimal routes in the case of nonuniform time delays, or even possibly optimizing the control performance. As shown in the previous chapter, acceleration consensus algorithms are robust to changes in the

graph topology, as long as the changing graphs are always strongly connected and balanced. That is what makes acceleration consensus algorithms and consensus filters a valuable tool to realize a flexible network. Examples of haptic consensus on a switching network in the experiments section will be shown to confirm that flexible network can be realized through consensus algorithms.

4.1.3 Haptic consensus with consensus filters

Experiments were also conducted to test if the consensus filters can succesfully track the average of the force inputs and whether these filter states could safely be used in haptic consensus. The results confirmed that force consensus filters are a good solution for the common force input generation on a Laplacian graph. It should be assessed whether the consensus filter based haptic consensus can be realized on a flexible network. Therefore experiments on a switching network at various frequencies was conducted(figure 4.3). In this scenario because the network was switching, the acceleration consensus algorithm as well as the force consensus filter had to switch. It was observed that when the switching frequency is high, there were high frequency oscillations in the consensus filter outputs. However this oscillations do not show up as a big problem in the acceleration outputs. Indeed the operator does not feel to much disturbance due this oscillation, however the sound created by the high frequency current pumped on the motors can be heard. The output of the consensus filters in a high frequency(10kHz) switching experiment can be seen on figure 4.4. The haptic consensus results can be seen on figure 4.5. It can be seen that the acceleration consensus value does not reflect the high frequency oscillation in the force feedback. When the switching frequency is decreased in the experiments, it can be observed that there is not a high frequency oscillation, however at every switching instant there is a spike on the filter output. These spikes can hardly be felt by the operators when the filter output is fedback. The consensus filter output

with a switching frequenchy of 10Hz can be seen in figure 4.6. Note the spikes at every switch. The haptic consensus results can be seen on figure 4.7. The spikes can be seen in the acceleration output as well, but are too short to be recognized by an operator. With a better implementation of the consensus filter it may be possible to solve these problems caused from switching.

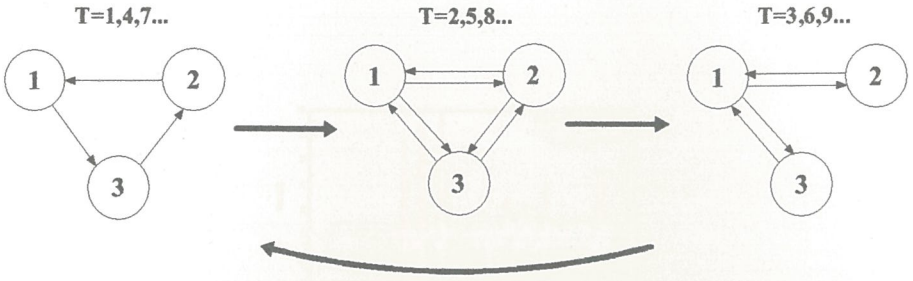


Figure 4.3: Switching Network Structure

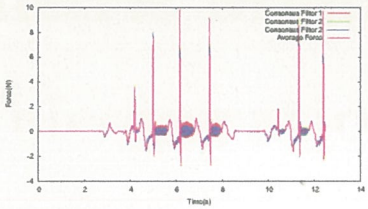
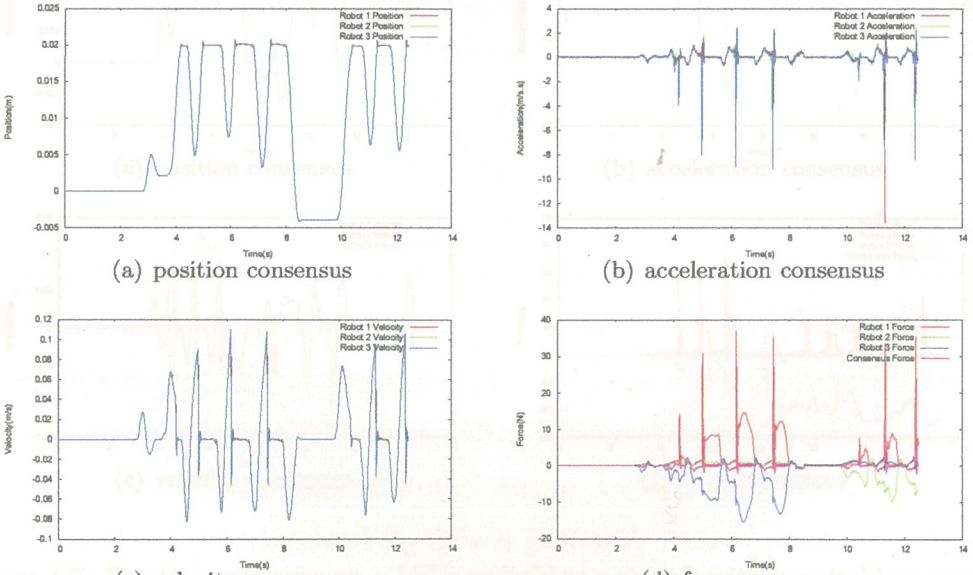


Figure 4.4: Consensus filter tracking in switching network structure



(a) position consensus

(b) acceleration consensus

(c) velocity consensus

(d) force consensus

Figure 4.5: Haptic consensus on a 10kHz switching network with a switching consensus filter

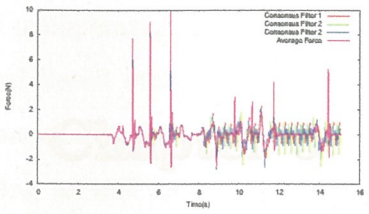
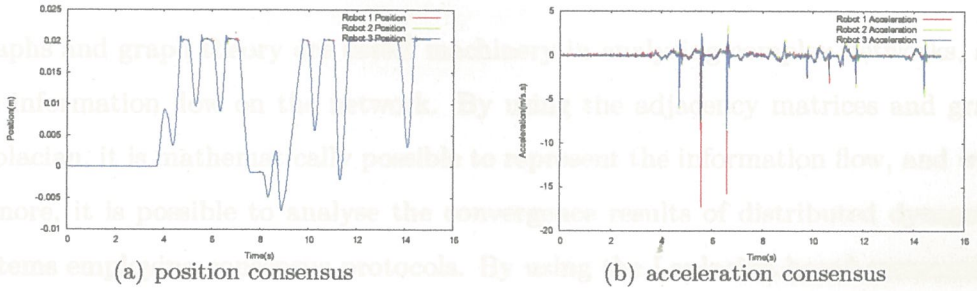


Figure 4.6: Consensus filter tracking in switching network structure



(a) position consensus

(b) acceleration consensus

(c) velocity consensus

(d) force consensus

Figure 4.7: Haptic consensus on a 10Hz switching network with a switching consensus filter

Chapter 5

Asymmetric Multilateral Teleoperation

Graphs and graph theory are useful machinery in analysing complex networks, and the information flow on the network. By using the adjacency matrices and graph Laplacian, it is mathematically possible to represent the information flow, and what is more, it is possible to analyse the convergence results of distributed dynamical systems employing consensus protocols. By using the Laplacian based communication protocols, the nodes on a graph can all agree on a certain value. For instance it was shown in chapter 3 that very simple dynamical systems defined by Laplacian based differential equations, can achieve distributed averaging of initial conditions or local inputs. Average consensus filters have been proposed for distributed sensor fusion on sensor networks[13]-[16], as well as flocking of multiple mobile robots[18]. The contributions of the proposal in this chapter are twofold; first a scaled consensus algorithm based on weighted digraphs is proposed. This represents a divergence from the classical application of the consensus algorithms for agreement on the same value. The scaled consensus algorithm still provides a consensus based on functions defined by the weights, however the node states are scaled, and are not the same.

This provides a richer set of parameters for various applications. The second contribution is that, it is shown that it is possible to employ the consensus algorithms for asymmetric multilateral teleoperation involving multiple telerobots and multiple operators. The four channel based architecture proposed in the previous section, which provides optimal transparency for shared telepresence, is generalized by taking into consideration scaling as well as communication constraints. This means that, this scheme enables scaling among human force inputs, as well as the motions of the telerobots. Thus it becomes possible to realize multi operator teleoperation when asymmetrical cooperation is desired, for instance when a surgeon is instructing a student to perform robotic surgery, the teacher's input can be scaled so that he can be the main master that dominates the operation or can be the secondary user. Furthermore, when the operators want to control micro-robots for minute manipulation tasks such as required by microsurgery, it is obviously necessary to use asymmetrical multilateral teleoperation.

5.1 Scaled Consensus

For scaled consensus a different set of digraphs with weights are used as can be seen on figure 5.1. Scaled consensus is different from other consensus algorithms in the sense that the nodes do not converge on the same value, but are tracking a function of a consensus value intrinsically reached by every node on the graph. The Laplacian matrices for these kinds of weighted digraphs is also introduced. Firstly, for the weighted graphs, the adjacency matrices A_w can be written as follows:

$$a_{w_{ij}} = \begin{cases} \frac{w_j}{w_i} & if\ (v_i, v_j) \in \mathcal{E} \\ 0 & otherwise \end{cases} \tag{5.1}$$

The degree matrix Δ=$[\delta_{ij}]$ is the same with the non-weighted case and Laplacian matrix is computed as $L_w = \Delta - A_w$. This weighted Laplacian matrix can be

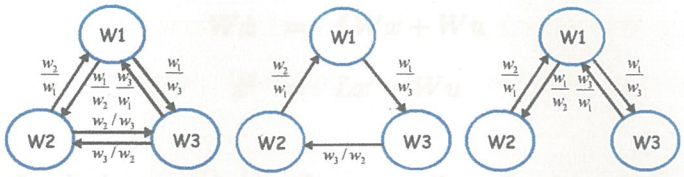


Figure 5.1: Weighted information graph

used for scaled consensus protocols. And the weighted consensus algorithm can be written as:

$$\dot{x}_i = \sum_{v_j \in N_i} \left(\frac{w_j}{w_i} x_j - x_i\right) + u_i \tag{5.2}$$

$$\dot{x} = -L_w x + \dot{u} \tag{5.3}$$

Because the Laplacian is modified, it is not possible to directly show the results available for conventional consensus algorithms. Therefore we define a new vector x' of weighted position values:

$$Wx = x' \tag{5.4}$$

where W is:

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & w_n \end{bmatrix} \tag{5.5}$$

such that the proposed algorithm can be written in the form of a conventional Laplacian algorithm:

$$\dot{x} = -L_w x + u \tag{5.6}$$

$$W\dot{x} = LWx + Wu \tag{5.7}$$

$$\dot{x}' = Lx' + Wu \tag{5.8}$$

where L is the Laplacian matrix for the same digraph without the weights. Thus a consensus algorithm is obtained, however the nodes employing this algorithm are tracking different values because of the weights. To see the tracking result of the algorithm Laplace transform is applied to get the transfer function:

$$WU(s) = (sI + L)X'(s) \tag{5.9}$$

$$H_{x'u} = (sI + L)^{-1}W \tag{5.10}$$

Both sides are differentiated to get the response to the state derivatives:

$$sH_{x'u} = s(sI + L)^{-1}W \tag{5.11}$$

If the input to be tracked is defined as a scaled average of consensus values on all local inputs; $\frac{1}{N}11^TWU$, where 11^T is a matrix of all ones and W is the weight matrix, the error transfer function becomes:

$$E(s) = \frac{1}{N}11^TWU(s) - sWX(s) \tag{5.12}$$

$$H_{eu} = \frac{1}{N}11^TW - sH_{x'u} \tag{5.13}$$

$$= \frac{1}{N}11^TW - s(sI + L)^{-1}W \tag{5.14}$$

Using spectral decomposition Laplacian matrix can be written as:

$$L = \sum_i \lambda_i P_i \tag{5.15}$$

where λ_i are the eigenvalues of the Laplacian and P_i are the orthogonal projections on the orthogonal eigenspaces. The Laplacian has the following properties:

$$\begin{aligned} (1)\ \lambda_1 &= 0 \\ (2)\ P_1 &= \frac{1}{N}11^T \\ (3)\ \lambda_i &> 0 \text{ for all } i > 1 \end{aligned} \tag{5.16}$$

Hence the error transfer function can be written as :

$$\begin{aligned} H_{cu} &= \frac{1}{N}11^T W - \left(\frac{1}{N}11^T W + \sum_{i>1}\frac{s}{s+\lambda_i}P_i W_i\right) \\ &= -\sum_{i>1}\frac{s}{s+\lambda_i}P_i W_i \end{aligned} \tag{5.17}$$

which implies that $e(t) \to 0$ as $t \to \infty$, because the eigenvalues of L are all positive. When the error goes to zero, weighted average tracking of the inputs is realized:

$$W\dot{x} = \frac{1}{N}11^T W u \tag{5.18}$$

Given that the initial conditions are the same, weighted integral tracking is also realized:

$$w_1 x_1 = w_2 x_2 = \ldots = w_n x_n \tag{5.19}$$

5.2 Scaled Acceleration Consensus

It is possible for robots on a network to realize agreement in their acceleration by using an acceleration consensus algorithm. Acceleration consensus algorithm is both an acceleration controller and a consensus filter. In this section a scaled acceleration consensus algorithm using the weighted graphs and scaling ratios is

developed. First the following servo consensus algorithm is defined where K_p and K_v are positive gains and u_i are the local servo input:

$$\ddot{x}_i = \sum_{j \in N_i} \frac{K_p}{w_i}(w_j x_j - w_i x_i) + \sum_{j \in N_i} \frac{K_v}{w_i}(w_j \dot{x}_j - w_j \dot{x}_i) + u_i \tag{5.20}$$

$$\ddot{x} = -K_p L_w x - K_v L_w \dot{x} + u \tag{5.21}$$

As the system is second order, the PD gains should be selected such that the system should at least be critically damped:

$$\frac{K_v}{w_i} = 2\sqrt{\frac{K_p}{w_i}} \tag{5.22}$$

Because scaled acceleration consensus algorithm is basically a scaled consensus filter which was shown in section 1, the acceleration of the robots will be determined by the servo inputs according to the following relationship:

$$W\ddot{x} = \frac{1}{N} 1 1^T W u \tag{5.23}$$

Again, if initial conditions are the same it can be shown that the accelerations,velocities and positions of the robots track each other with the scaling ratios as:

$$w_1 \dot{x}_1 = w_2 \dot{x}_2 = \ldots = w_n \dot{x}_n \tag{5.24}$$

$$w_1 x_1 = w_2 x_2 = \ldots = w_n x_n \tag{5.25}$$

5.3 Asymmetric Multilateral Teleoperation

Multilateral teleoperation based on four channel architecture was proposed[36]-[40] to realize a high performance teleoperation system, which enables shared impedance

matching between multiple operators and the environment. This means that the following force response and position response goals are set:

$$F_1 + F_2 + \ldots + F_N = 0 \tag{5.26}$$

$$X_1 = X_2 = \ldots = X_N \tag{5.27}$$

What these goals represent is actually the shared matching of impedances. Impedances felt at each robot can be written as: $Z_i = \frac{F_i}{sx_i}$. In conventional multilateral control this also translates to the impedance matching goals $Z_1 + Z_2 + \ldots + Z_N = 0$, meaning that the sum of the impedances(considering the direction as well) should be equal to zero. This will ensure that when there is only one operator, the environmental impedance will be exactly reproduced at the master side. When there are multiple masters and forces the combined impedances of the other operators and the environment will be reproduced. In the previous chapter, a multilateral teleoperation method called haptic consensus algorithm was proposed, which attains the same performance criteria of four channel based teleoperators, even if some of the channels among the robots do not exist. In this section, the scaled haptic consensus algorithm for asymmetric multilateral teleoperation will be introduced based on the scaled consensus algorithm proposed in the previous sections. First the goals in an asymmetric multilateral teleoperation system are defined:

$$w_1^f F_1 + w_2^f F_2 + \ldots + w_n^f F_N = 0 \tag{5.28}$$

$$w_1^p X_1 = w_2^p X_2 = \ldots = w_n^p X_N \tag{5.29}$$

Here the position and force scaling weights w_i^p and w_i^f can be independently utilized to realize desired position or force scaling among the robots. However, scaling is usually utilized because of force limits or limited task spaces. And it might be desirable that the teleoperation system should realize impedance matching regardless

of the scaling in forces and positions. This would mean the perfect reproduction of environmental or operator impedances. To understand impedance matching, where impedance is defined as a mapping from velocity to force, the velocity goals should also be defined:

$$w_1^p \dot{X}_1 = w_2^p \dot{X}_2 = \ldots = w_n^p \dot{X}_N \tag{5.30}$$

with the weights chosen the same as position weights. If impedance matching between the environment and operators is desired the weights for force and position/velocity should be the same: $w_i^p = w_i^f$. This follows from the relationship: $Z_i = \frac{w_i^f F_i}{w_i^p s X_i}$ and if $w_i^p = w_i^f$, we would have $Z_i = \frac{F_i}{sX_i}$ regardless of the weights. This means that as in conventional multilateral teleoperation we would have $Z_1 + Z_2 + \ldots + Z_N = 0$. Hence, regardless of the scaling of forces and positions the impedances will not be scaled. However in some cases, the scaling of impedances might also be necessary. If the weights for position and forces are selected differently impedance scaling is naturally realized. Next, the design of the controllers for various tasks such as position scaling, force scaling or impedance matching will be shown.

5.3.1 Position Scaling

Position scaling among robots might be desirable in cases where the robot workspaces have different dimensions, or where there are certain impedance matching requirements for the operators. Therefore the robots should track each other with some scaling. For this goal, first position values of the robots are scaled according to the weight matrix such that:

$$w_1^p X_1 = w_2^p X_2 = \ldots = w_n^p X_N \tag{5.31}$$

Thus, a scaled acceleration consensus algorithm can be written using the scaled Laplacian matrix as follows:

$$\ddot{x}^{ref} = -K_p L_{w^p} x - K_v L_{w^p} \dot{x} + \frac{1}{M_n}(F_1 + F_2 + \ldots + F_N) \tag{5.32}$$

where the scaled Laplacian L_{w^p} weights are chosen according to the desired position scaling:

$$w_i = w_i^p \tag{5.33}$$

Using the disturbance observers and sensitivity functions from chapter 2 this algorithm can be written as:

$$-\frac{1}{M_n} G(s) F_{dis} + \ddot{x} = -K_p L_{w^p} x - K_v L_{w^p} \dot{x} + \frac{1}{M_n}(F_1 + F_2 + \ldots + F_N) \tag{5.34}$$

If the disturbance estimation and compensation is successful, the sensitivity function $G(s)$ becomes zero. Thus the system dynamics is transformed into consensus dynamics. Stability of accelaration consensus algorithm also implies the overall stability of the multilateral teleoperation system. Because the inputs to all the robots are the same; $u_1 = u_2 = \ldots = u_N = \frac{1}{M_n}(F_1 + F_2 + \ldots + F_n)$, and the initial conditions are also same, the scaled acceleration consensus values will immediately become:

$$\begin{aligned} w_1 \ddot{x}_1 &= \ldots = w_N \ddot{x}_N \\ &= \frac{1}{N M_n} 1 1^T W (F_1 + F_2 + \ldots + F_N) \end{aligned} \tag{5.35}$$

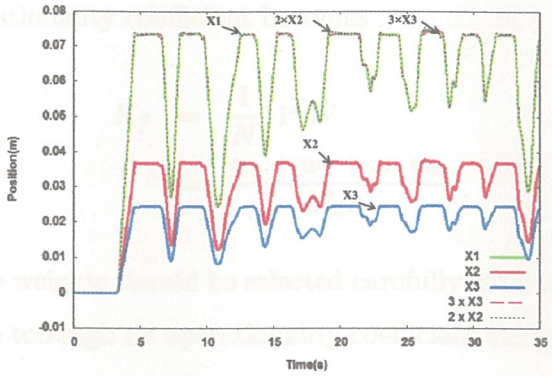


Figure 5.2: Position response of position scaled robots

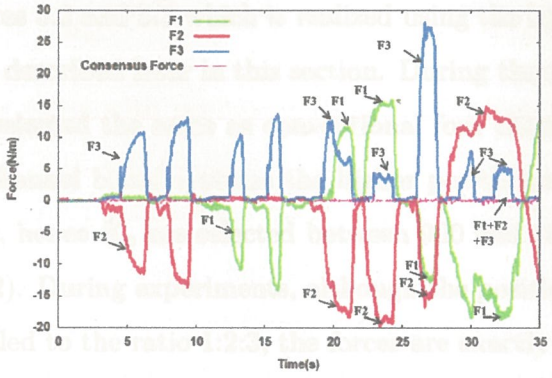


Figure 5.3: Force response of position scaled robots

Also the velocities and positions will be scaled:

$$w_1\dot{x}_1 = w_2\dot{x}_2 = \ldots = w_N\dot{x}_N \tag{5.36}$$

$$w_1x_1 = w_2x_2 = \ldots = w_Nx_N \tag{5.37}$$

Thus the haptic systems will reach scaled acceleration, velocity and position consensus. It can be seen that the weights have an effect on the force gain of the system. Actually it is possible to see that the weights change the operationality of the system

such that, the operationality coefficient becomes :

$$K_f = \frac{1}{N} 1 1^T W \tag{5.38}$$

$$= \frac{w_1 + w_2 + \ldots w_n}{M_n N} \tag{5.39}$$

This means that the weights should be selected carefully so as to make sure that the system doesn't have too high an operationality coefficient that can cause instability. The weights can be selected such that $K_f = 1$, as in conventional four channel teleoperators. Experiment results for position scaling on three identical 1DOF robots can be seen on figures 5.2 and 5.3 which is realized using the haptic consensus algorithm which will be described later in this section. During the experiments position control gains were selected the same as conventional four channel based teleoperators[2][5]. In four channel based systems the higher position control gain provides better transparency, hence K_p are selected between 900 and 1500. K_v is chosen to at least satisfy (5.22). During experiments, although the positions of the robots are always precisely scaled to the ratio 1:2:3, the forces are exactly reflected among the robots, and impedance matching is not realized.

5.3.2 Force Scaling

Scaling the forces of the robots can be realized seperately from the position scaling. In some applications it might be desired to scale the force inputs; such as during haptic instruction or collaboration, so that the sum of the scaled forces should equal:

$$w_1^f F_1 + w_2^f F_2 + \ldots + w_n^f F_N = 0 \tag{5.40}$$

In order to realize this, the acceleration feedforward is modified by using weights on the forces:

$$u_1 = u_2 = \ldots = u_N \tag{5.41}$$

$$= \frac{1}{M_n}(w_1^f F_1 + w_2^f F_2 + \ldots + w_n^f F_N) \tag{5.42}$$

and the controller becomes:

$$\ddot{x} = -K_p L_{w^p} x - K_v L_{w^p} \dot{x}$$

$$+\frac{1}{M_n}(w_1^f F_1 + w_2^f F_2 + \ldots + w_n^f F_N) \tag{5.43}$$

Force scaling results along with position scaling can be seen in figure 5.4,5.5. As the scaling ratios for force and position are chosen to be 1:2:3, perfect impedance matching is realized, which means that the feeling of the environment is the same, even though the motions are not. Furthermore, it is possible to realize position and force scaling by using distributed force consensus filters. One big disadvantage of multilateral controllers that do not use consensus algorithms is that, all the robot force data must be transmitted among the robots. However this can not be realized when there are communication constraints on the network, or some of the communication channels among the robots are missing. By using a consensus algorithm and a consensus filter it is possible to realize haptic consensus which guarantees that the position values are scaled and the scaled accelerations of the systems track common force reference, even under changing network topologies. It was previously defined that the common force reference is the sum of the forces $w_1 F_1 + w_2 F_2 + \ldots + w_N F_N$. It is possible to obtain a similar force reference, using force scaled consensus filters which give the average of the scaled forces applied on the robots. In effect this is a kind of distributed force sensor fusion. The forces measured at every node locally can be used as the inputs to a high pass consensus filter with states v:

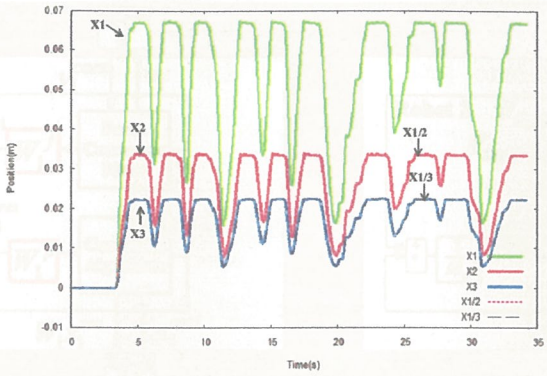


Figure 5.4: Position response of position and force scaled robots

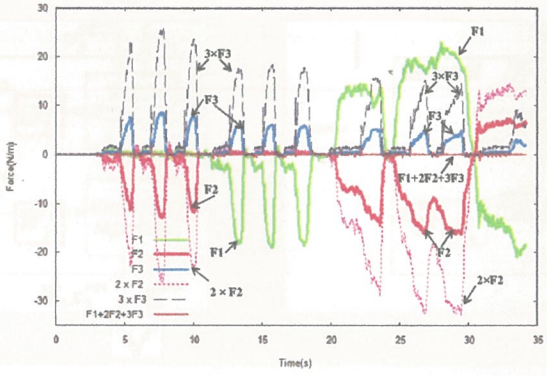


Figure 5.5: Force response of position and force scaled robots

$$\dot{v}_i = \sum_{j \in N_i} a_{ij}(v_j - v_i) + w_i^f F_i \tag{5.44}$$

It is known that the derivative of consensus filter states v_i estimate the average of the inputs which are scaled reaction forces measured locally at every node. As this is a high pass consensus filter defined in section 1, at steady state the derivative of

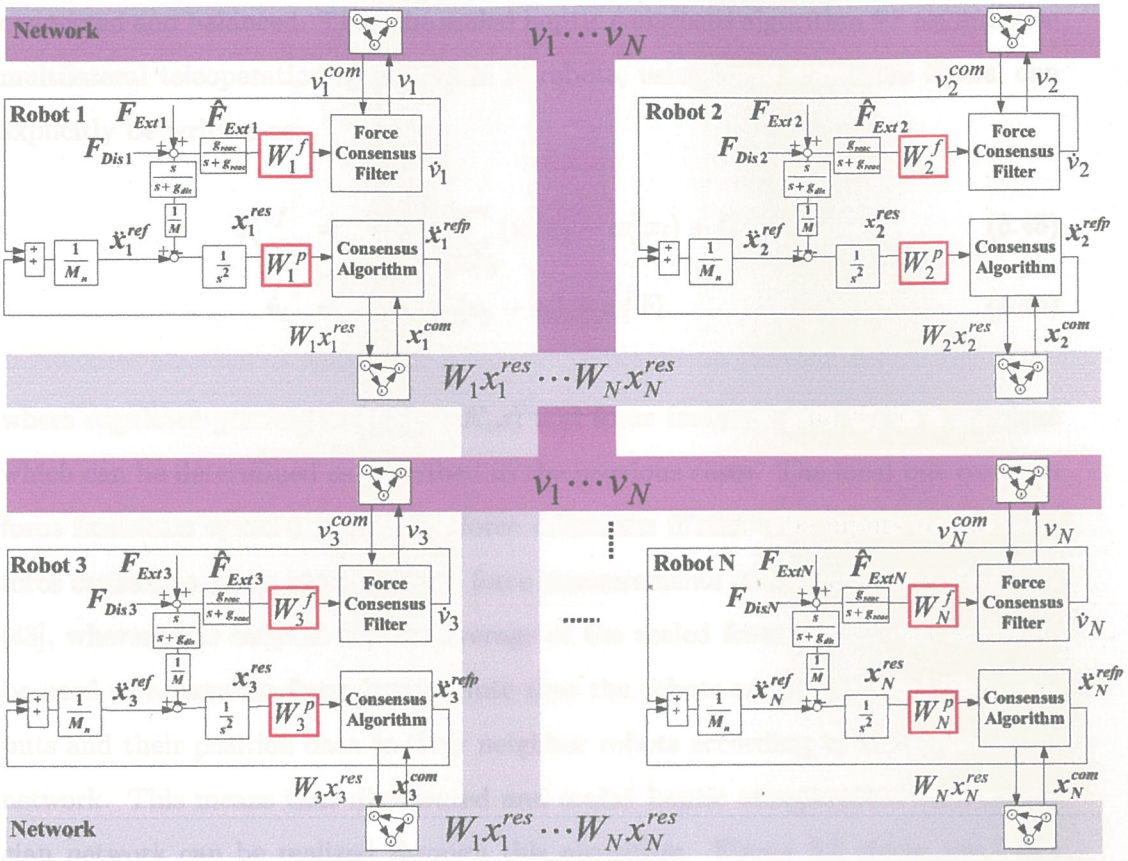


Figure 5.6: Scaled haptic consensus with force consensus filters

the filters converge to:

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_N \end{bmatrix} = \frac{1}{N} \begin{bmatrix} w_1^f & w_2^f & \dots & w_N^f \\ w_1^f & w_2^f & \dots & w_N^f \\ \vdots & \vdots & \ddots & \vdots \\ w_1^f & w_2^f & \dots & w_N^f \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix}$$

Thus if the average is multiplied with the number of robots N sum of the scaled forces can be obtained. For this the only condition is that the network is strongly

connected and balanced. Then the scaled haptic consensus algorithm for asymmetric multilateral teleoperation system with N robots, using force consensus filters, can explicitly be written as:

$$\ddot{x}_i^{ref} = \frac{C_p(s)}{w_i^p} \sum_{j \in N_i} (w_j^p x_j - w_i^p x_i) + C_f \frac{N}{M_n} \dot{v}_i \tag{5.45}$$

$$\dot{v}_i = \sum_{j \in N_i} a_{ij}(v_j - v_i) + w_i^f F_i \tag{5.46}$$

where regulator gain $C_p = \frac{1}{2}(K_p + K_v s)$ and force feedback gain C_f is a constant which can be determined as described in the previous cases. The local but common force feedbacks $\dot{v}_i$ are the states of force consensus filters. The inputs to the local force consensus filters are the scaled force measurements of reaction force observers [43], whereas the outputs are the average of the scaled forces which can naturally be used as a common force input. Note that the robots send their local filter outputs and their position data to their neighbor robots according to the graph of the network. This means that distributed and scaled haptic teleoperation on a Laplacian network can be realized through this algorithm. Figure 5.6 shows the block diagram for this algorithm whereas figures 5.2-5.5 show the asymmetric multilateral teleoperation experiments realized by the scaled haptic consensus algorithm, on a dynamically changing network.

5.3.3 Multi Robot Micro-Macro Teleoperation

It is possible to realize micro-macro teleoperation with multiple different size robots using the scaling ratios. The accelerations, velocities, positions and forces of the robots should be scaled down or up according to the size of the robots. If the robot is 1DOF, it is easy to determine the scaling of the robot according to the mass or the length. If the robot is smaller, it has to have a bigger scaling ratio. Therefore

the inverses of the masses are selected as the scaling ratios:

$$\begin{bmatrix} w_1^p \\ \vdots \\ w_N^p \end{bmatrix} = \begin{bmatrix} w_1^f \\ \vdots \\ w_N^f \end{bmatrix} = \begin{bmatrix} M_1^{-1} \\ \vdots \\ M_N^{-1} \end{bmatrix} \tag{5.47}$$

It becomes possible to realize micro-macro teleoperation through haptic consensus using the following algorithm:

$$\ddot{x}_i^{ref} = C_p(s) \sum_{j \in N_i} (M_j^{-1} x_j - M_i^{-1} x_i) + C_f N \dot{v}_i \tag{5.48}$$

$$\dot{v}_i = \sum_{j \in N_i} a_{ij}(v_j - v_i) + M_i^{-1} F_i \tag{5.49}$$

It follows from the previous subsection that the scaled consensus can be realized as:

$$M_1^{-1} F_1 + M_2^{-1} F_2 + \ldots + M_N^{-1} F_N = 0 \tag{5.50}$$

$$M_1^{-1} X_1 = M_2^{-1} X_2 = \ldots = M_N^{-1} X_N \tag{5.51}$$

enabling multirobot micro-macro teleoperation.

Experiments were realized using three 1 DOF robots connected to each other through a computer (figure 5.7), which can be seen on figures 5.8 and 5.9. Figures

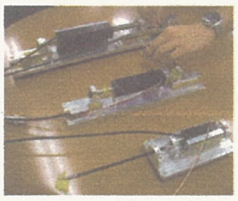


Figure 5.7: The experiment setup for micro-macro multilateral teleoperation

show the experiment results for scaled teleoperation on a hub network structure

where the robots 2 and 3 do not send any information to each other and are using robot 1 as a hub. In the experiments, the masses of the robots are $M_1 = 0.1kg$, $M_2 = 0.3kg$ and $M_3 = 0.5kg$. Therefore their outputs are scaled as 1:3:5 according to their masses. Figure 5.8 shows the position responses of the robots. As can be seen, the robot positions are scaled just as desired with ratios 1:3:5. Figure 5.9

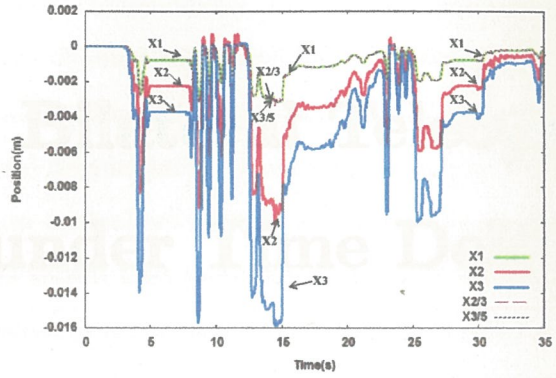


Figure 5.8: Position response of robots

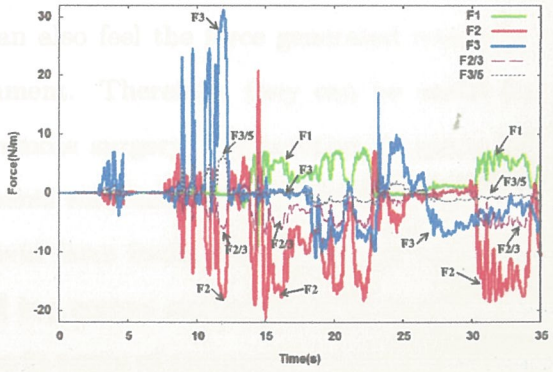


Figure 5.9: Force response of robots

shows the force responses of the robots. The sum of the scaled forces estimated by the consensus filter is zero when the robots are touching an environment. Also it can be observed that the forces are scaled according to the sizes.

Chapter 6

Stable Bilateral Teleoperation under Time Delay

Bilateral teleoperators consist of two robots which are connected by a communication line; the operator can control the remote(slave) robot by manipulating the master robot and can also feel the force generated remotely, when the slave robot contacts an environment. Therefore, they can be useful for a wide range of applications such as remote surgery, construction in space, remote manipulation in hazardous environments such as nuclear plants, where accurate and precise motion control as well as vivid force feedback is necessary. Four channel teleoperation architecture[26][27][30] is a control strategy devised to attain these goals and provides the best performance in terms of transparency; due to the transmission of force and position information from both robots, perfect position tracking and force reflection can be realized. A teleoperation system may work perfectly under conditions where there is no delay between the robots, however introduction of small delays of even 50-100ms can render the system totally useless and even dangerous. It has been shown that this performance optimized structure is the least robust against time delays[45]. Therefore, until now most of the research in time delayed teleopera-

tion has been concentrated on passivity based two channel teleoperation schemes which depend on scattering transformation[23] and wave variables[24]. Although these methods guarantee the overall stability of the system, they do not provide a good enough solution in terms of transparency. There are problems such as position drift or, the lack of high frequency force feedback, which limits the capabilities of the teleoperation system tremendously. In [32],[33] communication disturbance observer based and three channel based architectures were proposed for transparency in delayed teleoperation. However, these methods do not guarantee the stability of the overall system and depend on the careful operation by the operator. In chapter 3, robust acceleration controllers with local velocity feedback or damping was introduced, and it was shown that these algorithms are stable under the presence of time delay. Furthermore four channel based controllers using acceleration consensus was shown in chapter 4 and generalized in chapter 5. Based on these results, in this chapter, the classical four channel controller realized by acceleration consensus algorithms is augmented by injecting local damping which is obtained by local velocity feedback. A similar idea of damping injection in teleoperation was presented for nonlinear systems in [28] and [29] based on two channel controllers. Although the approach in [28] guarantees stable teleoperation, force tracking can be realized only in steady state and the system response is very slow. This means that there is no high frequency force feedback and the transparency is low. The approach in [29] extends the method of [28] with some sort of force feedback, however because only one robot receives position feedback, the position tracking error can be very big. In this research, by adopting the four-channel teleoperation and using local damping injection it is shown that stable teleoperation with considerably good transparency-good position tracking and high frequency force feedback- can be realized. It is generally difficult to show the stability of four channel controllers, because of the complex structure, which is further complicated by the introduction of the time

delay elements. However, four channel architectures need linear models/linearized structures to be realized. And in this paper the linear time invariance(LTI) property of four channel architecture is exploited and by transforming the four channel system into input output[50] form, the L_2[48]-[50] stability is analyzed. In [52], L_2 stability was first employed for the stability analysis of bilateral telerobots in contact with environment. Bilateral teleoperation independent of time delay was shown using the scattering parameters in [49]. In [53], L_2 stability of four channel teleoperators was investigated, however that analysis is not complete because it assumes all the forces applied on the system independent of the system states. This is not the case, when contact with environment is realized. In this research, this fact is also considered and by using a linear mass damper analogy for the environment and operator like in the analyses presented in [31][58][60][61][62], stability of the proposed method is shown. Because the proposed analysis is based on delay independent L_2 stability it is shown that the stability depends on controller parameters and the impedance parameters of the operator and the environment. The four-channel architecture is known to be not L_2 stable in most cases except when the environment is very soft with high damping[27][52]. The proposal in this chapter actually emulates this soft environment effect by locally injecting the damping to the robots. It is shown that by damping injection the four channel teleoperation architecture can be made L_2 stable. As L_2 stability is concerned with the energy of the systems and used to show the stability of dissipative systems, it is a very intuitive choice to show the stability of damping injection methods which dissipate energy. On the other hand, because stability and transparency are conflicting goals[27], the drawback of the proposed method is that there is constant damping in the free motion. However, position tracking is improved and high frequency force reflection can be realized because of the dual force feedback. In this chapter transparency analysis for the proposed system is provided and it demonstrates the effect of the damping on the system

transparency. The analysis is also confirmed with experiment results, using 1DOF robots, which can be utilized in various applications such as robotic surgery[34].

6.1 Four channel architecture and damping injection method

For bilateral teleoperation, we will assume two identical linear robots as modeled in chapter 2, which employ disturbance observers. Hence the dynamics of the robots will be double integrator dynamics as the disturbances such as friction, damping and interactive forces and gravity are compensated by disturbance observers. The robots will employ acceleration tracking controllers and the position and force control will be established between the robots following the four channel methodology which will be explained in the next subsection.

6.1.1 Four channel teleoperation

The goal in bilateral teleoperation is that the master and slave robots should track the same position x and the forces F acting on the robots should be reflected:

$$x_m = x_s \tag{6.1}$$

$$F_h = -F_e \tag{6.2}$$

This is called kinaesthetic coupling between the master and the slave systems, and means that the teleoperator is transparent. To attain these goals, the 4 channel teleoperation architecture has been proposed independently by Anderson[27], and Yokokohji[26]. The block diagram of a 4 channel teleoperation system employing disturbance observers can be seen in figure 6.1. In the figure, C_1 and C_4 represent the communication/control channels for position between the master and slave robots, whereas C_2 and C_3 represent the communication/control channels for force

Table 6.1: List of variable definitions

F	Force
x	Position
$\dot{x}$	Velocity
$\ddot{x}$	Acceleration
M/m	Mass
C_p	Position regulator
K_p	Proportional gain
K_v	Differential gain
C_f	Force control coefficient
g_{dis}	Cut–off frequency of disturbance observer
g_{reac}	Cut–off frequency of reaction force estimation
s	Laplace operator
w	Frequency
k	Stiffness coefficient
b	Damping
D	Viscous friction coefficient
Z	Impedance
T	Roundtrip time delay
T_1	Delay from master to slave
T_2	Delay from slave to master
I_a	Input current
K_t	Motor thrust constant
K_{tn}	Nominal motor thrust constant
F_l	Load disturbance force
F_{ext}	External disturbance force
F_{int}	Interactive disturbance force
f	coulomb friction
(superscript) cmd	Command value
(superscript) ref	Reference value
(superscript) $\hat{}$	Estimated value
(subscript) m	Master system
(subscript) s	Slave system
(subscript) n	Nominal value
(subscript) h	Human
(subscript) e	Environment
(subscript) dis	Disturbance

between the robots. These four channels enable the transmission of the 4 important variables on the communication line. On the other hand there are local controllers implemented on the robots themselves C_m and C_s respectively for the master and the slave. $\hat{F}_h$ and $\hat{F}_e$ terms are the estimated forces from human and environment respectively, by using RFOBs. F_{Dism} and F_{Diss} are the individual disturbance forces acting on the master and slave respectively, and are compensated by disturbance observers. It is possible realize transparent bilateral teleoperation by choosing the transfer functions for these channels and the local controllers as:

$$C_1 = C_4 = C_m = C_s = C_p(s) = K_p + K_v s \tag{6.3}$$

$$C_2 = C_3 = C_f \tag{6.4}$$

Note that the position control channels C_1 and C_2 are selected as the same PD controller C_p, and are the same with the local controllers C_m and C_s. On the other hand the force control channels C_2 and C_3 are selected as a constant coefficient C_f, which is normally taken as 1 but can be increased for improved operationality. Following these selections of the channels and controllers a traditional 4 channel controller can be realized using acceleration controllers[30]:

$$\ddot{x}_m^{ref} = C_p(s)(x_s - x_m) - C_f(\hat{F}_h + \hat{F}_e) \tag{6.5}$$

$$\ddot{x}_s^{ref} = C_p(s)(x_m - x_s) - C_f(\hat{F}_h + \hat{F}_e) \tag{6.6}$$

which is actually a simple consensus algorithm of the form:

$$\ddot{x}_i = \sum_{j \in N_i} C_p(s)(x_j - x_i) + C_f \sum_i u_i$$

$$\ddot{x} = -C_p L x + C_f 1 1^T u$$

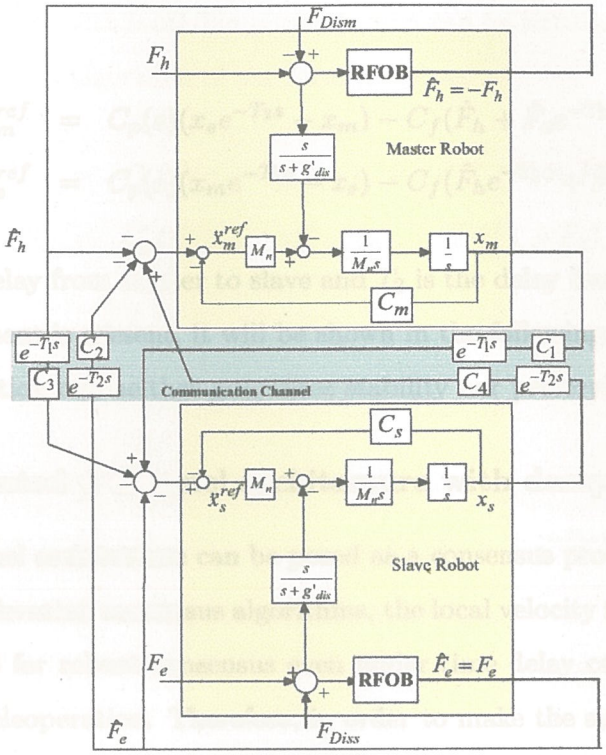


Figure 6.1: Four channel teleoperation architecture with disturbance and reaction force observer

where there will be only one neigboring node j for each i representing the master and the slave, the sum of the inputs will be the sum of the forces: $\sum_i u_i = -C_f(\hat{F}_h + \hat{F}_e)$ and the Laplacian is a two by two matrix:

$$L = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6.7}$$

The four channel architecture is a performance optimized architecture which means that it provides the ideal transparency or kinaesthetic coupling. However, when there is time delay present on the communication channels the control input to the

robots become:

$$\ddot{x}_m^{ref} = C_p(s)(x_s e^{-T_2 s} - x_m) - C_f(\hat{F}_h + \hat{F}_e e^{-T_2 s}) \quad (6.8)$$

$$\ddot{x}_s^{ref} = C_p(s)(x_m e^{-T_1 s} - x_s) - C_f(\hat{F}_h e^{-T_1 s} + \hat{F}_e) \quad (6.9)$$

where T_1 is the delay from master to slave and T_2 is the delay from slave to master. As the delay element is present, it will be shown in the following sections that four channel teleoperation can neither guarantee stability nor preserve transparency.

6.1.2 Augmented 4 channel architecture with damping injection

As the four channel architecture can be posed as a consensus problem, and be realized through acceleration consensus algorithms, the local velocity feedback approach used in chapter 3 for robust consensus even under time delay can also be utilized in four channel teleoperation. Therefore, in order to make the system more robust against time delays, local damping injection is applied to the robots as a function of their velocity by taking the 4 Channel gains in figure 6.1 as:

$$C_1 = C_4 = C_p(s) = K_p + K_v s \quad (6.10)$$

$$C_2 = C_3 = C_f \quad (6.11)$$

$$C_m = K_p + K_v s + b_m s \quad (6.12)$$

$$C_s = K_p + K_v s + b_s s \quad (6.13)$$

Note that damping injection terms b_m,b_s do not show up in the communication channel gains, which means that they are purely local. The local damping injection terms have a stabilizing effect unlike the K_v terms which can not guarantee stability on their own as it will be shown in the next section. It will also be shown in section 6.3 that, when there is time delay, and local damping injection is present, K_v becomes unnecessary, as it degrades the system transparency. In general a bilateral

acceleration controller with local damping injection can be written by using a robust acceleration consensus algorithm of the form (3.65) as:

$$\ddot{x}_m^{ref} = C_p(s)(x_s - x_m) - b_m\dot{x}_m - C_f(\hat{F}_h + \hat{F}_e) \tag{6.14}$$

$$\ddot{x}_s^{ref} = C_p(s)(x_m - x_s) - b_s\dot{x}_s - C_f(\hat{F}_h + \hat{F}_e) \tag{6.15}$$

so that in the presence of time delay, the damping terms are unaffected from the delay:

$$\ddot{x}_m^{ref} = C_p(s)(x_s e^{-T_2 s} - x_m) - b_m\dot{x}_m - C_f(\hat{F}_h + \hat{F}_e e^{-T_2 s}) \tag{6.16}$$

$$\ddot{x}_s^{ref} = C_p(s)(x_m e^{-T_1 s} - x_s) - b_s\dot{x}_s - C_f(\hat{F}_h e^{-T_1 s} + \hat{F}_e) \tag{6.17}$$

The small gain theorem and the concept of L_2 stability will be utilized to compare the stability of a conventional four channel teleoperation system and the proposed four channel architecture with damping injection in the following section.

6.2 Stability analysis

6.2.1 L_2 Stability and small gain theorem

First the definition of the terms that will be use in the analysis will be provided, mainly following [48]. The L_2 signal spaces consist of all the functions $f: \Re^+ \rightarrow \Re$ which are measurable and satisfy:

$$\int_0^\infty |f(t)|^2 \, dt < \infty \tag{6.18}$$

Where the norm of a function in L_2 space is given as:

$$||f||_2 = \left(\int_0^\infty (|f(t)|^2 \, dt) \right)^{1/2} \tag{6.19}$$

A truncation of f to the interval $[0, T]$is the function $f_T : \Re^+ \rightarrow \Re$ defined by:

$$f_T(t) = \begin{cases} f(t) \ , & 0 \leq t < T \\ 0 \ , & t \geq T \end{cases}$$

And the extended L_2 space -L_{2e}- consists of all measurable functions $f : \Re^+ \rightarrow \Re$ such that $f(t) \in L_2$ for all real T with $0 \leq T < \infty$. Now the L_2 stability will be defined; let $S : L_{2e}(U) \rightarrow L_{2e}(Y)$ with input U and output Y. Then S is said to be an L_2 stable input output mapping if

$$u \in L_2(U) \Rightarrow S(u) \in L_2(Y) \tag{6.20}$$

that is, S maps the subset $L_2(U) \subset L_{2e}(U)$ into the subset $L_2(U) \subset L_{2e}(U)$. Furthermore S is said to have a finite L_2 gain if there exist finite constants γ_2 and b_2 such that for all $T \geq 0$

$$||(S(u))_T||_2 \leq \gamma_2 ||u_T||_2 + b_2 \quad \forall u \in L_{2e}(U). \tag{6.21}$$

If S has a finite L_2 gain, this also means that it is L_2 stable. For the input output

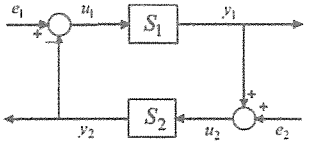


Figure 6.2: Input output connected systems

stability of a closed loop system $\sum_{S1,S2}^{f}$ of the form that can be seen in figure 6.2 which can be described by equations:

$$\begin{aligned} u_1 &= e_1 - y_2, \qquad & u_2 &= e_2 + y_1 \\ y_1 &= S_1(u_1), \qquad & y_2 &= S_2(u_2) \end{aligned}$$

where S represents input output mapping of the systems, e represents exogenous inputs which are independent of system states, u the system inputs and y the system outputs. The small gain theorem states that if both S_1 and S_2 have finite L_2 gains $\gamma_2(S_1)$ and $\gamma_2(S_2)$ respectively and are stable, and also if the multiplication of their gains satisfy

$$\gamma_2(S_1) \cdot \gamma_2(S_2) < 1 \tag{6.22}$$

the closed loop system has a small loop gain and is also L_2 stable. Note that the closed loop system is robustly stable against all perturbations in the input-output maps as long as the small gain condition is satisfied. On the otherhand, although it guarantees stability, it does not give the solutions for the equations.

6.2.2 L_2 stability of four channel teleoperation

In order to show the L_2 stability of four channel teleoperation system, it should be transformed into single loop form. Because the part of the forces generated during contact are functions of the system states, it is not possible to discard the impedances of the environment and the human in this analysis. Therefore the forces exerted on the system from the environment and the operator can be written as:

$$\textit{Operator} \quad : \quad F_h = F_h^* - Z_h \dot{x}_m \tag{6.23}$$

$$\textit{Environment} \quad : \quad F_e = F_e^* + Z_e \dot{x}_s \tag{6.24}$$

where F_h^* and $F_e^*(=0)$ are the exogenous input forces which mean that these forces are generated independent of the robot states, by the operator or the environment. On the other hand the environment and human impedances Z also map the system states(velocity) to forces with the parameters[31]:

$$Z_e = \frac{k_e}{s} + b_e + m_e s \tag{6.25}$$

$$Z_h = \frac{k_h}{s} + b_h + m_h s \tag{6.26}$$

where the k are the stiffness terms, b are the damping terms, and m are the masses. These linear impedance models for human and environment are widely used in the haptic teleoperation research [58,60,61,62] for showing stability. It is assumed that the robots are in perfect coupling with the operator or the environment, such that the velocity of the robot is the velocity of the human hand, etc... It is also assumed that the reaction force observer in this system works ideally such that $\hat{F}_h = -F_h, \hat{F}_e = F_e$ and it is possible to transform the estimated forces into system states using these impedance parameters. Note that there is a negative sign in the human force; this is because the operator creates a disturbance in the direction of the motion, which is estimated as a negative disturbance. For the L_2 stability, it is further assumed as in previous works[27], that exogenous forces are in L_2 space and are bounded. Also because, disturbance observers are used in the system, and the acceleration references are multiplied with the nominal masses, the effects of the robot masses disappear from the transfer functions as they are nominalized to 1. Hence, by using these velocity force mappings in frequency domain it is possible to obtain the four channel teleoperation architecture in input output form as in figure 6.3:

Here the systems S_1 and S_2 are input output connected and represent the master and slave respectively. Note that the time delay element is also a part of these systems. The exogenous inputs e are the cumulative exogenous components F^* of

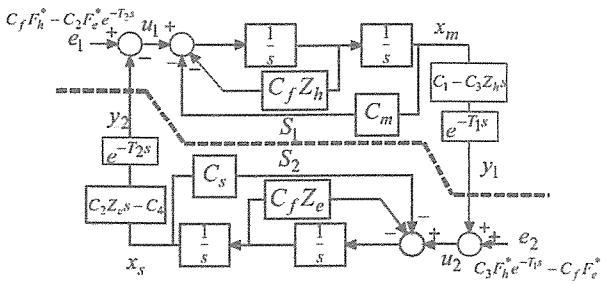


Figure 6.3: Input output transformation of the four channel teleoperation architecture

the forces F_e and F_h, which are independent of the system states. These cumulative exogenous forces show up in the system transfer functions when forces in 30-31 are inserted into the force feedforward terms; $-(C_f\hat{F}_h + C_2\hat{F}_e e^{-T_2 s})$, $-(C_3\hat{F}_h e^{-T_1 s} + C_f\hat{F}_e)$; with the assumption that $\hat{F}_h = -F_h, \hat{F}_e = F_e$. The same transformation was given in [27] and [31]. The overall equations for the system can be written in frequency domain as:

$$\begin{aligned} e_1 &= C_f F_h^* - C_2 F_e^* e^{-T_2 s}, \quad e_2 = -C_f F_e^* + C_3 F_h^* e^{-T_1 s} \\ u_1 &= e_1 - y_2, \qquad\qquad u_2 = e_2 + y_1 \\ y_1 &= S_1(u_1), \qquad\qquad y_2 = S_2(u_2) \end{aligned}$$

$$S_1 = \frac{(C_1(s) - C_3 Z_h(s)s)e^{-T_1 s}}{s^2 + C_f Z_h(s)s + C_m(s)} \tag{6.27}$$

$$S_2 = \frac{(C_2 Z_e(s)s - C_4(s))e^{-T_2 s}}{s^2 + C_f Z_e(s)s + C_s(s)} \tag{6.28}$$

The L_2 gains of the systems in frequency domain are the maximum gains of the systems for all $w > 0$:

$$\gamma_2 = \sup_{w>0} |S(jw)| = \sup_{w>0} \sqrt{S(-jw)S(jw)} \tag{6.29}$$

And the gains of each system are:

$$|S_1| = \sqrt{\frac{(C_1(s) - C_3 Z_h(s)s)e^{-T_1 s}}{s^2 + C_f Z_h(s)s + C_m(s)} \cdot \frac{(C_1(-s) + C_3 Z_h(-s)s)e^{T_1 s}}{s^2 - C_f Z_h(-s)s + C_m(-s)}} \quad (6.30)$$

$$|S_2| = \sqrt{\frac{(C_2 Z_e(s)s - C_4(s))e^{-T_2 s}}{s^2 + C_f Z_e(s)s + C_s(s)} \cdot \frac{(-C_2 Z_e(-s)s - C_4(-s))e^{T_2 s}}{s^2 - C_f Z_e(-s)s + C_s(-s)}} \quad (6.31)$$

If the traditional 4 channel controllers are selected as in 6.3,6.4, these norms can be written as a function of the frequency w, the controller parameters and the impedance parameters :

$$|S_1| = \sqrt{\frac{(K_p - C_f k_h)^2 + 2(K_p - C_f k_h)(m_h)w^2 + (K_v - C_f b_h)^2 w^2 + (C_f m_h w^2)^2}{(K_p + C_f k_h)^2 - 2(K_p + C_f k_h)(C_f m_h + 1)w^2 + (C_f b_h + K_v)^2 w^2 + (C_f m_h + 1)^2 w^4}} \quad (6.32)$$

$$|S_2| = \sqrt{\frac{(C_f k_e - K_p)^2 + 2(C_f k_e - K_p)(m_e)w^2 + (C_f b_e - K_v)^2 w^2 + (C_f m_e w^2)^2}{(K_p + C_f k_e)^2 - 2(K_p + C_f k_e)(C_f m_e + 1)w^2 + (C_f b_e + K_v)^2 w^2 + (C_f m_e + 1)^2 w^4}} \quad (6.33)$$

If $|S_1(w)| \cdot |S_2(w)| < 1, \forall w > 0$, the four channel connection will be stable according small gain theorem. It can be clearly seen that as constant time delay is an expanding operator, the gain of the systems are independent of time delay. Hence the L_2 stability of the four channel teleoperators is independent of time delay on the channels. It is rather dependent on the impedance parameters and the controller parameters. Now it will be shown why the four channel teleoperator architecture is not L_2 stable. First, for simple demonstration the least complicated case of free motion will be taken into consideration, where it can be assumed that the input to the robots are exogenous, and the human and environmental impedance parameters can be neglected. Even in this simple case the four channel system will be inherently unstable as the system norms are:

$$|S_1| = |S_2| = \sqrt{\frac{K_p^2 + K_v^2 w^2}{K_p^2 - 2K_p w^2 + K_v^2 w^2 + w^4}} \quad (6.34)$$

Note that for the gain to be less than 1 for all $w > 0$, the denominator of the norms

given above must be less than the numerator for all $w > 0$. It can immediately be seen that this condition will not be satisfied for $w^2 < 2K_p$. The maximum gain occurs at the resonance frequency determined by the nominal masses and the control gains K_p in the free motion case. This problem can be easiliy tackled by introducing damping only in the denominator of the gains, by making the system at least critically damped. If local damping injection is introduced as given in equations 6.11,6.12; this is exactly what can be achieved. Thus the systems will have the gains:

$$|S_1| = \sqrt{\frac{K_p^2 + K_v^2 w^2}{K_p^2 - 2K_p w^2 + (K_v + b_m)^2 w^2 + w^4}} \tag{6.35}$$

$$|S_2| = \sqrt{\frac{K_p^2 + K_v^2 w^2}{K_p^2 - 2K_p w^2 + (K_v + b_s)^2 w^2 + w^4}} \tag{6.36}$$

where the damping injection terms b_m and b_s are only in the denominator unlike the K_v term, and this damping term will indeed be instrumental in making the gains less then by 1, if the damping terms are chosen to be the same b:

$$b_m = b_s = b \geq \sqrt{2K_p} \tag{6.37}$$

which will yield the gain equations:

$$|S_1| = |S_2| \leq \sqrt{\frac{K_p^2 + K_v^2 w^2}{K_p^2 + K_v^2 w^2 + 2K_v b w^2 + w^4}} < 1\,, \quad \forall w > 0 \tag{6.38}$$

satisfying small gain condition:

$$\sup_{w>0}(|S_1|) \sup_{w>0}(|S_2|) < 1 \tag{6.39}$$

and making the closed loop system stable. On the other hand when the slave system is in contact with the envionment we have to take into consideration the human and

environmental impedance parameters as the forces generated will be functions of the system states; thus affecting the system stability. The most generalized system gains with the local damping injection applied can be written as:

$$|S_1|=\sqrt{\frac{(K_p-C_fk_h)^2+2(K_p-C_fk_h)(C_fm_h)w^2+(K_v-C_fb_h)^2w^2+(C_fm_hw^2)^2}{(K_p+C_fk_h)^2-2(K_p+C_fk_h)(C_fm_h+1)w^2+(C_fb_h+b_m+K_v)^2w^2+(C_fm_h+1)^2w^4}} \tag{6.40}$$

$$|S_2|=\sqrt{\frac{(C_fk_e-K_p)^2+2(C_fk_e-K_p)(m_e)w^2+(C_fb_e-K_v)^2w^2+(C_fm_ew^2)^2}{(K_p+C_fk_e)^2-2(K_p+C_fk_e)(C_fm_e+1)w^2+(C_fb_e+b_s+K_v)^2w^2+(C_fm_e+1)^2w^4}} \tag{6.41}$$

with the addition of the damping terms b_m and b_s in the denominators. These terms can be used to compenstate for the minus terms in the denominator without increasing the value of the numerator and it can make the gains less then one if chosen as :

$$b_m \geq 2\sqrt{(K_p + C_fk_h)(C_fm_h + 1)} \tag{6.42}$$

$$b_s \geq 2\sqrt{(K_p + C_fk_e)(C_fm_e + 1)} \tag{6.43}$$

Here, the 2 is intentionally left outside the square root, in order to compensate for the impedance masses in the numerator. The gains can be obtained as:

$$|S_1| \leq \sqrt{\frac{\alpha}{\beta}}, \qquad |S_2| \leq \sqrt{\frac{\chi}{\delta}} \tag{6.44}$$

where

$$\alpha = (K_p - C_fk_h)^2 + 2(K_p - C_fk_h)(C_fm_h)w^2 + (K_v - C_fb_h)^2w^2 + (C_fm_h)^2w^4$$

$$\beta = (K_p + C_fk_h)^2 + 2(K_p + C_fk_h)(C_fm_h + 1)w^2 + (C_fK_v + b_h)^2w^2 +$$

$$2(b_m)(K_v + C_fb_h)w^2 + (C_fm_h + 1)^2w^4$$

$$\chi = (C_fk_e - K_p)^2 + 2(C_fk_e - K_p)(C_fm_e)w^2 + (C_fb_e - K_v)^2w^2 + (C_fm_e)^2w^4$$

$$\delta = (K_p + C_fk_e)^2 + 2(K_p + C_fk_e)(C_fm_e + 1)w^2 + (C_fb_e + K_v)^2w^2 +$$

$$2(b_s)(C_f be + Kv)w^2 + (C_f m_e + 1)^2 w^4$$

It can be clearly seen that as all the constants are non-negative:

$$\alpha < \beta\,, \quad \forall w > 0 \;\Rightarrow\; |S_1(w)| < 1, \quad \forall w > 0 \tag{6.45}$$

$$\chi < \delta\,, \quad \forall w > 0 \;\Rightarrow\; |S_2(w)| < 1, \quad \forall w > 0 \tag{6.46}$$

which will guarantee for all human and environment impedance values that

$$\sup_{w>0}(|S_1|)\sup_{w>0}(|S_2|) < 1 \tag{6.47}$$

which completes the proof. The stability of the system depends on the environmental and human stifnesses, masses and the control parameters, and it is possible for any combination of these parameters, to find a damping injection term as proposed, which will guarantee that the system is delay independent L_2 stable. It can also be seen from these equations that K_v is not necessary to guarantee the stability. Also it should be added that the criteria for stability as shown in equations 6.41,6.42 could even be conservative. If the environmental and operator masses m_e, m_h were assumed to be 0, choosing the damping injection terms according to

$$b_m \geq \sqrt{2(K_p + C_f k_h)} \tag{6.48}$$

$$b_s \geq \sqrt{2(K_p + C_f k_e)} \tag{6.49}$$

would have been adequate. This means that equations 6.41 and 6.42 provide the most general solution to the problem, however if all the parameters of the environment are known beforehand, or can be estimated it may be possible to relax these conditions. There might be various ways to determine these parameters, for instance they can be obtained via online estimation methods such as proposed in[57].

These methods are not within the scope of this paper so some values from the literature will be assumed. For stability analysis the maximum environmental stiffness has been selected as 10000N/m in [58]. On the other hand in [59] , it has been shown that the human arm impedance is upper bounded by the transfer function $Z_e = 300 + 1000/s$ with the stiffness of $1000N/m$ with no mass. The hand mass is also negligible in [60]. Nevertheless it is possible to assume a hand mass of 0.5 kg for robustness. It can also be assumed that the environment is static and therefore does not have a mass component. The damping of the environment and human operator is not a part of the criteria that were proposed in this section, so their values are irrelevant on the choice of damping injection terms, but it is clear from the system gains that human and environmental damping have a stabilizing role. The masses of the robots we use in this research are 0.5 kg, however because disturbance observers are utilized, the mass becomes irrelevant because it is nominalized to 1. If K_p is selected as 2500, and C_f then roughly $b_m > 145, b_s > 160$ should be sufficient. Obviously, precise measurements would be necessary to calculate the minimum necessary damping each time, but in the next section will be presented an interesting result, which implies that the higher the damping injection term is chosen, the better position and force tracking will be. Hence it might be desirable that, the damping injection terms are selected much bigger than the stability criteria presented, which makes the precise determination of damping terms unnecessary. It is hard to obtain tracking or performace results from the small gain theorem. Whether the control performance will be degraded due to the small gain of the system is not obvious. However, it can be said that the system will have a low pass filter like loop gain. Still this can be better in terms of performance than conventional methods that can not satisfy the small gain condition. To further understand the effects of L_2 stabilization in haptic teleoperation systems, transparency analysis in frequency domain will be presented in the next section and comparison with the conventional

methods will also be provided.

6.3 Transparency analysis

As mentioned before, the 4 channel architecture provides almost ideal transparency given that there is no time delay. This can be seen by using the hybrid matrix[25] which is a useful tool in determining the performance characteristics of the teleoperator system, as it represents the relationship between the master and slave forces and velocity responses:

$$\begin{bmatrix} F_h \\ -\dot{x}_s \end{bmatrix} = \underbrace{\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}}_{H} \begin{bmatrix} \dot{x}_m \\ -F_e \end{bmatrix} \tag{6.50}$$

Ideal transparency is defined in [25] with the hybrid parameters; $H_{11} = 0$, $H_{12} = H_{21} = 1$, $H_{22} = 0$. However in reality, the robot inertias show up in the H_{11} unless exact acceleration measurement and feedback among the robots is realized, which is impractical. With acceleration control based four channel control almost perfect transparency is provided by the 4 channel teleoperator under no time delay, which means that the system has $H_{11} = \frac{-s}{C_f}$, $H_{12} = H_{21} = 1$, $H_{22} = 0$[33] where force coefficient C_f can be utilized to decrease the perceived inertia of the system. It is obvious that this is not the case when there is time delay. The hybrid paremeters with delay are very complicated and do not satisfy the ideal conditions as given in [45]:

$$H_{11} = D((s + C_m)(s + C_s) + C_1C_4) \tag{6.51}$$

$$H_{12} = (C_2(s + C_s) - C_4)D \tag{6.52}$$

$$H_{21} = -(C_3(s + C_m) + C_1)D \tag{6.53}$$

$$H_{22} = D(1 - C_2C_3) \tag{6.54}$$

$$D(s) = ((s + C_s) - C_3C_4)^{-1} \tag{6.55}$$

By using the controller for traditional four channel controllers and the damping injection based ones, it is possible to obtain the transparency results using the above given hybrid parameters. In the presence of time delay, a trade-off between stability and performance is necessary. To realize this trade-off, first the transparency performance of the system should be assessed in a quantitave way, i.e by measuring the extreme values of transmitted impedances Z_{to} denoting the impedance transmitted to the operator and Z_{te} denoting impedance transmitted to the environment. In typical four channel controllers the goal is to have impedance matching on master and slave sides; such that the impedance transmitted to the operator equals the environmental impedance: $Z_{to} = Z_e$ and the impedance transmitted to environment equals operator impedance $Z_{te} = Z_h$. These transmitted impedances are[31]:

$$Z_{te} = \frac{H_{11} + Z_h}{(H_{11}H_{22} - H_{12}H_{21}) + H_{22}Z_h} \tag{6.56}$$

$$Z_{to} = \frac{H_{11} + (H_{11}H_{22} - H_{12}H_{21})Z_e}{1 + H_{22}Z_e} \tag{6.57}$$

It is possible to check the bandwidth of the transmitted impedances to quantify the transparency. Ideally when the robots are in free motion they should have no transmitted impedances ($Z_{to} = 0/Z_{te} = 0$), which represents the minimum of the transmitted impedances. Again, ideally when both of the robots are clamped ($Z_{to} \rightarrow \infty/Z_{te} \rightarrow \infty$) the range of impedances that can be transmitted should go to infinity, representing all kinds of environments. Analysis of these values is called the Z-width analysis[45][56]. The dynamic range or Z-width can be easily obtained by inserting the boundary conditions $Z_e, Z_h = 0$ and $Z_e, Z_h \rightarrow \infty$ in equations 6.55-56:

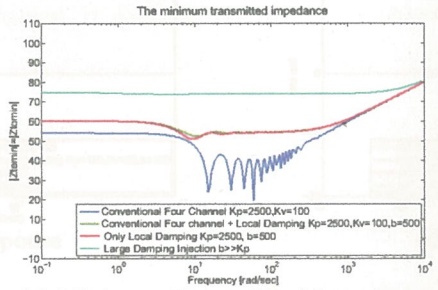
$$Z_{tomin} := Z_{to}|_{Z_e=0} = H_{11} \tag{6.58}$$

$$Z_{temin} := Z_{te}|_{Z_h=0} = \frac{H_{11}}{\Delta h} \tag{6.59}$$

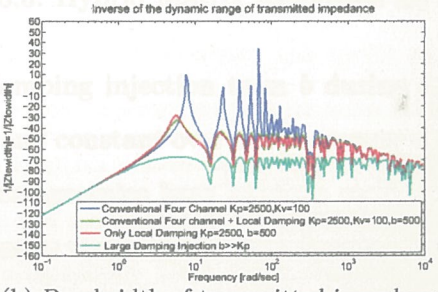
$$Z_{towidth} := Z_{to}|_{Z_e\to\infty} - Z_{tomin} = \frac{-H_{12}H_{21}}{H_{22}} \tag{6.60}$$

$$Z_{tewidth} := Z_{te}|_{Z_h\to\infty} - Z_{temin} = \frac{-H_{12}H_{21}}{H_{22}\Delta h} \tag{6.61}$$

where $\Delta h := H_{11}H_{22} - H_{12}H_{21}$. A high transparency system with maximum bandwith can be obtained if; $|Z_{tomin}| \to 0, |Z_{temin}| \to 0, |\frac{1}{Z_{tewidth}}| \to 0, |\frac{1}{Z_{towidth}}| \to 0$. Figure 6.4a shows the minimum transmitted impedances Z_{to}, Z_{te} for conventional



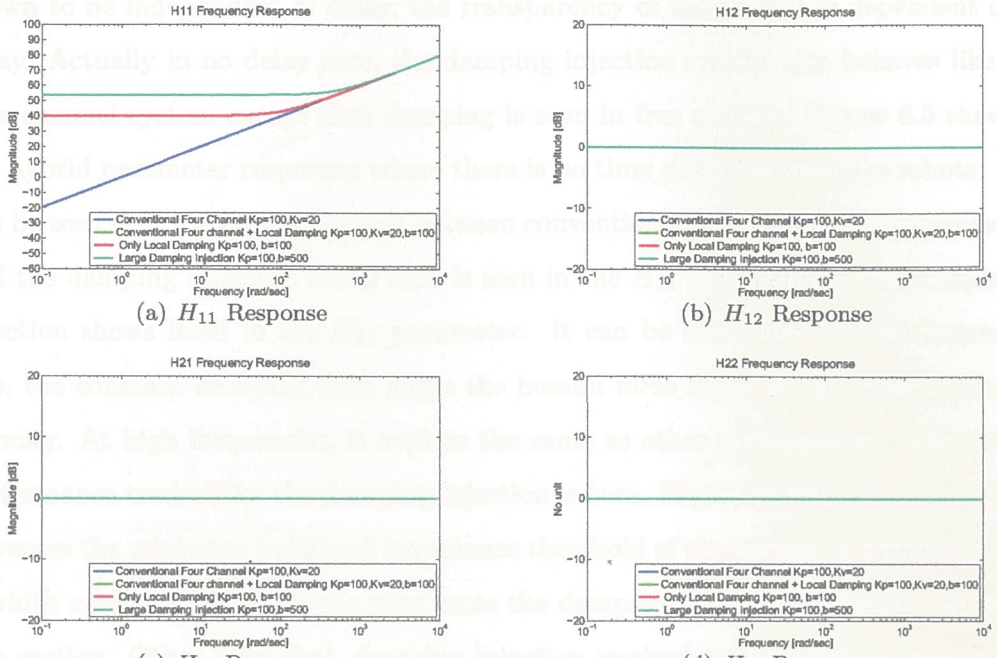
(a) Minimum Transmitted Impedance



(b) Bandwidth of transmitted impedance

Figure 6.4: The Minimum and the bandwith of transmitted impedances

four channel, damping injection with and without channel velocity feedback as well as large damping with a delay of 0.4 seconds. Damping injection terms are selected equal as $b_m = b_s = b$. Note that the transmitted impedances to both sides are equal. As expected, the minimum transmitted impedance for the damping injection case is close to the constant b instead of the ideal 0. This means that the operator feels



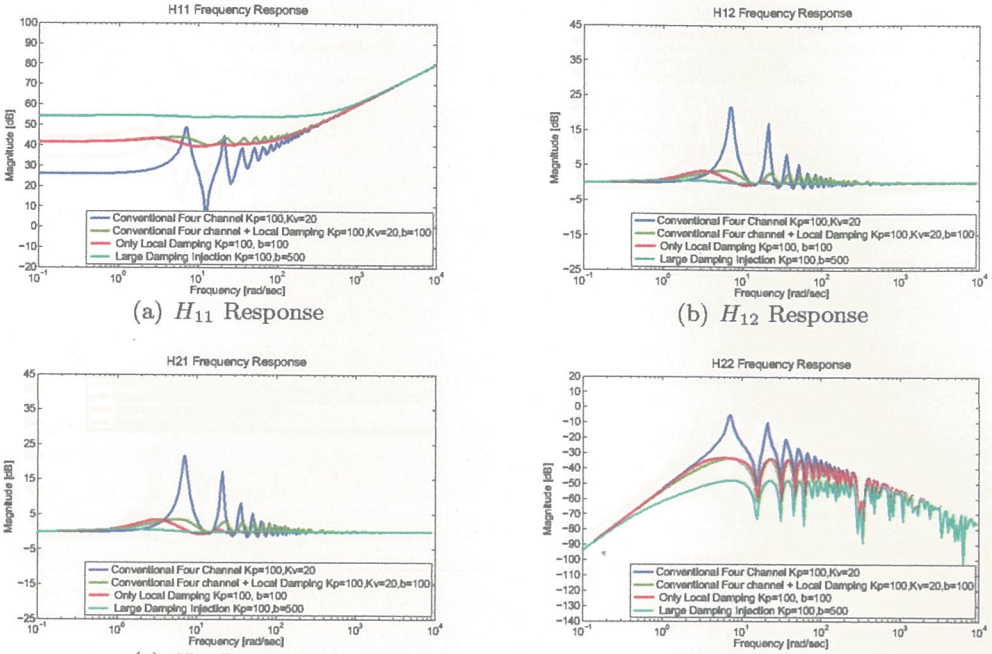
(a) H_{11} Response (b) H_{12} Response

(c) H_{21} Response (d) H_{22} Response

Figure 6.5: Hybrid Matrix Responses for no delay

the impedance of the damping injection term b during free motion. However the behavior is more robust and constant over the frequency domain, compared to other controllers. There is a compromise here between stability and transparency. Figure 6.4-b shows the dynamic range of the impedances, or the maximum impedance that can be transmitted. Again these are equal for both sides. As can be seen, the damping injection 4 channel controller with the greatest damping has the best performance, because it gives the minimum response. This means that a high range of environmental impedances can be transmitted through this architecture meaning high frequency components of rigid surfaces can also be reflected. Figures 6.5-6.9 show hybrid parameter responses in frequency domaing for various time delays and various controllers. These plots also show us how the forces and velocities are reflected in the frequency domain. Note that, although the stability of the system was

shown to be independent of delay, the transparency of the system is dependent on delay. Actually in no delay case, the damping injection system also behaves like a four channel system except that damping is seen in free motion. Figure 6.5 shows the hybrid parameter responses where there is no time delay between the robots. It can be seen that the only difference between conventional four channel teleoperation and the damping injection based ones is seen in the H_{11} parameters. The damping injection shows itself in the H_{11} parameter. It can be seen that for low frequencies, the constant damping term maps the human force input to the master robot velocity. At high frequencies, it evolves the same as other architectures. The only performance tradeoff for the damping injection is here. Higher damping term simply increases the minimum perceived impedance threshold of the system as explained by Z-width analysis. This actually represents the damping felt by the operator during free motion. Other than that, damping injection method enables perfect force and position tracking as can be seen from H_{12} and H_{21} parameters. Also H_{22} parameters are the same, and there is no force velocity coupling on the slave side. When there is time delay on the channels however the merit of the damping injection is obvious. Figure 6.6 shows the experiment results with 0.2s time delay between each robot making the round trip(RT) time delay 0.4s. It can be seen that for the time delay case damping does not decrease the transparency of a four channel controller, instead it has better transparency than, four channel without damping. This is actually because of the fact that system is also more stable. H_{11} parameter shows that four channel teleoperation also has high perceived damping under time delay and it is not constant. On the other hand, increasing the damping makes this response more constant, and better for teleoperation although the perceived damping is increased as well. However it can be seen from H_{12}, which maps human force to environment force, that damping injection increases force reflecting capability. As damping term is increased the response goes to the ideal 0(as it is a bode plot)

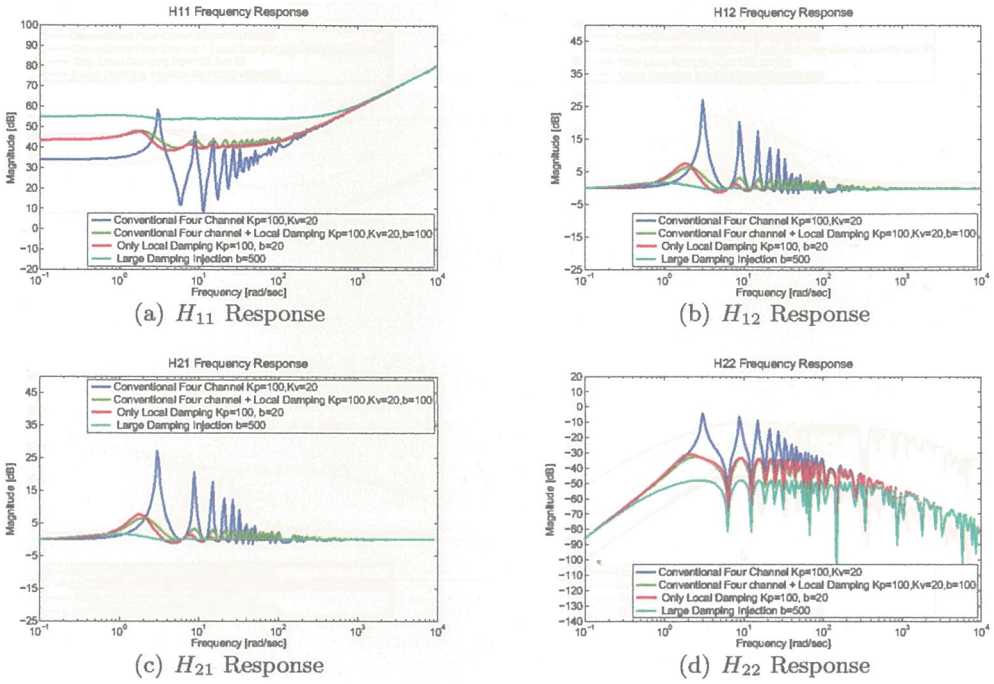


(a) H_{11} Response (b) H_{12} Response

(c) H_{21} Response (d) H_{22} Response

Figure 6.6: Hybrid Matrix Responses for 0.4s RT Delay

for all frequencies. This also means that force reflection in high frequencies can also be realized. The behavior is similar for the H_{21} response which maps the slave velocity to master velocity. Increased damping provides better tracking. Plots for H_{12} and H_{21} also imply that increasing the damping with respect to position gain considerably increases the velocity tracking performance of the system as well as the force reflection performance. Also it can be seen that not using the K_v parameter is better for tracking. H_{22} shows that damping injection also minimizes the influence of slave force on slave velocity, as desired. Figure 6.7 shows similar results for an increased time delay of 0.5 second delay on each channel with RT delay of 1 second. Again, increase in damping makes tracking better, although the tracking is a little worse compared to the 0.4s RT delay case at the low frequencies. Therefore the main tradeoff in choosing the damping gain depends on whether we want a lighter

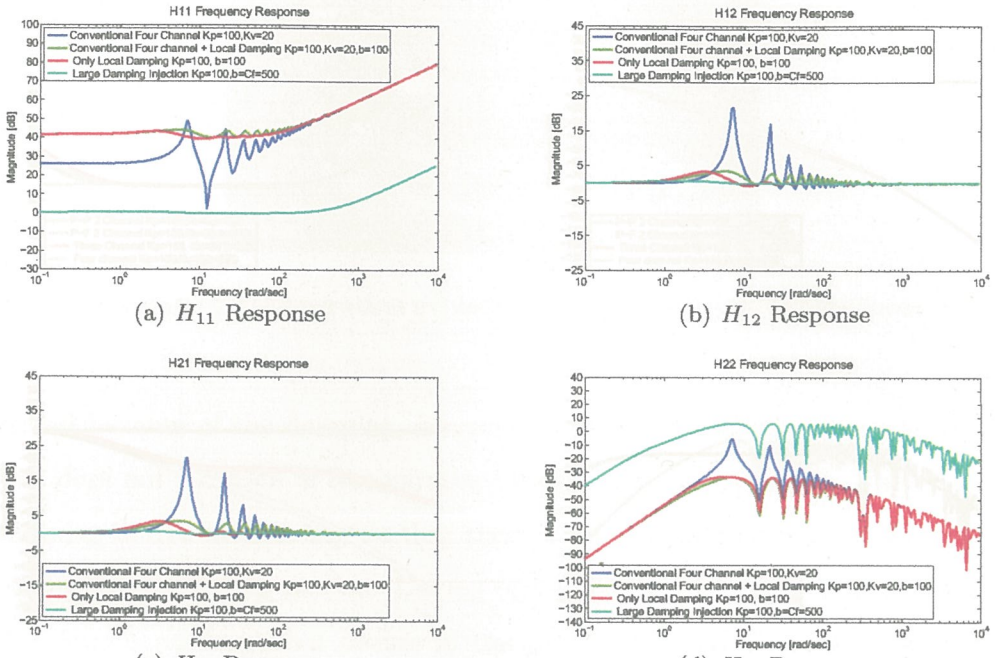


(a) H_{11} Response (b) H_{12} Response

(c) H_{21} Response (d) H_{22} Response

Figure 6.7: Hybrid Matrix Responses for 1s RT Delay

feeling system with low performance position/force tracking or we want a heavy system with better position and force tracking system. However there is one way, to fight against the damping felt by the user, and that is to use the force gains of the four channel controller C_f. Although, for practical uses it cannot be increased ad infinitum, choosing it as $C_f = b$ would yield a force to velocity ratio of 1 as can be seen in figure 6.8. This means that the user would not feel much damping, and what is more important is that it does not change stability results and the transparency responses of the tracking parameters H_{12} and H_{21}. However it does increase the H_{22} parameter, so it should not be increased arbitrarily. However, application wise, it should be possible to find values which would satisfy both user needs and stability needs. Figure 6.9, shows the difference between the damping injection methods proposed in [28][29] and the four channel version. The three channel version used



(a) H_{11} Response (b) H_{12} Response

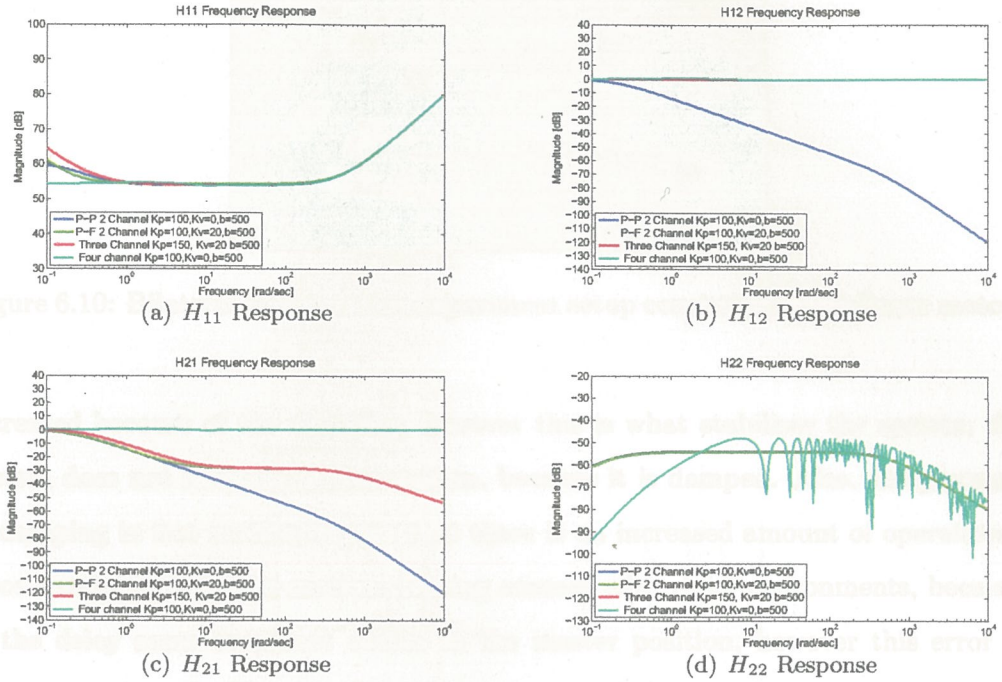
(c) H_{21} Response (d) H_{22} Response

Figure 6.8: Hybrid Matrix Responses for 0.4s RT Delay with $C_f = b$

in [31] is also included with damping injection, and it is clear that the damping injection with four channel architecture is the best performing method.

6.4 Experiment Results

Experiments were conducted with two identical linear 1DOF robots connected through a computer which emulated the time delay on the network. Figure 6.10 shows the experiment setup. The motors that are used are linear motors with very low friction, and high resolution encoders are used for position measurement. Also, force sensing is realized by the RFOB based on the encoder data. In the experiments a human operator pushes the master robot starting in free space, until the slave robot contacts a rigid metal object. Some amount of force is applied by the human



(a) H_{11} Response (b) H_{12} Response

(c) H_{21} Response (d) H_{22} Response

Figure 6.9: Hybrid Matrix Responses for various architectures with damping injection under 0.4s RT delay

also after the contact, to test stability and force reflection, and then the robot is retracted. This motion is repeated numerous times. In the case of no time delay, the damping injection based system behaves very similar to a conventional four channel system, except that the perceived inertia of the robots is slightly higher due to damping. Delays of various sizes and combination (with different T_1 and T_2 values) were also applied and it was confirmed that the damping injection method provided stability independent of delay. Traditional four channel controllers with $K_p = 2500$ $K_v = 100$ were unstable under delays as small as 100 ms; they started resonating in free motion, a behavior which can be understood from the stability analysis provided before. Figure 6.11 shows the unstable behavior of the teleoperation system. When the damping injection was applied, the perceived inertia of the master robot

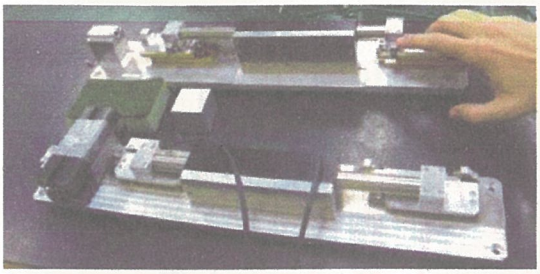


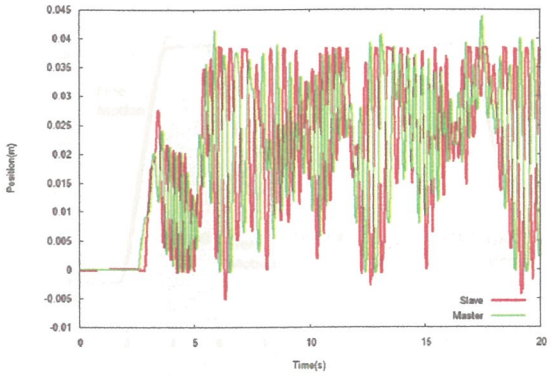
Figure 6.10: Bilateral teleoperation experiment setup consisting of two linear motors

increased because of the damping, however this is what stabilizes the system; the system does not resonate in free motion, because it is damped. Also, the presence of damping in free motion implies that there is an increased amount of operational force exerted during free motion. During contact with rigid environments, because of the delay some overshoot occurs in the master position, however this error is soon compensated. This in effect gives a spongy feeling to the operator, however the rigidity of the environment can be felt with delay. During damping injection

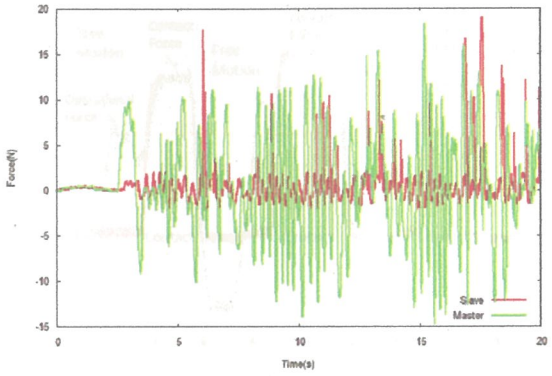
Table 6.2: Damping injection experiment parameters

Proportional gain	K_p	2500
Differential gain	K_v	0
Force control coefficient	C_f	1
Master damping injection	b_m	500
Slave damping injection	b_s	500
Master and Slave mass	M_n	0.5kg
Control sampling time	st	1ms
Cut-off frequency of DOB	g_{dis}	500rad/s
Cut-off frequency of RFOB	g_{reac}	500rad/s

experiments control parameters were selected as $K_p = 2500$ and $b_m = b_s = 500$. Note that these values were the same regardless of the size of the delay. High proportional gain was selected because it enables much better position tracking. On the other hand, damping injection must also be chosen considerably big as demon-



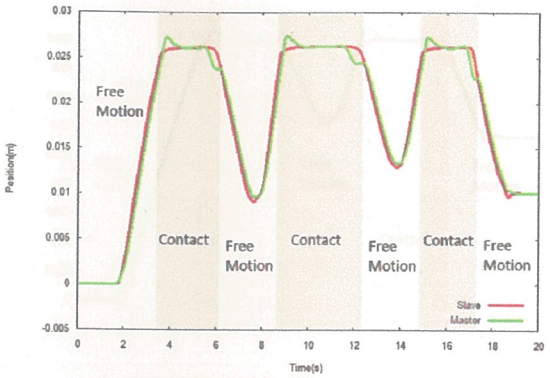
(a) Position Response



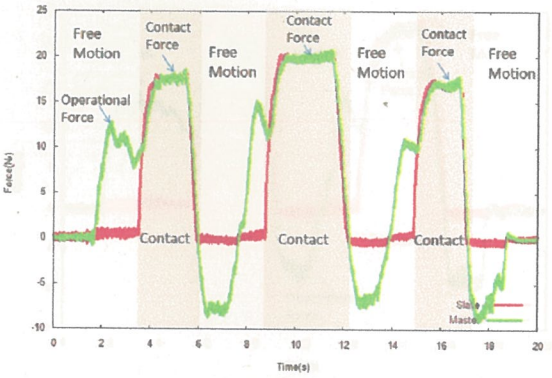
(b) Force Response

Figure 6.11: Unstable four channel teleoperation with roundtrip time delay 400 ms

strated by the transparency analysis. However it is not always practical to choose the damping injection term bigger than the proportional gain, as this implies a very large operational force, and a very heavy teleoperation system. Therefore the damping terms were selected as 500, which create a relatively lighter teleoperation system, but obviously this value is higher than the critical condition for stability. Figure 6.12 shows the experiment results for the case where there is a time delay of 200ms from slave to the master. Therefore the slave motion seems to be leading the master motion. However position tracking is realized. As can be seen, due to time delay there is some overshoot in the master manipulator when contact is real-



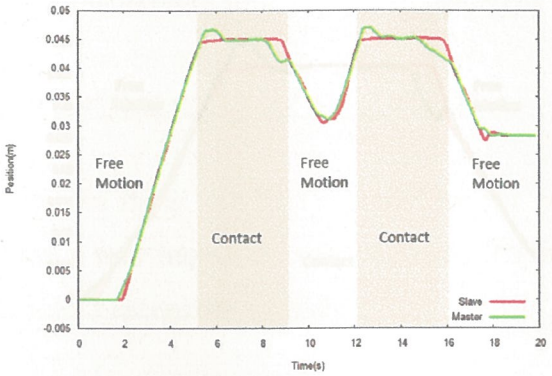
(a) Position Response



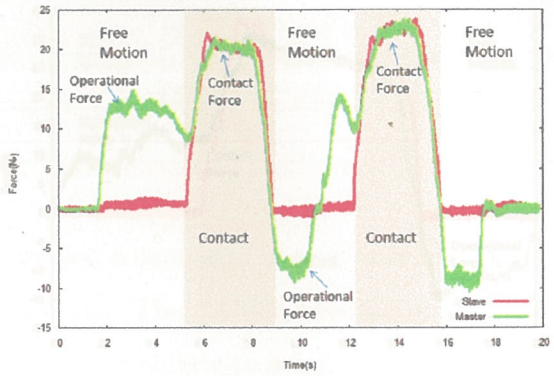
(b) Force Response

Figure 6.12: Experiment results with one way delay of 200ms from slave to master

ized. Also force reflection is realized succesfully during contact. Figure 6.13 shows the experiment results with 400ms roundtrip time delay which is considerably big. The position tracking is still good this time with no slave lead, except the slight overshoot in rigid contact. However there is no position drifting as in previously proposed methods based on 2 channel controllers. High frequency force reflection is also realized, as seen from the high frequency components of the forces. Figure 6.14 shows the experiment results with 1 second time delay. This delay is much bigger than can be seen in most telerobotic applications using the internet. However, the system was stable regardless of the delay. But as the transparency is dependent on



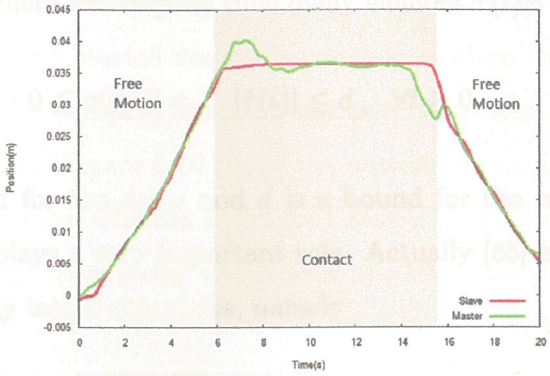
(a) Position Response



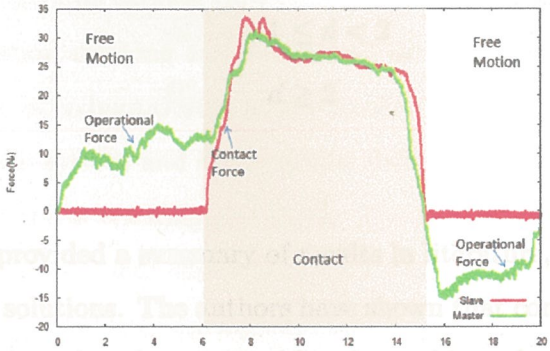
(b) Force Response

Figure 6.13: Experiment results with 0.4 second roundtrip time delay

delay, position tracking deteriorated compared to the 0.4 second case. The system in general becomes slower than previous cases. For instance it takes a longer time for the master manipulator to track the settling position of the slave after the contact and the force reflection is also affected by this phenomenon. Still the result is quite good considering the size of the delay. The experiments indeed confirm the validity of the damping injection method and the analysis.



(a) Position Response



(b) Force Response

Figure 6.14: Experiment results with 1 second roundtrip time delay

6.5 Bilateral Teleoperation under Varying Time Delay

Varying time delay is a more complicated problem than constant time delay and in network environments with packet based communication, the size of the delay on the network could also be probably varying. Analysis of varying delays in control system is considerably more difficult, however in [65] it was shown that the difficulty of the problem can be classified according to the speed with which the delay is changing.

According to [65], where the varying time delay element $\tau(t)$is defined as:

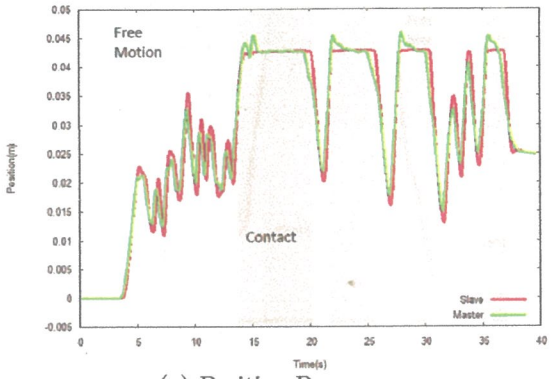
$$0 \leq \tau(t) \leq \tau_o\,, \quad |\dot{\tau}(t)| \leq d\,, \quad \forall t \geq 0 \tag{6.62}$$

where τ_o is a bound for the delay and d is a bound for the rate of change of the delay, the size of d plays a very important role. Actually [65] has divided the cases of varying time delay into 3 categories, namely

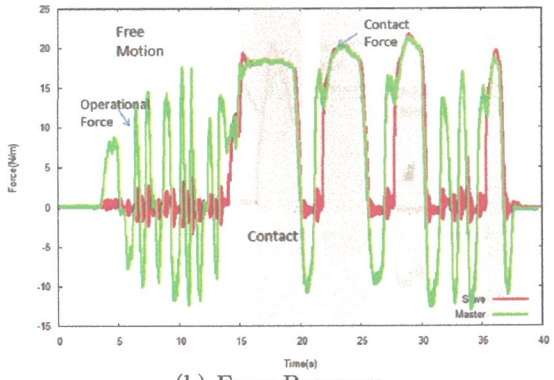
$$\begin{aligned} &(1) \qquad d \leq 1 \\ &(2) \qquad 1 \leq d < 2 \\ &(3) \qquad d \geq 2 \end{aligned} \tag{6.63}$$

The paper has also provided a summary of results in literature, showing that case 3 does not have many solutions. The authors have shown that conventional small gain based methods would not be adequate to solve the problems for cases two and three, and have proposed an IQC framework to solve such problems. Therefore the delay independent stability analysis previously provided in this dissertation for constant time delays would not be valid for varying time delay cases. However, the validity of damping injection can be tested under conditions satisfying extreme cases such as 3, with experiments. Experiments were conducted with two linear robots connected through a computer which emulated the time delay on the network. In the case of no time delay, the damping injection based system behaves very similar to a conventional four channel system, except that the perceived inertia of the robots is higher due to damping. Experiments with variable time delay were realized. This is to show the efficiency of the proposed method in almost real life conditions, such as seen in teleoperation through packet based systems like the internet. Figure 6.15 shows the experiment results for conventional four channel teleoperation under

round trip variable time delay of 200+200sin(2000t)[ms]. The four channel teleoperator was unstable and started resonating in free motion, in contact it can also start resonating. When damping injection was used it was seen that stable teleoperation could be realized. Figure 6.16 shows experiment result with round trip delay of 200+ 200sin(5000t)[ms] whereas figure 6.17 shows 400+200sin(t)[ms]. Note that the rate of change for these delay elements d are 1000 and 0.2 respectively. The delay shown in 6.17 has $d = 1000$ which is much bigger than the lower bound 2 of case 3. Therefore it shows that the damping injection method can be succesfully implemented under extreme varying time delay conditions as well. The experiments show that good position tracking can be realized in both free motion and contact. During contact some overshoot occurs at the master side due to the delay, but this doesn't destabilize the system and the operator feels a sponge like effect as the position of the master starts tracking the slave. The contact forces can also be felt with delay. The first experiment shows that with a faster changing time delay some chatter in the motion, especially during environmental contact can be seen.

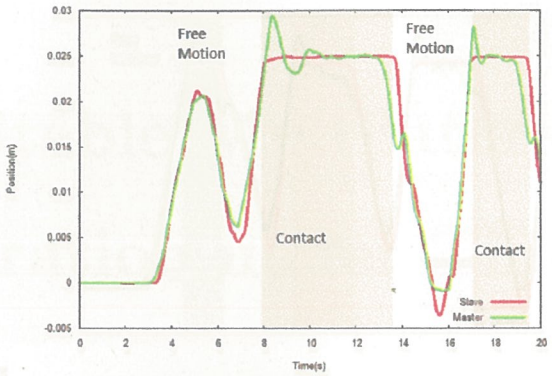


(a) Position Response

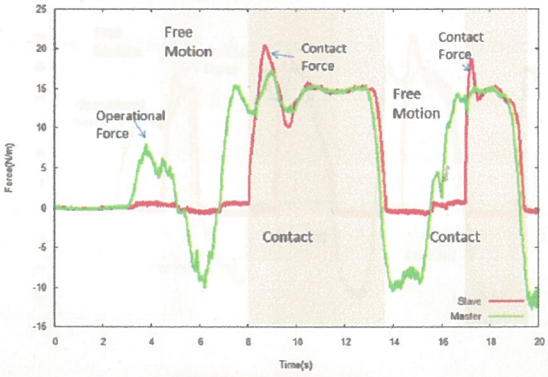


(b) Force Response

Figure 6.15: Experiment results with one-way delay 100 + 100sin5000t[ms] variance

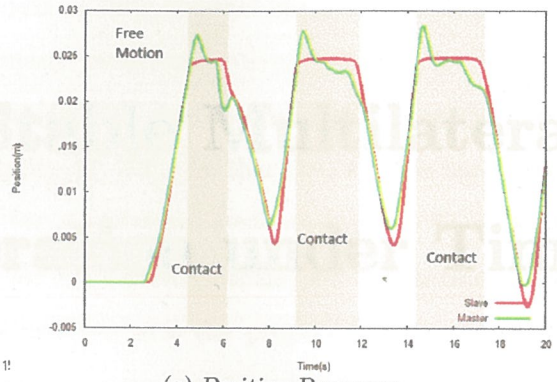


(a) Position Response

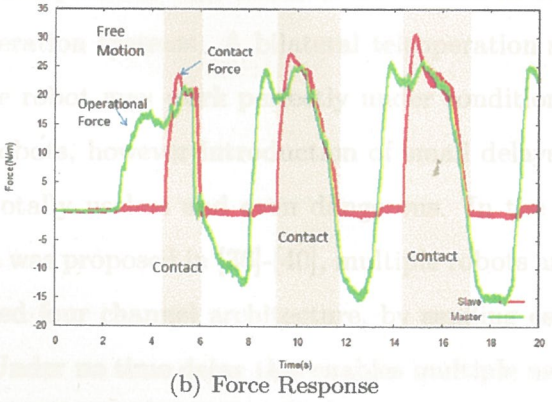


(b) Force Response

Figure 6.16: Experiment results with one-way delay 200 + 100sint[ms] variance



(a) Position Response



(b) Force Response

Figure 6.17: Experiment results with one-way delay 200 + 100sin2000t[ms] variance

Chapter 7

Stable Multilateral Teleoperation under Time Delay

Time delay is a crucial problem that needs to be tackled in order to realize practically useful teleoperation systems. A bilateral teleoperation system consisting of a master and a slave robot may work perfectly under conditions where there is no delay between the robots, however introduction of small delays of even 100ms can render the system totally useless and even dangerous. In the case of multilateral teleoperation, which was proposed in [36]-[40], multiple robots are connected to each other via an extended four channel architecture, by sending each other their force and position data. Under no time delay this enables multiple users to feel the forces exerted on the slave robot as well as the forces exerted by other operators. These kinds of systems can be useful in collaborative teleoperation, where multiple masters can share the control over the slave and can be employed for tasks such as surgical instruction, etc... However because the four channel architecture causes instability under delay, it becomes even a bigger problem when there are multiple channels among multiple robots with delay. Therefore until now, not much research has been done in realizing a stable multilateral teleoperation system under time delay. In

the previous chapter a method to realize stable four channel bilateral teleoperation under time delay was proposed. This involves augmenting robot controllers by local damping injection based on robust acceleration consensus algorithms. In this chapter this damping injection based four channel architecture is extended to the multilateral case by making use of the scalability of consensus algorithms. Furthermore, stability analysis of four channel systems has been a difficult problem until now because of the many variables involved in the controllers. Input output analysis for connected systems[48]-[50] can be applied to bilateral teleoperation systems in the form of L_2 stability[49],[52]. In this chapter the input output stability analysis is also extended for multilateral systems and the L_2 stability is shown by using small gain theorem for connected systems[50]. Furthermore it is shown that experiments support this analysis and the proposed method can indeed provide stable multilateral teleoperation under considerably big delays.

7.1 Multilateral teleoperation with damping injection

In multilateral haptic teleoperation, the forces felt at each robot must be transmitted to the other, and the robots must have the same motion under all circumstances. This can be especially useful when two or more operators are controlling a slave robot. All the robots should do the same motion controlled by the operators. And if the slave contacts any object, the impact force should be transmitted to all the operators. This means that at steady state; the sum of the forces felt at every robot must equal zero, and the positions of each robot should be equal:

$$F_1 + F_2 + \ldots + F_N = 0 \tag{7.1}$$

$$X_1 = X_2 = \ldots = X_N \tag{7.2}$$

For these goals N decentralized acceleration controllers can be designed as:

$$\begin{aligned} \ddot{x}_1^{ref} &= \ddot{x}_1^{cmd} + C_p(x_1^{cmd} - \delta_{11}x_1) \\ \ddot{x}_2^{ref} &= \ddot{x}_2^{cmd} + C_p(x_2^{cmd} - \delta_{22}x_2) \\ &\vdots \\ \ddot{x}_N^{ref} &= \ddot{x}_N^{cmd} + C_p(x_N^{cmd} - \delta_{nn}x_N) \end{aligned}$$

where $C_p = K_p + K_v s$, δ_{ii} are the diagonal entries of the outdegree matrix Δ and the command values for the position are defined by the connection matrix $\mathcal{A}$:

$$x^{cmd} = \mathcal{A}x \tag{7.3}$$

and all the acceleration feedforwards are the same:

$$\ddot{x}_1^{cmd} = \ddot{x}_2^{cmd} = \ldots = \ddot{x}_N^{cmd} = C_f(F_1 + F_2 + \ldots + F_N) \tag{7.4}$$

In this chapter a full connection among the robots which can be represented by the

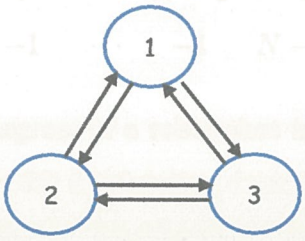


Figure 7.1: Strongly connected graph structure

graph in figure 7.1 is assumed, which means that all the robots send their information to one another. The adjacency and degree matrices of the strongly connected graph

can be generalized as:

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & & \vdots \\ \vdots & & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \Delta = \begin{bmatrix} N-1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & \cdots & N-1 & 0 \\ 0 & \cdots & 0 & N-1 \end{bmatrix}$$

This means a simplified acceleration consensus algorithm of the form is obtained:

$$\begin{aligned} \ddot{x}_i^{ref} &= \sum_{j=1, j\neq i}^{N} C_p(x_j - x_i) - C_f \sum_{i=1}^{N} F_i \\ \ddot{x}^{ref} &= -C_p(s)Lx - C_f \sum_{i=1}^{N} F_i \end{aligned}$$

where $-C_f \sum_{i=1}^{N} F_i = \sum_{i=1}^{N} u_i$ and the Laplacian can be written as:

$$L = \begin{bmatrix} N-1 & -1 & \cdots & -1 \\ -1 & \ddots & -1 & -1 \\ \vdots & \cdots & N-1 & -1 \\ -1 & \cdots & -1 & N-1 \end{bmatrix}$$

Figure 7.2 shows the block diagram of a robot that is networked with other robots and controlled multilaterally, with acceleration control using disturbance observers. Where the channel and controller gains for each robot i are according to equation:

$$C1_i = C_p(s) = K_p + K_v s \tag{7.5}$$

$$C2_i = C_f = 1 \tag{7.6}$$

$$C_i = K_p + K_v s \tag{7.7}$$

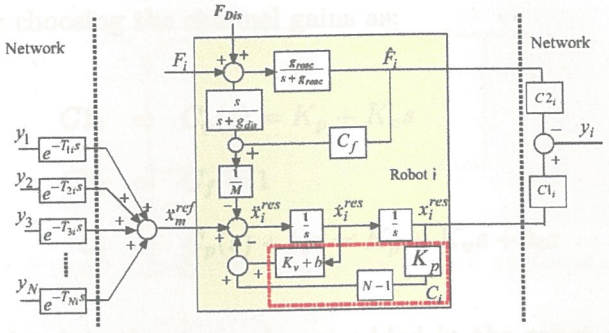


Figure 7.2: Single robot block diagram and communication for multilateral teleoperation

where $C1_i$ is the position channel, $C2_i$ is the force channel and Cf is a force control coefficient which will be assumed as 1 in this chapter, and C_i are the local controllers of the robots. When channel $C1$ and $C2$ have delay, the system controllers will be much more complicated and will have the delay in all the elements:

$$\ddot{x}_i^{ref} = \sum_{j=1, j\neq i}^{N} C_p(x_j(e^{-T_{ij}s}) - x_i)$$
$$-C_f(\sum_{j=1, j\neq i}^{N} e^{-T_{ij}s}F_j + F_i)$$

It was shown in chapter 3 for acceleration consensus algorithms local velocity feedback can stabilize a system under time delay. As the multilateral control law proposed in this chapter is a special form of acceleration consensus algorithm, using local damping injection could stabilize the system. Instead of or in addition to using velocity feedback on the communication channels, using an exclusive local damping loop in all the robot controllers will be a simple but powerful approach to stabilize the overall system. Because the method involves only local information it is very easy to implement on multiple robot systems and will have a symmetric architecture. It is possible to employ the local damping injection for this kind of multilateral

control scheme by choosing the channel gains as:

$$C1_i = C_p(s) = K_p + K_v s \tag{7.8}$$

$$C2_i = C_f = 1 \tag{7.9}$$

$$C_i = C_p(s) + b_i s = K_p + K_v s + b_i s \tag{7.10}$$

where local damping injection terms b_i are added in the robot controllers so that the acceleration controller laws become the same as equation (3.65):

$$\ddot{x}_i^{ref} = \sum_{j=1, j\neq i}^{N} C_p(x_j - x_i) - b_i x_i - C_f \sum_{j=1}^{N} F_j$$

and in the existence of time delay they would would become:

$$\ddot{x}_i^{ref} = \sum_{j=1, j\neq i}^{N} C_p(x_j(e^{-T_{ij}s}) - x_i) - b_i x_i$$
$$-C_f(\sum_{j=1, j\neq i}^{N} e^{-T_{ij}s} F_j + F_i)$$

As can be seen there is a local feedback term that is unaffected from time delay and this will make the system stable. In the next section L_2 stability concept will be introduced and both controllers will be compared.

7.2 Small gain theorem and L_2 stability of multilateral teleoperation

A closed loop connected system of the form that can be seen in figure 7.3 can be described by equations:

$$u_i = e_i - \sum_{j=1}^{N} H_{ij} y_j$$

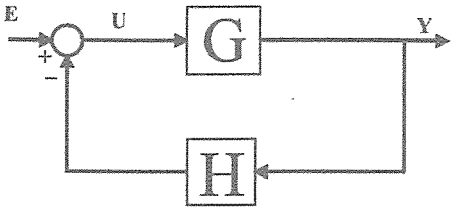


Figure 7.3: Input output connected systems

$$y_i = G_i(u_i)$$

where $G = diag(G_i)$ and H are matrices. According to [50] the overall connected system will be L_2 stable if both G and H have finite L_2 gains $\gamma_2(G)$ and $\gamma_2(H)$ respectively and are stable, and also if the multiplication of their gains satisfy

$$\gamma_2(G) \cdot \gamma_2(H) < 1 \tag{7.11}$$

the closed loop system has a small loop gain and is also L_2 stable.

7.2.1 L_2 stability of multilateral teleoperation

In order to show the L_2 stability of multilateral teleoperation system, the system should be transformed into single loop form as presented previously. Because the forces generated during contact are functions of the system states, it is not possible to discard the impedances of the environment and the human in this analysis. Therefore the forces exerted on the system from the environment and the operator can be written as:

$$Operator/Environment\ i \quad : \quad F_i = F_i^* - Z_i V_i \tag{7.12}$$

where F_i^* are the exogenous input forces that do not depend on system states. And the impedances are mapped from the velocity to force such that:

$$Z_i = \frac{k_{zi}}{s} + b_{zi} + m_{zi}s \tag{7.13}$$

For the L_2 stability, it is assumed as in previous works(27), that these exogenous forces are in L_2 space and are bounded. Hence, by using these velocity force mappings in frequency domain it is possible to obtain the multilateral teleoperation architecture in input output form as in figure 7.4. Here the systems G and H are

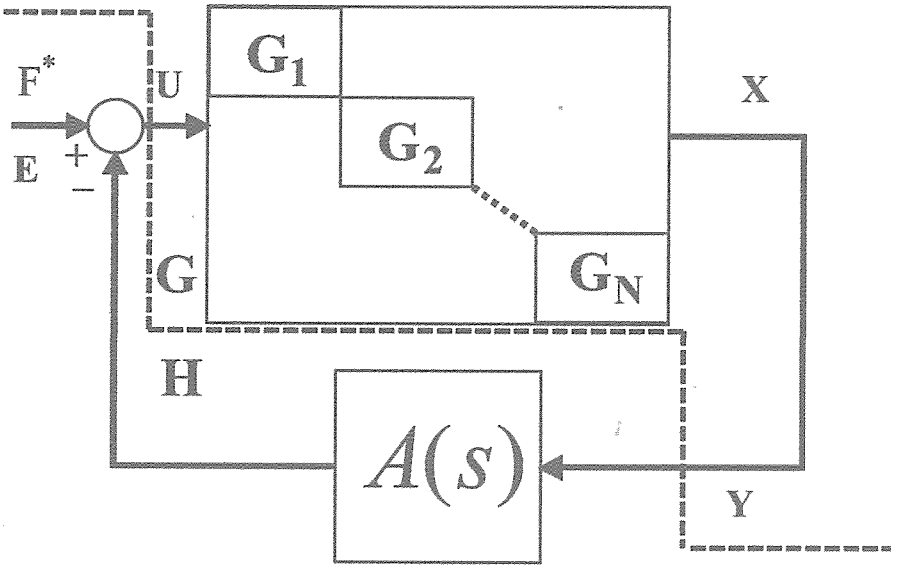


Figure 7.4: Input output transformation of the multilateral teleoperation architecture

input output connected and represent the local dynamics of the robots and the interconnection between them respectively. In the proposed multilateral teleoperation scheme the interconnection is represented by the adjacency matrix for a strongly connected graph. Note that the time delay element is also a part of these systems.

The overall equations for the system can be written as:

$$\begin{aligned} e_i &= \sum_{j=1, j\neq i}^{N} e^{-T_{ij}s} F_j^* + F_i^* \\ u_i &= e_i - \sum_{j=1}^{N} H_{ij} y_j \\ y_i &= G_i(u_i) \\ G_i &= \frac{(C1_i(s) - C2_i Z_i(s)s)}{s^2 + (b_i + Z_i(s))s + \delta_{ii} C_p(s)} \\ H &= \mathcal{A}(s) = \begin{cases} a_{ij} e^{-T_{ij}s} & if \ (v_i, v_j) \in \mathcal{E} \\ 0 & otherwise \end{cases} \end{aligned}$$

where the graph link weights are taken to be one: $a_{ij} = 1$. The L_2 gains of the matrix system can be obtained by the supremum of the greatest singular values of the matrices in frequency domain:

$$\gamma_2(S) = \sup_{w>0} \sigma_{max}(S(jw)) \tag{7.14}$$

$$\sigma_{max}(S(jw)) = \sqrt{\lambda_{max}\{S'(-jw)S(jw)\}} \tag{7.15}$$

This means that if $\gamma_2(G(w)) \cdot \gamma_2(H(w)) < 1$ for all $w > 0$, the multilateral connection will be stable. In order to obtain the sufficient conditions to guarantee stability, it should be guaranteed locally that each subsystem will be stable. According to Gersgorin's disk theorem the eigenvalues of the connection matrix $\mathcal{A}(s)$, will have eigenvalues λ located in or on the union of the following discs:

$$D_i(s) := \{\lambda \in C : \ |\lambda| \leq \sum_{j \in N_i} \left| a_{ij} e^{-T_{ij}s} \right| \} \tag{7.16}$$

By definition the sum of off diagonal elements of each row equals to the degree of the row: $\sum_{j \in N_i} a_{ij} = \delta_{ii}$ and the delay element will equal to 1: $|e^{-T_{ij}s}| = 1$. Hence

it can be shown that

$$\sum_{j \in N_i} |a_{ij} e^{-T_{ij} s}| = \sum_{j \in N_i} a_{ij} = \delta_{ii} \tag{7.17}$$

Which means that the eigenvalues will be on or inside the circle with radius δ_{ii}. This is the spectral radius of A(s) and will be the greatest eigenvalue of the matrix which is also the degree number of the matrix. In the case of a strongly connected graph of the form in figure 7.1 it will be equal to $N-1$:

$$\rho(A(s)) = \lambda_{max}(A(s)) = \delta_{ii} = N - 1 \tag{7.18}$$

where $\rho(A(s)$ is the spectral radius of $A(s)$. It can be shown for symmetrical matrices that the spectral radius of the matrix is also equal to the 2 norm of the matrix which is in turn equal to the greates singular value and the L_2 gain. Hence when the forward and backward delay elements between any two robots is equal, i.e. $T_{ij} = T_{ji}$ the matrix $A(s)$ will be symmetric and have the L_2 gain:

$$\gamma_2(H(jw)) = \rho\{A(jw)\} = \delta_{ii} = N - 1 \tag{7.19}$$

For the more complicated case where the delay elements are not symmetric, it is necessary to obtain the spectral radius of the matrix $A'(-jw)A(jw)$ as:

$$\gamma_2(A(jw)) = \sqrt{\lambda_{max}\{A'(-jw)A(jw)\}} = \sqrt{\rho(A'(-jw)A(jw))} \tag{7.20}$$

This can be realized by applying the Gersgorin's disk theorem on $A'(-jw)A(jw)$. When the graph is strongly connected as in figure 7.1, the diagonal elements of $A'(-jw)A(jw)$ will be equal to d_{ii} and the off diagonal elements will all be nonzero with values of some complex delay element. Gersgorin's disk theorem states that

the eigenvalues z will be in or on the disks:

$$D_i(s) := \{\lambda \in C : \ |\lambda - a_{ii}| \leq \sum_{j \in N_i} |(A_i'(-jw)A_j(jw)|\} \tag{7.21}$$

When these are computed again by using the fact that $|e^{-T_* s}| = 1$ for every element, the diagonals are zero and all the off diagonal elements are nonzero we obtain $\sum_{j \in N_i} |(A_{i*}'(-jw)A_{*j}(jw))| = \delta_{ii} \, |(\delta ii - 1)|$ and $a_{ii} = d_{ii}$, and the spectral radius of the matrix will be:

$$\rho(A'(-jw)A(jw)) = \delta_{ii} + (\delta_{ii}(\delta ii - 1)) = \delta_{ii}^2$$

Hence the gain of the matrix can be obtained as

$$\gamma_2(A(jw)) = \sqrt{\lambda_{max}\{A'(-jw)A(jw)\}} = \sqrt{\rho(A'(-jw)A(jw))} = \delta_{ii} = N - 1$$

Thus it is shown for this more general case as well that, that the gain will be $N-1$. By using these results it possible to rewrite the small gain theorem to obtain the constraints for the design of subsytems G as:

$$\gamma_2(G(jw)) \cdot (N-1) < 1 \, , \quad \forall w > 0 \tag{7.22}$$

$$\gamma_2(G(jw)) < \frac{1}{N-1} \, , \quad \forall w > 0 \tag{7.23}$$

Furthermore, because G is a diagonal matrix, the maximum singular value of the matrix will belong to the subsystem $G_i(s)$ with the maximum complex value. This can be written as follows:

$$\gamma_2(G(jw)) = \sup_{w>0} \{\sqrt{\lambda_{max} G(-jw)G(jw)}\} \tag{7.24}$$

$$\gamma_2(G(jw)) = \sup_{w>0}\{\max_i\{\sqrt{G_i(-jw)G_i(jw)}\}\} \tag{7.25}$$

Hence the stability of the overall system can be guaranteed by locally guaranteeing:

$$\sqrt{G_i(-jw)G_i(jw)} < \frac{1}{N-1}\,, \quad \forall w\,, i \tag{7.26}$$

where G_i was defined as follows:

$$G_i = \frac{(C1_i(jw) - C2_i Z_i(jw))e^{-T_i(jw)}}{(jw)^2 + (b_i + Z_i(jw))jw + \delta_{ii}C_p(jw)} \tag{7.27}$$

Which leads to:

$$\sqrt{\frac{(C1_i(jw)-C2_iZ_i(jw))}{(jw)^2+(b_i+Z_i(jw))jw+\delta_{ii}C_p(jw)} \cdot \frac{(C1_i(-jw)-C2_iZ_i(-jw))}{(-jw)^2-(b_i+Z_i(-jw))jw+\delta_{ii}C_p(-jw)}} < \frac{1}{N-1}\,, \quad \forall w\,, i \tag{7.28}$$

It will next be shown that damping injection method can satisfy the above condition given that there is adequate damping, whereas the conventional four channel architecture cannot. The conventional four channel arcitecture will have the gains again dependent only on impedance, controller parameters and the frequency, and will be independent of time delay: If the traditional 4 channel controllers are selected as in equations (7.5) -(7.7) these norms can be written as a function of the frequency w, the controller parameters and the impedance parameters :

$$|G_i| = \sqrt{\frac{(K_p-k_{zi})^2+2(K_p-k_{zi})(m_{zi})w^2+(K_v-b_{zi})^2w^2+(m_{zi}w^2)^2}{(\delta_{ii}K_p+k_{zi})^2-2(\delta_{ii}K_p+k_{zi})(m_{zi}+1)w^2+(b_{zi}+\delta_{ii}K_v)^2w^2+(m_{zi}+1)^2w^4}} \tag{7.29}$$

Even for the simplest case of free motion, where the effects of the environmental and operator impedances are neglected, the local gains are:

$$|G_i| = \sqrt{\frac{(K_p)^2 + (K_v)^2w^2}{(\delta_{ii}K_p)^2 - 2(\delta_{ii}K_p)w^2 + (\delta_{ii}K_v)^2w^2 + w^4}}$$

It can easily be seen that for $w^2 < 2(\delta_{ii}K_p)$ the transfer function will have gain greater than $\frac{1}{\delta_{ii}} = \frac{1}{N-1}$, rendering the system unstable. On the other hand with the introduction of local damping injection terms, b_i will show up in the denominator, helping to solve this problem:

$$|G_i| = \sqrt{\frac{(K_p)^2 + (K_v)^2 w^2}{(\delta_{ii}K_p)^2 - 2(\delta_{ii}K_p)w^2 + (\delta_{ii}K_v + b_i)^2 w^2 + w^4}}$$

if they are chosen to satisfy:

$$b_i \geq \sqrt{2\delta_{ii}K_p} \tag{7.30}$$

which will yield

$$|G_i| = \sqrt{\frac{(K_p)^2 + (K_v)^2 w^2}{(\delta_{ii}K_p)^2 + (\delta_{ii}K_v)^2 w^2 + 2(\delta_{ii}K_v b_i)^2 w^2 + w^4}} < \frac{1}{\delta_{ii}} = \frac{1}{N-1}, \quad \forall w > 0$$

which will guarantee L_2 stability of the multilateral system. Note that for the free motion case where the impedance terms are neglected, the stability conditions actually are equivalent for the stability of the acceleration consensus algorithm, which was shown following equation (3.52) in chapter 3. Slightly different results can be shown for the more generalized form where human and operator impedances are also included:

$$\sqrt{\frac{(K_p-k_{zi})^2+2(K_p-k_{zi})(m_{zi})w^2+(K_v-b_{zi})^2w^2+(m_{zi})^2w^4}{(\delta_{ii}K_p+k_{zi})^2-2(\delta_{ii}K_p+k_{zi})(m_{zi}+1)w^2+(b_i+b_{zi}+\delta_{ii}K_v)^2w^2+(m_{zi}+1)^2w^4}} < \frac{1}{\delta_{ii}} = \frac{1}{N-1} \tag{7.31}$$

this inequality can be rewritten as:

$$\frac{(K_p-k_{zi})^2+2(K_p-k_{zi})(m_{zi})w^2+(K_v-b_{zi})^2w^2+(m_{zi})^2w^4}{(\delta_{ii}K_p+k_{zi})^2-2(\delta_{ii}K_p+k_{zi})(m_{zi}+1)w^2+(b_i+b_{zi}+\delta_{ii}K_v)^2w^2+(m_{zi}+1)^2w^4} < \frac{1}{(\delta_{ii})^2} \tag{7.32}$$

that can be transformed into

$$(\delta_{ii}K_p - \delta_{ii}k_{zi})^2 + (2\delta_{ii}^2(K_p - k_{zi})m_{zi} + (\delta_{ii}K_v - \delta_{ii}b_{zi})^2)w^2 + (\delta_{ii}m_{zi})^2w^4 <$$
$$(\delta_{ii}K_p + k_{zi})^2 - 2(\delta_{ii}K_p + k_{zi})(m_{zi} + 1)w^2 + (b_i + b_{zi} + \delta_{ii}K_v)^2w^2 + (m_{zi} + 1)^2w^4$$

which would be satisfied if the following set of conditions for the impedances hold; for the stiffness values:

$$2K_pk_{zi}(\delta_{ii}^2 + \delta_{ii}) \quad \geq \quad k_{zi}^2(\delta_{ii}^2 - 1) \tag{7.33}$$

for the mass values:

$$\frac{1}{\delta_{ii} - 1} \geq m_{zi} \tag{7.34}$$

and the damping injection terms can be chosen as :

$$b_i \geq \sqrt{2(\delta_{ii}K_p + k_{zi})(m_{zi} + 1) + 2\delta_{ii}^2(K_p - k_{zi})m_{zi} + (\delta_{ii}^2 - 1)b_{zi}^2} \tag{7.35}$$

to guarantee that equation (7.31) will be satisfied, and the overall multilateral system will be L_2 stable. Note that, to prove the stability of the multilateral system all the relationships (7.33)-(7.35) should be satisfied. This is different from the bilateral case, where the environmental mass and stiffness values do not have to satisfy constraints such as given in (7.33) and (7.34). These analytical solutions provide us an insight in designing the controller parameters with respect to the environment and the number of robots. The position controller proportional gain terms K_p and the local damping injection terms b_i should be increased as the anticipated environmental stiffnesses increase. On the other hand the ratio of the environmental masses with respect to the robot masses plays an important role in this analysis. As the number of robots increase the environmental masses tolerated would be less. This

can be surpassed by increasing the masses of the robots (through disturbance observers). However, it should be noted that as mentioned in the previous chapter for bilateral teleoperation, the human and environmental masses are usually neglected in literature, as they are either static or very light. This could let us neglect the equation 7.34 for practical uses.

7.3 Experiment Results

For experiments, three linear motors were interfaced to a computer, emulating strongly connected network with delay. Figure 7.5 shows the experiment setup. During experiments the robots were pushed alternatingly and simultaneously at various times to realize multilateral teleoperation. Figure 7.6 shows the position and force responses of multilateral teleoperation with damping injection when there is no time delay between the robots. As can be seen close to perfect position tracking and force reflection can be realized, however some operational force during free motion can be felt due to damping injection. Figure 7.7 shows the position and force responses of the three robots when a small delay of 100 ms was introduced on each communication link. As can be seen after a few seconds of free motion, the robots start oscillating and the system becomes unstable. Figure 7.8 shows the position and force responses when bigger delays of 200ms between all the robots exist, with damping injection. It can be seen that the robots are stable during free motion as well as contact. During free motion, the operators feel the damping and this makes the system slower, however it also makes the system stable. After contact, there are some overshoots due to time delay, but position control is realized without position drift because of the four channel based architecture. Furthermore force reflection is realized among all three robots, meaning that the sums equal to zero. Figure 7.9 shows a similar result when delay is increased even more to 500ms between each robot. The performance is a little deteriorated due to delay, however stability is

preserved and force reflection as well as position tracking without drift is realized.

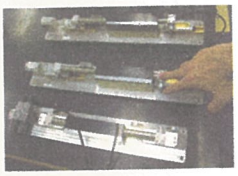
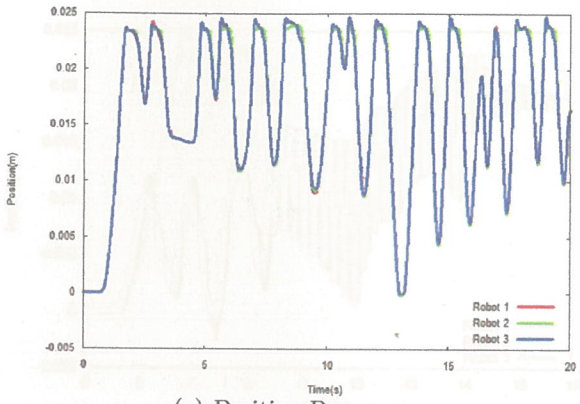


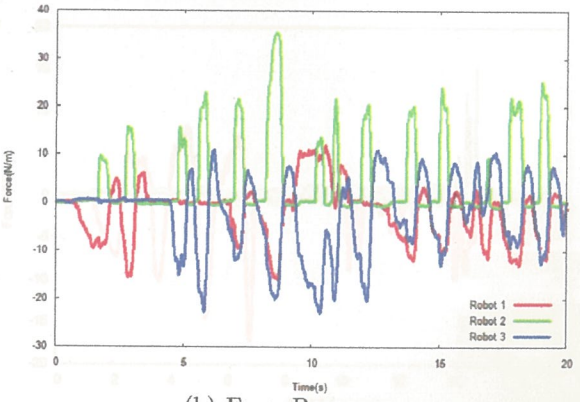
Figure 7.5: Multilateral experiment setup

7.4 Multilateral Teleoperation under Varying Time Delay

Similar to the bilateral case, the validity of damping injection based multilateral teleoperation under varying time delay was confirmed through experiments. Again a time varying delay changing with high frequency was chosen to show efficacy of the method. Figure 7.10 shows the case where there is a delay of 200+100sin(2000)t[ms] between all the robots, and for this case as well, multilateral teleoperation was succesfully realized as can be seen from the experiment results. It can be said that there is not a big difference in terms of performance between varying time delay and constant time delay cases. Only, in the varying delay cases with very high frequency some chattering can be felt.

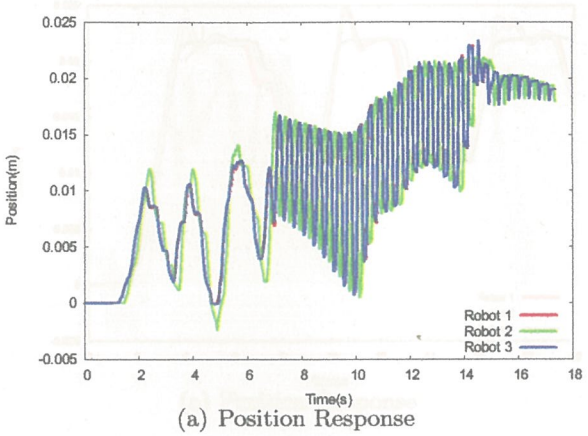


(a) Position Response

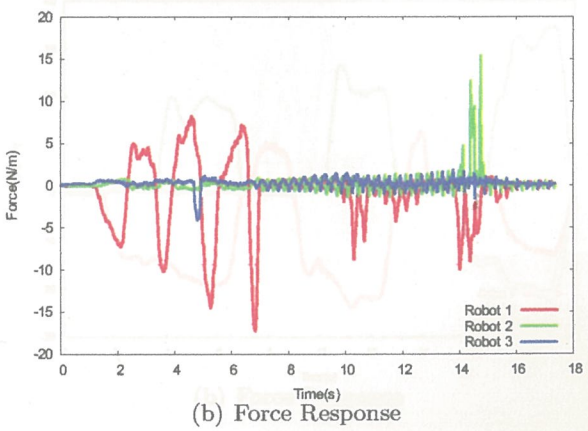


(b) Force Response

Figure 7.6: Performance of damping injection based Multilateral teleoperation with no time delay

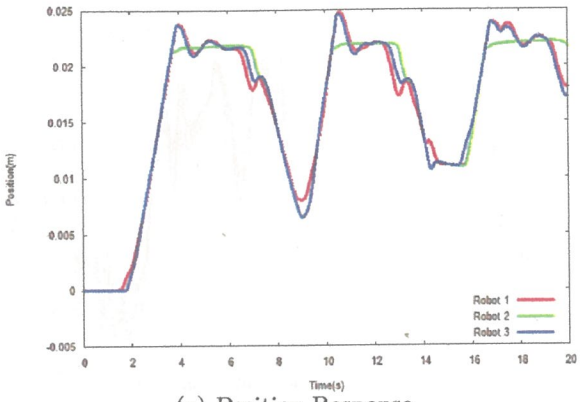


(a) Position Response

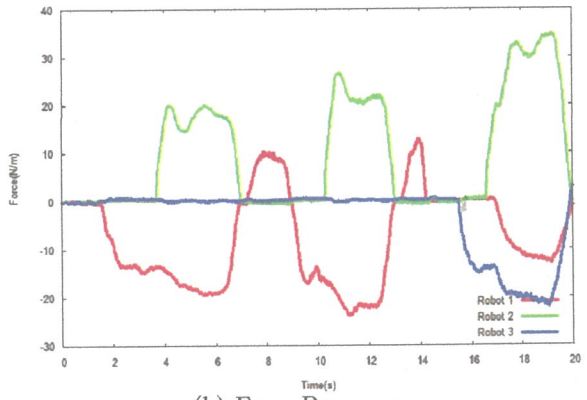


(b) Force Response

Figure 7.7: Unstable experiment results with 100ms time delay on a link

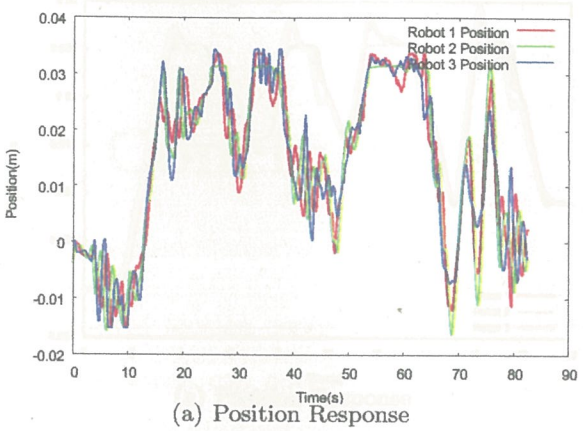


(a) Position Response

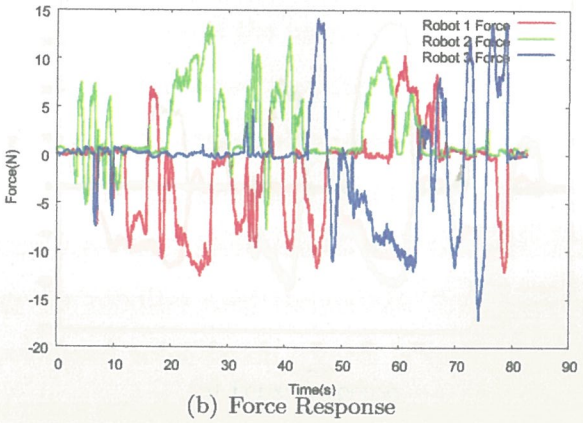


(b) Force Response

Figure 7.8: Performance of damping injection based Multilateral teleoperation with 200 ms delay between each robot

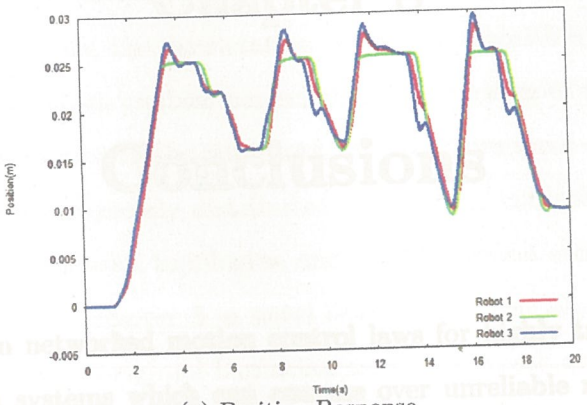


(a) Position Response

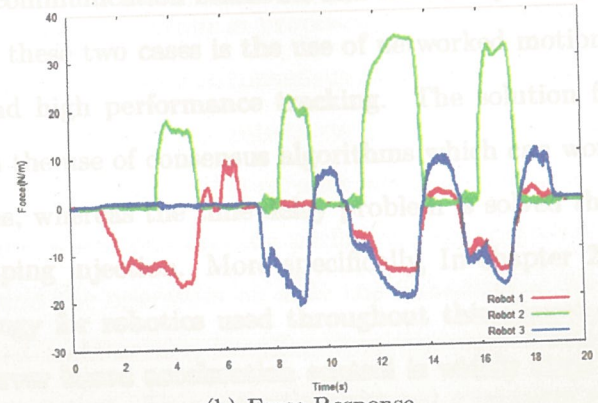


(b) Force Response

Figure 7.9: Stable experiment results with 0.5 second time delay between each robot



(a) Position Response



(b) Force Response

Figure 7.10: Stable experiment results with 200+100sin(2000(t))[ms] time delay between each robot

Chapter 8

Conclusions

In this dissertation networked motion control laws for highly transparent and robust teleoperation systems which can operate over unreliable networks that pose such problems as communication blackouts and time delays are proposed. The common approach for these two cases is the use of networked motion controllers which provide robust and high performance tracking. The solution for communication blackouts involves the use of consensus algorithms which can work under switching network topologies, whereas the time delay problem is solved through the application of local damping injection. More specifically, In chapter 2, the main motion control methodology for robotics used throughout this dissertation is introduced. Disturbance observer based acceleration control is widely used for robust position and force control of robot manipulators. In this dissertation this control methodology is generalized for networked motion control. In chapter 3, a graph theory based networked control algorithm is introduced, called the consensus algorithm. Firstly, some results regarding consensus based averaging algorithms are shown through simulations. By integrating the acceleration control framework introduced in chapter 2, with the consensus algorithms, the acceleration consensus algorithm is proposed. With acceleration control algorithm it is possible to control the relative speeds and

positions of networked robots, whereas the coordinated motion of the robots in the null space can be controlled through an acceleration feedforward or controller. Furthermore a robust acceleration consensus algorithm is proposed which can be utilized when there are delays on the communication links. In chapter 4, it is shown that if the null space controller in the acceleration consensus algorithm is chosen to be the force inputs of the robots, the acceleration consensus algorithm can realize multilateral teleoperation, which enables perfect position tracking of multiple robots, as well as force reflection among the operators. Furthermore force consesus filters are proposed to realize a completely distributed multilateral control system which can work under switching network topologies, and can circumvent communication blackouts in the network. In chapter 5, a novel scaled consensus algorithm is proposed which enables nodes on a weighted Laplacian network to track with some scaling the weighted average of the inputs. Following that, a networked acceleration controller based on this consensus algorithm is proposed, and utilized for the scaled motion control of robots. The scaled haptic consensus algorithm is proposed to realize scaled haptic teleoperation with multiple robots on a flexible network, with dynamically changing topologies. The proposed controller is a generalization of the four channel control architecture, which can provide perfect shared impedance matching among the environment and the operators or scale the impedances. Because this is a performance optimized architecture, it suffers from the same problems that surface in four channel teleoperation. This algorithm can not guarantee stability under time delay. However, local damping injection that was shown for stability of four channel teleoperators under time delay, can be applied to asymmetric multilateral teleoperation to provide robustness under time delay as well. In chapter 6, the damping injection method for four channel architecture based teleoperators that are unstable under time delay is proposed. It is shown that this method guarantees the delay independent stability, provides good position tracking without drift and enables high

frequency force reflection. It is also shown that L_2 stability can be achieved for four channel systems; this is an important result in the sense that the analysis can be further developed for slowly varying time delays. Experiment results show that the system can be stable even under fast varying time delays, therefore IQC methods could be employed in the future to analyze the stability of the system. Also the simple implementation of the method makes it applicable for multi-robot teleoperation under time delay. In chapter 7, a framework to analyze the delay independent stability of multilateral teleoperation systems is proposed, and local damping injection is introduced as a method to render the system delay independently stable. The validity of the method is with constant delay experiments. The method is shown to be efficient under fastly varying time delays as well, however the stability analysis for such cases is more difficult and might require IQC analysis.

Bibliography

[1] M. W. Spong, "Motion control of robot manipulators" in Handbook of Control (W. Levine, ed.), pp. 1339.1350, CRC Press, 1996.

[2] S. Komada, M. Ishida, K. Ohnishi and T. Hori "Disturbance observer based motion control of direct drive motors," *IEEE Transactions on Energy Conversion* , vol.6, Issue. 3, pp. 553–559, 1991.

[3] K. Ohnishi, M. Shibata and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Trans. on Mechatronics*, vol. 1, no. 1, pp. 56–67, 1996.

[4] Richard M. Murray and Karl J. Astrom and Stephen P. Boyd and Roger W. Brockett and Gunter Stein, "Future Directions in Control in an Information-Rich World", 2003.

[5] M. W. Spong, "The Interplay Between Robotics and Control Theory," Keynote Lecture, IEEE International Conference on Robotics and Automation, 2010.

[6] P. Antsaklis, J. Baillieul(Eds.), Special issue on networked control systems, IEEE Transactions. on Automatic Control 49(9), 2004.

[7] Y. Fujimoto, T. Yakoh and K. Ohnishi, "Analysis and synthesis of decentralized systems with informational connection" *IECON 2000 26th Annual Conference of the IEEE Industrial Electronics Society,* 2000.

[8] Y. Uchimura, T. Yakoh, K. Ohnishi, "A controller design method on decentralized systems" *Industrial Electronics Society, 2004*, Vol. 3, pp. 2961- 2966, November 2004.

[9] A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. on Automatic Control*, vol. 49, pp. 1465-1476, September 2004.

[10] R. Olfati-Saber, J. A. Fax, and R. M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems," Proceedings of the IEEE, vol. 95, no. 1, pp. 215-233, 2007.

[11] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. on Automatic Control*, vol. 49(9), pp. 1520-1533, 2004.

[12] R. Olfati-Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents" *Proc. Amer. Control. Conf.* , pp. 951-956, June 2003.

[13] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference*, December 2005.

[14] R. Olfati-Saber and J. S. Shamma, "Distributed Kalman filter with embedded consensus filters," *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference*, December 2005.

[15] D.P Spanos, R. Olfati-Saber and R. M. Murray, "Dynamic consensus on mobile networks" *The 16th IFAC World Congress, Praque, Czech Rep.*, 2005.

[16] D.P Spanos, R. Olfati-Saber and R. M. Murray, "Distributed sensor fusion using dynamic consensus" *The 16th IFAC World Congress, Praque, Czech Rep.*, 2005.

[17] D.P Spanos, "Distributed Gradient Systems and Dynamic Coordination," *Ph.D Thesis*, 2005.

[18] D. Lee and M. W. Spong, "Stable flocking of multiple inertial agents on balanced graphs" *American Control Conference*, June 2006.

[19] D. Lee, M.W. Spong, "Agreement with non-uniform information delays", in: Proceedings of the American Control Conference, Minneapolis, MN, pp. 756-761, 2006.

[20] U. Muenz, A. Papachristodoulou and F. Allgower, "Delay-dependent rendezvous and flocking of large scale multi-agent systems with communication delays." Proc. of the IEEE CDC, 2008.

[21] Lee, E.A. "Cyber physical systems: Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California", Berkeley, January 2008.

[22] Hokayem, P.F., and Spong, M.W., "Bilateral Teleoperation: An Historical Survey," Automatica, Vol. 49, No. 12, December 2006.

[23] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Trans. Automatic Control*, Vol. 34, No. 5, pp. 494-501, May 1989.

[24] G. Niemeyer and J. E. Slotine, "Stable adaptive teleoperation" IEEE. Transactions on Automat. Control, vol. 36, pp. 152–162, January 1991.

[25] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 4, pp. 426-434, 1989.

[26] Y. Yokokohji, T. Yoshikawa, "Bilateral control of master-slave manipulators for

ideal kinesthetic coupling- formulation and experiment" *IEEE Transaction on Robotics and Automation*, vol. 10, no. 5, pp. 605-620, October 1994.

[27] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 5, pp. 624-637, 1993.

[28] D. Lee and M. Spong, "Passive bilateral teleoperation with constant time delay", IEEE Trans. Robot., vol.22 no.2, pp.269.281, April 2006.

[29] Nuno, E. Ortega, R. Barabanov, N. Basanez, "A Globally Stable PD Controller for Bilateral Teleoperators," IEEE Transactions on Robotics, vol 24, No 3, June 2008.

[30] Y. Matsumoto, S. Katsura and K. Ohnishi, "An Analysis and design of bilateral control based on disturbance Observer," *Proc. of the 10th IEEE Int. Conf. on Industrial Technology*, pp. 802–807, 2003.

[31] K. Hashtrudi-Zaad, S. E. Salcudean: "On the Use of Local Force Feedback for Transparent Teleoperation," ICRA 1999, pp. 1863-1869, 1999.

[32] Kenji Natori, Kouhei Ohnishi, "A Design Method of Communication Disturbance Observer for Time-Delay Compensation, Taking the Dynamic Property of Network Disturbance Into Account," IEEE Transactions on Industrial Electronics, Vol. 55, No. 5, pp. 2152-2168, May, 2008.

[33] Ryogo Kubo, Noriko Iiyama, Kenji Natori, Kouhei Ohnishi and Hirotaka Furukawa, "Performance Analysis of a Three-Channel Control Architecture for Bilateral Teleoperation with Time Delay," IEEJ Trans. IA, Vol. 127, No. 12, pp.1224-1230, 2007.

[34] S. Katsura, W Iida and K. Ohnishi "Medical mechatronics- An application to haptic forceps," *Annual Reviews in Control* vol. 29, no.2, pp. 237-245, 2005.

[35] T. Shimono, K. Ohnishi, "Transmission of Force Sensation by Micro-Macro Bilateral Control with respect to Standardized Modal Space," *The 4th IEEE International Conference on Mechatronics, ICM '07* , May 2007.

[36] S. Katsura, Y. Matsumoto and K. Ohnishi, "A Realization of haptic training system by multilateral control," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 5, pp. 1196–1205, 2005.

[37] S. Katsura, Y. Matsumoto and K. Ohnishi, "Realization of law of action and reaction by multilateral control," *IEEE Trans. on Industrial Electronics*, vol. 53, no. 6, pp. 1935–1942, 2005.

[38] S. Katsura, T. Suzuyama, K. Ohishi: "A Realization of Multilateral Force Feedback Control for Cooperative Motion," IEEE Transactions on Industrial Electronics, Vol. 54, No. 6, pp. 3298-3306, December 2007.

[39] S. Katsura, K. Ohishi; "Modal System Design of Multi-Robot Systems by Interaction Mode Control," IEEE Transactions on Industrial Electronics, Vol. 54, No. 3, pp. 1537-1546, June 2007.

[40] T. Shimono, S. Katsura, R. Kubo and K. Ohnishi, "Multilateral control for skill education based on haptic data storage," *IEEE International Conf. on Industrial Technology*, pp. 334- 339 , December 2006.

[41] D. Lee and M. W. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: theory," *IEEE Int'l Conf. on Robotics and Automation*, 2005.

[42] P. F. Hokayem, D. M. Stipanovic, and M. W. Spong. "Semiautonomous Control of Multiple Networked Lagrangian Systems," International Journal of Robust and Nonlinear Control, November 2008.

[43] T. Murakami, F. Yu and K. Ohnishi, "Torque sensorless control in multidegree-of-freedom manipulator," *IEEE Trans. on Industrial Electronics*, vol. 40, no. 2, pp. 259–265, 1993.

[44] S. Katsura, Y. Matsumoto, K. Ohnishi, "Modeling of Force Sensing and Validation of Disturbance Observer for Force Control" *IEEE Transactions on Industrial Electronics* ; V.54 No.1 pp.530-538, 2007.

[45] K. Hashtrudi-Zaad and S.E. Salcudean "Analysis and evaluation of stability and performance robustness for teleoperation control architectures," *IEEE International Conference on Robotics and Automation, ICRA '00* , 2000.

[46] U. Tumerdem, K. Ohnishi, "Robust Four Channel Teleoperation under Time Delay by Damping Injection," IEEE International Conference on Mechatronics, Apr. 2009.

[47] K. Hashtrudi-Zaad and S.E. Salcudean, "Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators," The International Journal of Robotics Research, 20(6), pp. 419.445, June 2001.

[48] Arjan van der Schaft, L2-Gain and Passivity Techniques in Nonlinear Control, Springer Verlag, London, 2000.

[49] Nikhil Chopra and Mark W. Spong, "Delay-Independent Stability of Interconnected Systems with Finite L_2-Gain," IEEE Conf. on Decision and Control, New Orleans, LA, December, 2007.

[50] M. Vidyasagar, Input-output analysis of large-scale interconnected systems : decomposition, well-posedness, and stability, Springer-Verlag, 1981.

[51] Polushin I.G., P.X. Liu, and C.-H. Lung "A Force Reflection Algorithm for

Improved Transparency in Bilateral Teleoperation with Communication Delay," IEEE/ASME Transactions on Mechatronics, Vol. 12, No. 3, pp. 361 - 374, 2007.

[52] H. Kazerooni, T.I Tsay and C.L Moore, "Telefunctioning: an approach to telerobotic manipulations," in Proc. Amer. Contr. Conf. San Diego, CA, pp. 2778-2783, May 1990.

[53] Daisuke Yashiro and Kouhei Ohnishi, "L2 stable four-channel control architecture for bilateral teleoperation with time delay," Tenth International Workshop on Advanced Motion Control, pp. 324-329, March 2008.

[54] Jing Wang and Elia, N. "Consensus over networks with dynamic channels," American Control Conference, pp. 2637 - 2642, June 2008.

[55] J. Chen and H.A. Latchman, "Frequency sweeping tests for stability independent of delay," IEEE Trans. Autom. Control 40, pp. 1640–1645, 1995.

[56] J. Michael Brown, "Factors affecting the Z-Width of a haptic display," IEEE Conference on Robotics and Automation, pp. 3205-3210, 1994.

[57] Tomoyuki Shimono, Seiichiro Katsura, and Kouhei Ohnishi: "Abstraction and Reproduction of Force Sensation from Real Environment by Bilateral Control," IEEE Transactions on Industrial Electronics, Vol. 54, No. 2, pp. 907-918, April, 2007.

[58] Peer, A. Buss, M. "Robust Stability Analysis of a Bilateral Teleoperation System Using the Parameter Space Approach," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2350-2356, 2008.

[59] Adams, R.J. Hannaford, B "Control law design for haptic interfaces to virtual reality," *IEEE Transactions on Control Systems Technology* vol. 10, no.1, pp 3-13, 2002.

[60] H. Flemmer, J. Wikander, "Transparency and Stability Analysis of a Surgical Teleoperator System," 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03), pp.382, 2003.

[61] S.I Niculescu , D. Taoutaou, R. Lozano "Bilateral teleoperation with communication delays" *International Journal of Robust and Nonlinear Control*, vol. 13,no.9, pp. 873-883, 2003.

[62] A. Eusebi and C. Melchiorri, "Force Reflecting Telemanipulators with Time-Delay: Stability Analysis and Control Design," *IEEE Transaction on Robotics and Automation*, vol.14, no.4, pp. 635-640, 1998.

[63] T. Shimono, S. Katsura, S. Susa, T. Takei, K. Ohnishi, "Transmission of Force Sensation by Micro-Macro Bilateral Control with Respect to Standardized Modal Space," The 4th IEEE International Conference on Mechatronics, ICM '07-Kumamoto, pp. 1-6, 2007.

[64] Skogestad, S. and Postlethwaite, I. Multivariable feedback control - analysis and design. Wiley, Chichester, 2005. 572 pp (1996); 2nd Edition, 2005.

[65] C.-Y. Kao and A. Rantzer. "Stability analysis of systems with uncertain time-varying delays". Automatica, vol.43: pp. 959-970, 2007.

[66] Hartley, H. A. High fidelity, Audio Design Handbook. New York, New York: Gernsback Library. p. 200, 1958.

[67] Tan, H. Z., Srinivasan, M. A., Eberman, B., Cheng, B. "Human factors for the design of force-reflecting haptic interfaces," Dynamic Systems and Control (ASME), 1994.

Achievements

Journal Publications

1. Ugur Tumerdem, Kouhei Ohnishi "Delay-Independent L_2 Stability of Four-Channel Bilateral Teleoperators with Damping Injection". IEEJ Transactions on Industry Applications, accepted for publication.

2. Ugur Tumerdem, Tomoyuki Shimono and Kouhei Ohnishi, "Asymmetric Multilateral Teleoperation through Scaled Consensus Reaching on Graphs," IEEJ Transactions on Industry Applications, Vol 129, No 10, pp.972-980

International Conferences

1. Ugur Tumerdem, Kouhei Ohnishi, "L_2 Stability Analysis of Four Channel Teleoperation and Experiments under Varying Time Delay", IEEE International Conference on Advanced Motion Control, Nagaoka, Japan, 24th March 2010

2. Ugur Tumerdem, Kouhei Ohnishi, "Delay Independent L_2 Stable Multilateral Teleoperation with Damping Injection", IEEE International Conference on Industrial Technology ,Vina Del Mar, Chile, 13-17 March 2010

3. Ugur Tumerdem, Kouhei Ohnishi, "Robust Four Channel Teleoperation under Time Delay by Damping Injection", IEEE International Conference on Mechatronics,Malaga, Spain, April 2009

4. Ugur Tumerdem and Kouhei Ohnishi, "Multi-robot Teleoperation under Dynamically Changing Network Topology," IEEE International Conference on Industrial Technology, 10-13 February 2009 Melbourne, Australia.

5. Ugur Tumerdem, Tomoyuki Shimono, Kouhei Ohnishi, "Micro-Macro Multilateral Teleoperation through Scaled Information Flow, " The 34th Annual Conference of the IEEE Industrial Electronics Society, IECON'08, Orlando, Florida, November, 2008

6. Ugur Tumerdem and Kouhei Ohnishi "Haptic Consensus in Multilateral Teleoperation,"2008 IEEE International Symposium on Industrial Electronics, ISIE'08, Cambridge, UK, June-July 2008

7. Ugur Tumerdem and Kouhei Ohnishi, "Acceleration Consensus for Networked Motion Control of Telerobots,"The 10th IEEE International Workshop on Advanced Motion Control, AMC'08, Trento, Italy, March 2008.

8. Ugur Tumerdem and Kouhei Ohnishi, "Haptic Consensus in Bilateral Teleoperation," The 4th IEEE International Conference on Mechatronics, ICM'07 Kumamoto, Japan, May 2007.

Domestic Conferences

1. Ugur Tumerdem, Tomoyuki Shimono, Kouhei Ohnishi, "Scaled Haptic Consensus and Multilateral Teleoperation," Technical Meeting on Industrial Instrumentation and Control, IEE Japan. March 2008, Tokyo Denki University, Tokyo, Japan.

2. H.A. Varol, Ugur Tumerdem, M. Unel, "Serbest Sekilli Nesnelerin Scara Robotun Gorme Tabanli Denetimi ile Manipulasyonu," Otomatik Kontrol Ulusal Toplantisi 2005 (TOK 2005), 2-3 June, 2005. (Turkish)

Technical Presentations

1. Ugur Tumerdem, Kouhei Ohnishi, "Four channel bilateral teleoperation with varying time delay: experiments" International Workshop on Vision and Control for Access Space 2009, In Conj. with IEEE ACCV 2009, 24 Sept. 2009, Xian, China

2. Ugur Tumerdem, Kouhei Ohnishi, "Robust Four Channel Teleoperation under Time Delay ," Keio-ECN Joint Workshop 2008, 24 October 2008 Ecole Centrale de Nantes, Nantes, France

3. Ugur Tumerdem, Kouhei Ohnishi, "Multirobot Micro-Macro Teleoperation through Scaled Haptic Consensus," Keio-XJTU Joint Workshop on Electronic and Information Engineering 2008, 13-14 March 2008 Xian Jiaotong University, Xian, China