Improving Transparency in Network-based Multi-user Haptic Simulations

IMPROVING TRANSPARENCY IN NETWORK-BASED MULTI-USER HAPTIC SIMULATIONS

BY

SINA NIA-KOSARI, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2009)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Improving Transparency in Network-based Multi-user Haptic Simulations
AUTHOR:	Sina Nia-kosari B.Sc., (Electrical Engineering) Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
SUPERVISOR:	Dr. Shahin Sirouspour
NUMBER OF PAGES:	xiii, 99

To my family

Abstract

Collaborative haptic simulations allow multiple users in a virtual environment to simultaneously interact with shared virtual objects. The implementation of shared virtual environments over a network removes the geographical barriers and enables users from around the globe to modify the environment and in addition feel the presence of other users. However, network issues arise in the communication of data over a network such as the Internet. Communication channel delay, jitter, packet loss and limited packet transmission rate can adversely affect the performance of collaborative haptic systems and may even cause instability.

This thesis builds upon our group's recent work in distributed networked haptics (Fotoohi *et al.* [31]). The proposed distributed peer-peer architecture improved the haptic simulation performance over the centralized architecture by providing local high-rate feedback to the users in a Local Area Network (LAN). Virtual spring-damper couplers synchronized the multiple copies of the virtual environment and coupled the users to the virtual objects.

Forming on the above distributed architecture, this thesis proposes methods for improving the performance and stability of shared haptic environments with a stronger emphasis on the effects of time delay in the context of Internet communication. To this end new quantitative measures are presented for quantifying the fidelity of haptic simulations in such environments. User's perceived admittance and discrepancy among local copies of

virtual objects are considered in defining these measures. Furthermore, state prediction and feedforward schemes are proposed to compensate for the negative effects of the network communication delay on the transparency and stability of the haptic simulation. An optimization problem is formulated for selecting the virtual coupling gains that can enhance the performance while maintaining system stability. The solution to the this problem provides us with the set of control parameters that optimize the defined performance measures.

A three user distributed architecture is presented to show the extension of the proposed methods to haptic simulations involving more than two users. Numerical analysis and haptic interaction experiments over the Internet are carried out to demonstrate the effectiveness of the proposed approach in two-user and three-user platforms. The obtained analytical and experimental results verified improvements by the prediction and feedforward mechanisms.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Shahin Sirouspour for his guidance and support through the course of my M.A.Sc. program. This thesis would not have been possible without his stimulating suggestions and encouragement.

Special thanks to my family for their unwavering love and support throughout my life.

I would like to thank my colleagues at the Telerobotics, Haptics and Computational Vision Laboratory at McMaster University for sharing their knowledge and experience with me and for their help in my research. Special thanks to Mehryar Rahmatian and Pardis Miri for hosting the packet mirror program.

I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this research.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The Internet, ever since its public debut in early 1990s, has grown enormously to become the main medium for the exchange and management of information in the developed world and is rapidly expanding into developing countries. This global network has fundamentally transformed the way knowledge and information are shared and has spurred an endless stream of applications in e-business, tele-health, education and scientific research, news and multimedia, social networking, and gaming and entertainment among others.

The growth of the Internet has been gradually removing geographical barriers to information exchange by providing a widespread network through which people around the globe can access information and even interact and communicate with each other. Existing Internet-based applications are mostly restricted to the exchange of textual, visual, and auditory data. For instance in e-conferencing and network-based gaming, users can share multimedia contents in the form of voice and video in real time. Network and human-computer-interface technologies have advanced to the point that more sophisticated and interactive forms of network communication are now becoming feasible. In particular, Internet-based collaborative virtually-reality (VR) environments which permit users across a network to co-exist and interact in a shared virtual world can be developed. Ideally, through the exchange of multi-modal sensory observations among the users' workstations, such systems should create a sense of tele-co-presence in the virtual environment.

The sense of force and kinesthetic feedback, also known as haptics, are essential in the human exploration and perception of his/her surrounding environment in daily activities. The lack of haptic feedback has fairly limited the effectiveness of exiting VR applications in achieving the goal of tele-co-presence. The addition of this sensing modality will be crucial for the success of VR systems in many important network-based applications. Players' experience in network-based distributed gaming can be greatly enriched by the introduction of haptic feedback capability. The use of force-feedback interfaces would provide a level of realism far beyond those offered by the exiting computer games. Network-based virtual communities, which have already gained remarkable popularity, can similarly benefit from the use of haptics. The ability to interact with virtual objects and other people over the Internet would create enormous possibilities for new improved methods of distance learning and education. In computer-aided design and manufacturing, product designers can employ haptic-enabled networked design tools to remotely collaborate in virtual prototyping and design of new products. The rise of the Internet has spurred a remarkable interest in new technologies for remote delivery of healthcare services. There are promising tele-health applications which can significantly benefit from network-based haptic interaction. In tele-rehabilitation, for instance, a therapist can remotely guide a patient in the comfort of his/her home in performing physical exercises inside a virtual environment. Network-based

VR simulators can greatly facilitate remote training and mentoring of medical practitioners in performing complex medical interventional procedures. They can also be integrated into advanced computer-assisted surgical systems enabling interactive collaboration among surgeons with complimentary expertise across a network.

The above examples and many other underscore the strategic importance of research in the area of network-based haptics to address some of the key issues currently inhibiting widespread adoption of such technologies.

1.2 Problem Statement

Existing haptic control algorithms are primarily designed for single-user applications and are only effective when the entire simulation can be executed on a dedicated computing platform with control update rates in excess of 1kHz. In network-based haptics, however, the fidelity and stability of the simulation can drastically degrade due to network impediments such as time delay, delay jitter, limited packet transmission rate, and packet loss.

Communication over a Wide Area Network (WAN) can be subject to significant delays. The communication delay over a WAN such as the Internet depends on factors such as geographical distribution of the users and the network traffic. This delay can drastically affect the transparency of the haptic simulation and may even result in instability. Compared to audio and visual feedback systems, the delay in networked haptic interactions imposes greater difficulties. This is partly due to the fact that sight and hearing are passive senses, whereas, touch is an active sense. In other words, in contrast to haptic systems, there is no exchange of force and energy between the users and virtual objects in visual and audio feedback systems. Although the performance and stability of audio and visual feedback systems are affected by the network communication delay, instability occurs at larger delays compared to haptic feedback systems.

Moreover, depending on the network condition, the packet transmission rate is limited. In visual feedback systems an update rate of 30 frames per second tricks the eye into perceiving motion. Haptic applications, however, require an update rate of 1kHz for high fidelity rendering of rigid objects (Basdogan and Srinivasan [11]). At this point it is useful to distinguish the network packet transmission rate from the network bandwidth. In shared haptic applications the amount of data to be transmitted is usually small. However, this data must be transmitted with minimum latency. Therefore, networked haptic applications require a fast packet transmission rate, although a large bandwidth is not necessary. In our experience, the maximum achievable packet rate over the network was well below the 1kHz. The effects of limited packet transmission rate must be considered in design and analysis of network-based haptic systems.

The mentioned restrictions necessitate the development of new dedicated control algorithms that can guarantee a high level of performance, i.e. tele-co-presence, over the Internet while maintaining the system stability.

1.3 Thesis Contributions

In this thesis, the addition of force feedback to multi-user virtual environments is considered. In particular Internet-based haptic simulations in which users across a Wide Area Network (WAN) participate in the haptic interaction are investigated.

Built on the distributed multi-user architecture proposed by [31], this thesis introduces methods for improving the performance and stability of shared haptic environments with a stronger emphasis on the effects of time delay in the context of Internet communication. To this end new quantitative measures are presented for quantifying the fidelity of haptic simulations in such environments. User's perceived admittance and discrepancy among local copies of virtual objects are considered in defining these measures. Furthermore, state prediction and feedforward control schemes are proposed to compensate for the negative effects of the network communication delay on the transparency and stability of the haptic simulation.

Virtual spring-damper couplers synchronize the multiple copies of the virtual environment and in addition couple the users to the virtual objects in the distributed architecture. Based on the defined performance measures, an optimization problem is formulated for selecting the virtual coupling gains that can enhance the performance while maintaining system stability. The solution to the this problem provides us with the set of control parameters that optimize the defined performance measures.

A three user distributed architecture is presented to show the extension of the proposed methods to haptic simulations involving more than two users. Numerical analysis and haptic interaction experiments over the Internet are carried out to demonstrate the effectiveness of the proposed approach in two-user and three-user platforms. The obtained analytical and experimental results verified improvements by the prediction and feedforward mechanisms.

The main contributions of the thesis can be summarized as:

- Definition of objective and quantitative measures of transparency in distributed haptic simulations.
- Formulating an optimization problem for selecting the appropriate virtual coupling gains.
- Improving the transparency of the shared haptic interaction by utilizing predictive and feedforward/feedback schemes.

- Experimental Evaluation of the proposed methods over the Internet in two-user and three-user platforms.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. A review of the haptics literature is presented in Chapter 2. Centralized and distributed architectures for multi-user haptic interaction are presented in Chapter 3 and a state-space multi-rate model is provided. New measures for evaluating the performance of the distributed architecture are introduced in Chapter 4 and an optimization method for selecting the virtual coupling gains in this architecture is presented in this chapter. Distributed predictive and feedforward control schemes are proposed in Chapter 5 to minimize the effects of the network delay and to improve the performance of the collaborative haptic interaction. Optimization results are presented in Chapter 6. A haptic platform for two-user haptic experiments and the results of the two-user experiments over the Internet are presented in Chapter 7. Chapter 8 extends the proposed method to three users. The thesis is concluded in Chapter 9 where some possible directions for future research are also suggested.

1.5 Related Publications

- Niakosari, S.; Sirouspour, S., ”Improving transparency in network-based haptics,” Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. pp.547-552, 18-20 March 2009 [67].

Chapter 2

Literature Review

Haptics research is a truly multidisciplinary field between robotics, control systems, computer science, mechanical engineering, mechatronics and psychology and others. Haptics literature, therefore, contains works from different disciplines and on a variety of subjects related to haptics research. This chapter presents a brief review of the literature and works relevant to this thesis.

2.1 Haptic Devices

Haptic devices are robots that enable users to feel and manipulate remote or virtual objects. In a tele-operation system a user can perform a task remotely with the aid of master and slave robots. The user controls the slave robot through the master robot. Ideally, the user would feel present at the remote side and be able to manipulate the remote environment as if he/she was at the remote location. In a haptic simulation system the slave robot and the environment are virtual. As the user moves the haptic device, the virtual environment generates and sends the proper control commands to the haptic device. Ideally, the user should feel as if he/she was interacting with a real object.

Although haptic devices for manipulation of human tactile sense have been produced, this thesis focuses on devices that are capable of generating motion and exerting force to the users. Haptic devices are usually equipped with sensors and encoders to measure the position/angle of the joints. Force sensors may be employed in some devices to measure the force applied to the device by the user. Depending on its input and output signal and mechanical behavior, a haptic device can be classified as admittance- or impedance-type.

2.1.1 Admittance-type Devices

The input command to an admittance-type haptic device is position. In admittance-type devices force sensors measure the force applied to the device. Some admittance-type devices are also equipped with encoders for measuring position. Based on these measurement, an admittance-type controller will make the proper displacement in the haptic device.

Admittance-type haptic devices are capable of generating large forces. Therefore, they are suitable for rendering high stiffness environments. Admittance-type devices are commonly used in large workplaces. However, they usually posses large inertia and low backdrivability and are therefore not suitable for rendering low masses (Van der Linde *et al.* [81]). The FCS HapticMaster (see Fig. 2.1(a)) (Van der Linde *et al.* [81]) and VIRTUOSE 3D15-25 by Haption (see Fig. 2.1(b)) are examples of admittance-type haptic devices.

2.1.2 Impedance-type Devices

Including most commercially available haptic devices, impedance-type devices measure the position of the haptic device and exert the proper force based on the displacement made by the user. Some impedance-type devices are also equipped with force sensors.

(a)

(b)

Figure 2.1: Examples of admittance-type devices: (a) Hapticmaster by FCS (Image from Haptic Photo Gallery); (b) VIRTUOSE 3D15-25 by Haption (Image from http://www.haption.com)

Compared to admittance-type devices, impedance-type devices have low inertia and high backdrivability and are usually lightweight. These characteristics make the impedance-type devices more suitable for rendering of free motion, soft contact and low masses. The Phantom series of products by SensAble Technologies is an example of the impedance-type devices. PHANTOM Premium 1.5 Haptic Device (see Fig. 2.2(a)) is an instance of this series with 3 degrees-of-freedom (DOF) positional sensing and 3 degrees-of-freedom force feedback. Originally designed by Prof. Tim Salcudean at the University of British Columbia, the planar Pantograph by Quanser (see Fig. 2.2(b)) is another impedance-type device with three degrees of freedom allowing planar translation and unlimited rotation about a single axis.

(a) (b)

Figure 2.2: Examples of impedance-type devices: (a) Phantom by SensAble Technologies; (b) Planar Pantograph by Quanser

2.2 Haptic Rendering

Computing the correct interaction forces between the user and the virtual object is called haptic rendering. Haptic rendering algorithms can be very different from one application to another but most impedance-type haptic rendering algorithms consist of the following three blocks (Salisbury *et al.* [75]).

Collision Detection Collision Detection algorithms are responsible for detecting collision between the haptic interface and a virtual object. These algorithms may also derive some contact information such as angle of contact, velocity of contact, etc.

Force Response Force response algorithms calculate the forces between the virtual object and the haptic interface based on the information from the collision detection unit and the position of the virtual objects and haptic interface.

Control Algorithms Control Algorithms determine the forces to be applied to the user through the haptic interface. Due to hardware or implementation limitations these

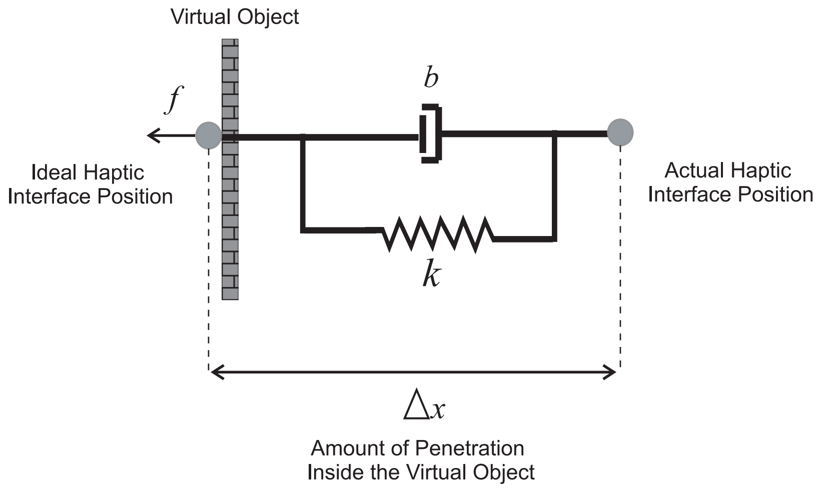

Figure 2.3: Penalty based method in force generation.

forces may differ from the output of the Force Response unit.

Different methods for generation of contact forces have been proposed in the literature. Although force calculation methods that take into account the shape of the haptic device exist (e.g. Ho *et al.* [47], McNeely *et al.* [62]), the focus in this work is on methods that assume a simpler form of contact. Assuming that the user interacts with the virtual world with a point probe, the general 6 DOF problem can be simplified to calculating the forces along the x,y and z axes. The point probe represents the end point of the haptic device and is commonly referred to as Haptic Interface Point (HIP) or avatar. Methods based on this assumption can be divided into two categories: penalty based methods and impulse based methods.

In penalty based methods the forces applied to the user are proportional to the penetration of the haptic device inside the virtual object. Fig. 2.3 shows a common one DOF implementation of the penalty based method where spring and damper elements are used to generate the interaction forces. The interaction forces are calculated by:

$$f = k\Delta x + b\Delta\dot{x} \tag{2.1}$$

God object and proxy algorithms were introduced by Zilles and Salisbury [88] and Ruspini *et al.* [74], respectively, as methods to generate interaction forces with three dimensional objects. Based on the same principle both methods used a virtual point, referred to as god object or proxy, to represent a realistic point of contact on the surface of the virtual object. Interaction forces can then be generated by placing a spring and damper between the god object/proxy and the actual haptic interface position. Walker and Salisbury [82] enhanced the performance of proxy method by restricting the proxy location to the edges and vertices of the object.

Penalty-based methods generate forces based on the amount of penetration of the haptic interface inside the virtual object. Due to the discrete-time implementation of the haptic simulation, the amount of stiffness that can be rendered with the penalty-based methods are limited. Impulse based methods solve this problem by applying a series of impulses upon contact and therefore generating the feeling of rigid contact. An impulse based method has been employed by Mirtich and Canny [65] for dynamic simulation of contact between rigid objects. Upon detection of contact a series of impulses are calculated and applied to the objects to avoid interpenetration of objects. Chang and Colgate [21] used the impulse based simulation for haptic rendering of contact with rigid objects. Kuchenbecker *et al.* [53] proposed a method for rendering contact using open loop force pulses. The approach improves the users' sense of touch by applying high frequency impulses. Constantinescu *et al.* [25] combined the penalty based and impulse based methods by applying impulsive forces upon contact and penalty and friction forces during contact. Abdossalami and Sirouspour [2] proposed two adaptive nonlinear controllers for interaction with an impedance- or admittance-type virtual environment.

2.3 Applications

The growing interest in haptics research has brought about new application in several fields ranging from medical training to gaming and entertainment. In surgical training haptic simulators have been used for laparoscopic surgery (Webster *et al.* [85], Acosta and Temkin [3], Hu *et al.* [50]), minimally invasive surgery (Basdogan *et al.* [13], Richards *et al.* [73]), prostate biopsy (Wang and Fenster [84]), and needle insertion (Gerovichev *et al.* [34]). These surgical simulators give trainees the opportunity to practice complex procedures on the simulator many times before operating on the patient.

In rehabilitation haptic interfaces have been used to help patients regain their abilities. Broeren *et al.* [16] designed a computer game as a training utility to promote motor rehabilitation. McLaughlin *et al.* [61] developed a virtual environment with different levels of haptic feedback for post stroke rehabilitation. Haptic virtual reality simulators were used by Houtsma and Van Houten [49] for post-stroke upper-limb rehabilitation. Mali and Munih [59] designed and built a 2 DOF haptic device with a tendon-driven transmission system. The device can generate forces of up to 10 N and is suitable for finger rehabilitation exercises.

Kim and Park [51] proposed haptic simulation methods for a virtual dental training simulator in which the students could learn dental procedures such as drilling and filling the cavities with realistic tactual feelings. Cao *et al.* [19] designed a six DOF haptic device for dental surgery training system. Dai *et al.* [26] proposed a multirate control algorithm to guarantee the stability of haptic dental training system. A two level up-sampling method was introduced for high frequency force interpolation.

Haptic technology has been implemented in applications for blind and visually impaired people. Levesque [57] carried out a survey on the use of haptics in the design of aids for the blind. Yu *et al.* [87] developed a system for making graphs accessible to visually impaired people through haptic and audio media. Lécuyer *et al.* [56] employed force, thermal and auditory feedback to enable visually impaired people to explore and navigate through virtual environments.

Force feedback has been used in the literature for educational purposes. Minogue and Jones [63] examined the role of touch in cognition and learning. Williams *et al.* [86] developed a program for teaching simple machine concepts such as lever, pulley and inclined plane to elementary school students. Using a 1 DOF force feedback slider (FFS) Kretz *et al.* [52] developed a software application that allows users to study the laws of physics.

Haptic feedback can enhance the users' experience in video games. Force feedback joysticks and haptic based games that allow a physical feeling of the games have been developed. The "Haptic Battle Pong", (Morris [66]), is a network game that uses three-degree-of-freedom force-feedback and six-degree-of-freedom position input. Andrews *et al.* [8] discuss the integration of haptic feedback in a shooter style video game which uses a 3D game engine. Gourishankar *et al.* [39] worked on developing a haptic device and virtual environment for playing billiards.

Brewster [15] investigated the possibility of applying haptic technology and virtual reality to cultural applications. In a museum, for instance, visitors living far from the museum can see and feel the objects at a distance. Orozco *et al.* [69] studied the extraction of biometric features from the measured data in a haptic interaction. Such methods could be used for user authentication as an alternative to passwords. Other applications of haptics are in e-commerce (El-Far *et al.* [28]), arts (Frank Dachille *et al.* [32], Blanch *et al.* [14]), scientific discovery (Brooks [17]), and computer-aided design (Hollerbach *et al.* [48]).

2.4 Multi-user Haptics

Shared virtual environments over a network have attracted a great deal of interest in recent years. Shared virtual environments have been implemented for training (Stansfield *et al.* [79]), research and development (Macedonia and Noll [58]), and gaming (Shaw and Green [77]). Despite the growth of the Internet most networked virtual reality applications have been limited to the exchange of textual, audio and visual information. Haptic feedback has been noticeably absent from most exiting interactive network applications. Beside factors such as the high cost and relative complexity of haptic devices, this is partly due to the inadequacy of conventional haptic control algorithms for multi-user interaction over large distances. The emergence of low-cost haptic interfaces for mass market use has created a unique opportunity for the growth of network intensive haptics applications and has generated a great deal of interest in this area. The importance of haptic feedback in multi-user virtual environments has been underlined by an experimental study by Basdogan *et al.* [12] where the addition of haptic feedback has been shown to significantly improve the sense of togetherness and task performance. Buttolo *et al.* [18] classified the interaction with shared virtual environments as static, collaborative, and cooperative based on how the users can manipulate the virtual environment. In static environments users are able to explore the virtual environment, but cannot modify the environment. Collaborative environments allow users to take turns in manipulating the virtual objects. However simultaneous manipulation of objects are not allowed. In cooperative environments more than one user can interact with a virtual object at a time.

Several issues arise in haptic rendering of networked virtual environments. Network delay can drastically affect the sense of co-presence in a shared virtual environment. In addition to transmission time of the network medium, latency can be caused by network

Destination	Distance (km)	Mean round-trip delay (ms)	Delay jitter RMS (ms)	Packet rate (Hz)
Bristol, UK	230	14.3	0.34	69.9
Amsterdam, The Netherlands	496	27.6	4.54	36.2
North Carolina, USA	6062	104	0.038	9.62
LAN	–	0.46	0.047	2170

Table 2.1: Characteristics of network communication from the University of Manchester to various locations. Results taken from Marsh *et al.* [60]

traffic and the time required for unpacking and processing messages. Moreover, users' network bandwidth, distance between the users and the number of routers in between the users can affect the amount of delay over a network (Gutwin [42]). Effects of network delay on performance in a collaborative virtual environment have been investigated by Park and Kenyon [71]. Although users were able to perform the defined tasks with delays of up to 200 ms, it was observed that users tended to adopt a move and wait control strategy.

Packets aimed for the same destination may be sent through different routes and may therefore be subject to different delays. Variance in network transmission time, also known as delay jitter, can degrade the users' performance in cooperative virtual environments. Hikichi *et al.* [45] evaluated the effect of delay jitter on haptic collaboration over the internet. Shirmohammadi and Ho Woo [78] employed visual cues called decorators to inform the users of delay and jitter. The decorator would change color depending on the delay and jitter over the network. Marsh *et al.* [60] measured network characteristics including latency and jitter in 8000 packets sent from the University of Manchester, UK to various locations. Table 2.1 shows some results from their measurements. In addition to delay and jitter, packet loss is another issue in packet switched networks.

Due to hardware limitations packet transmission is limited over the Internet. Multi-rate haptic control has been proposed in the literature to tackle this problem. Sankaranarayanan and Hannaford [76] proposed one client-server and two peer-peer architectures for multi-user haptic simulations running at different network rates. Fotoohi *et al.* [31] employed a peer-peer multi-rate approach in which high-rate haptic rendering of local objects were not affected by the slow packet transmission rate over the network. Cho *et al.* [23] used wave variables to provide a stable low-rate haptic interaction.

2.5 Stability

Haptic rendering algorithms are implemented using computers and are therefore sampled-data systems. The effects of the sample and hold operation on stability of the haptic interaction has been investigated by Colgate *et al.* [24], Minsky *et al.* [64]. Abbott and Okamura [1], Diolaiti *et al.* [27] addressed issues caused by sensor quantization in haptic rendering. In addition, actuator and haptic device dynamics affect the stability of the haptic interaction.

Gil *et al.* [36] used the Routh-Hurwitz criterion to study the stability of 1 DOF haptic interface and to derive parameter conditions that garuantee the stability of the system. Eom *et al.* [29] utilized the small gain theorem to design a robust stabilizing haptic controller. Lyapunov analysis (Diolaiti *et al.* [27]), deadbeat control (Gillespie and Cutkosky [38]), and port-Hamiltonian systems (Stramigioli *et al.* [80]) have also been used for stability analysis of haptic simulations.

In networked haptics, delay, jitter, packet loss and limited bandwidth and packet transmission rate can adversely affect the stability of the haptic interaction and may even result in unstable systems. The concept of passivity has been widely used in stability analysis of tele-operation (Anderson and Spong [6, 7]) under time delay and was reformulated using wave variables by Niemeyer and Slotline [68]. A system is considered passive if energy is not supplied from within the system. In the context of haptics the condition is restated as: the energy supplied to the hand being less than or equal to the energy supplied by it (Hayward and MacLean [44]). Arioui *et al.* [9], Carignan and Olsson [20], Arioui *et al.* [10] utilized the passivity-based techniques for stability analysis of multi-user haptic rendering. While these methods can ensure the stability of haptic interaction, they usually yield poor transparency due to their inherent conservatism.

Gil *et al.* [37] proposed a stability condition for delayed haptic systems including the effect of delay and virtual damping. Fotoohi *et al.* [31] analyzed the stability of a centralized and peer-peer multi-user haptic system taking into account the network delay and multi rate nature of the system. Haptic data compression was studied by Ortega [70], and Hinterseer *et al.* [46] in order to reduce haptic data traffic for low-bandwidth networked haptics.

2.6 Performance Measures

Various methods for the evaluation of the fidelity of haptic interactions have been proposed in the literature. Hannaford [43] asked users to perform a predefined task. Data was recorded and analyzed and factors such as task completion time and visual observation of task errors were used in evaluating the haptic interaction. Alhalabi and Horiguchi [4] employed mean and standard deviation of the perceived force in a cooperative haptic interaction in measuring performance of the system. Root Mean Square (RMS) and peak position error and RMS of force applied to the users were compared by Sankaranarayanan and Hannaford [76] for different peer-to-peer and client-server architectures. Kuchenbecker *et al.* [54] asked subjects to rate the level of realism they encountered in haptic interaction. These measures may provide meaningful results in practice. However, due to their ad-hoc and/or subjective nature, they have very limited utility in an objective analysis of performance as well as control synthesis.

Griffiths and Gillespie [41] characterized the behavior of tele-operator feedback systems using analytical transparency and tracking measures. Dependency of these measures on tele-operator system parameters were investigated. A performance measure, referred to as *distortion* was defined by Griffiths [40] which measured the virtual environment dynamics and actual closed-loop dynamics perceived by the user. Fotoohi *et al.* [31] compared the admittance rendered to the users and the admittance of the object being simulated. Frequency response of the admittances were used as graphical measures of performance in the peer-peer architecture.

Chapter 3

Cooperative Haptics: Background and Modeling

From an architectural perspective, prior work on network-based haptics can be classified into two main categories: centralized and distributed. In this chapter the two architectures are introduced and a comparison between them from an earlier research conducted in our group (Fotoohi *et al.* [31]) is provided. Also a modeling technique for deriving the system dynamics are presented. Although the models presented in this chapter are for a two-user configuration, the same method can be employed in modeling of haptic interactions involving more than two users.

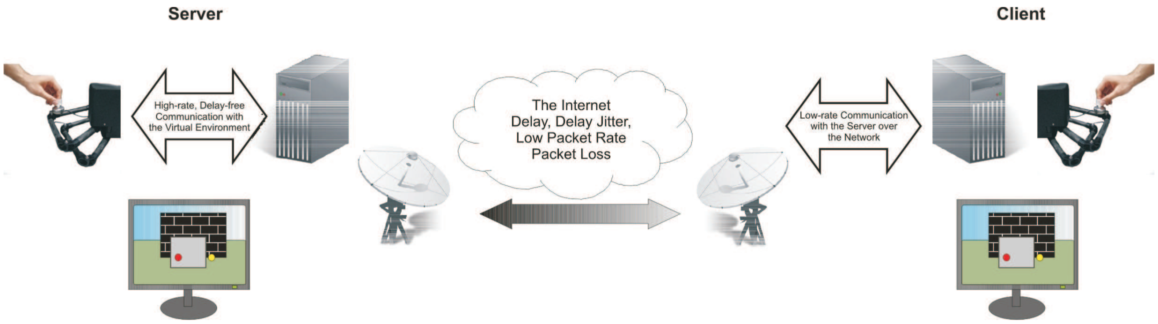

Figure 3.1: General representation of a centralized dual-user haptic system over a packet switched network.

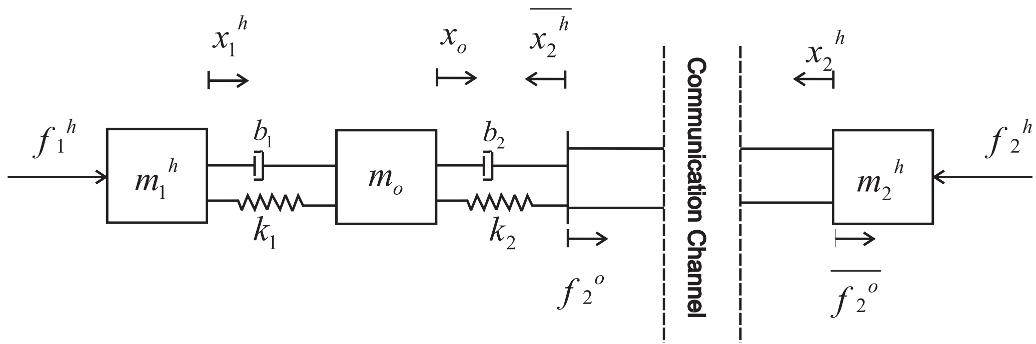

Figure 3.2: Model of centralized single-axis cooperative haptics.

3.1 Architectures

3.1.1 Centralized

Fig. 3.1 shows the general representation of a centralized dual-user haptic system over a network. In the centralized (client-server) architectures the entire haptic simulation runs on a single server communicating with individual user workstations. Data (positions, velocities, etc.) from all workstations (clients) are sent to the central workstation (server) where the interaction forces are calculated and along with the objects' and users' states are sent to clients over the network. Fig. 3.2 displays the single-axis model of the centralized network-based control architecture proposed by Fotoohi *et al.* [31]. In this model, dynamics of the user and the haptic device are modeled by point masses m_1^h and m_2^h representing the combined masses of two users and haptic devices; m_o denotes the mass of the virtual object; k's and b's are stiffness and damping of corresponding virtual couplers; x's and $\bar{x}$'s are local and network transmitted positions; f_2^o and $\bar{f}_2^o$ are the calculated and transmitted forces applied to the remote user; and f_1^h and f_2^h are users' exogenous force inputs.

3.1.2 Distributed

Fig. 3.3 shows the general representation of a distributed dual-user haptic system over a network. Distributed architectures (peer-peer) involve multiple copies of the virtual environment simulation running on the user workstations. The position and velocity of each user along with those of the shared virtual object are sent over the channel to the other workstation. The two copies of the virtual object are connected through spring-damper type couplings to maintain synchronization of the objects. Fig. 3.4 displays the single-axis model of the distributed network-based control architecture proposed by Fotoohi *et al.* [31]. In this model, dynamics of the user and the haptic device are modeled by point masses m_1^h and m_2^h representing the combined masses of two users and haptic devices; m_{o1} and m_{o2}

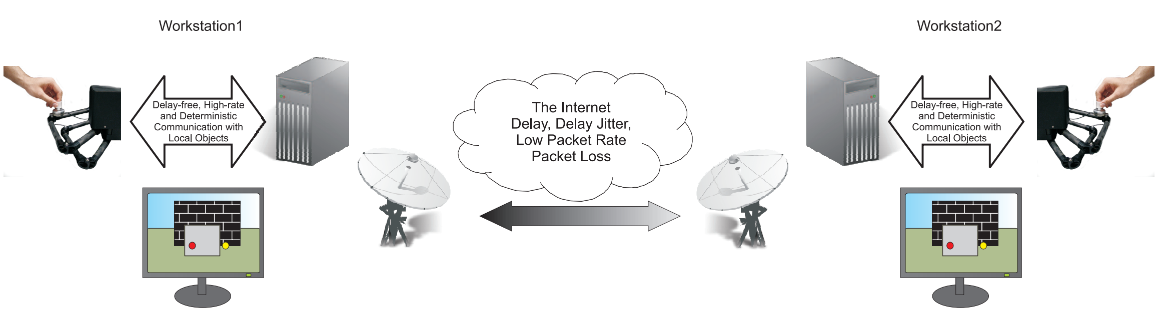

Figure 3.3: General representation of a distributed dual-user haptic system over a packet switched network.

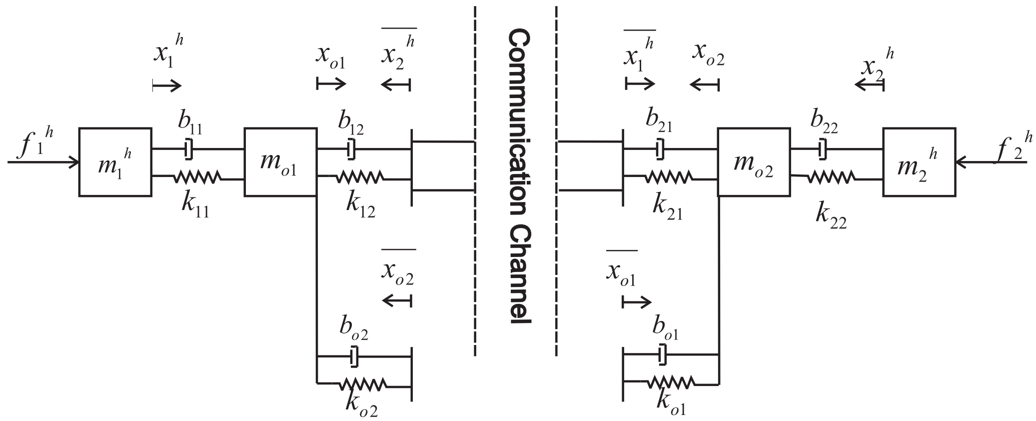

Figure 3.4: Model of distributed single-axis cooperative haptics.

denote the masses of the local copies of the virtual object; k's and b's are stiffness and damping of corresponding virtual couplers; x's and $\bar{x}$'s are local and network transmitted positions; and f_1^h and f_2^h are users' exogenous force inputs.

3.2 A Comparison Between Centralized and Distributed Architectures

The main advantage of a centralized architecture is its simplicity as it only involves one copy of the virtual environment and requires no synchronization. The performance of centralized architectures, however, can largely degrade due to the constraints of the network communication imposed on the data exchange between the clients and the server. Distributed architectures, in contrast, permit users to interact with their local copy of the virtual environment at a high control rate with negligible delay, thereby diminishing the impact of the network limitations on local interactions. The main challenge in such architectures is to maintain the synchronization among copies of the virtual environment.

It has been shown by Fotoohi *et al.* [31] that in contact with a virtual wall the maximum achievable wall stiffness in the centralized architecture is lower than that of the distributed architecture. This is in part due to the limited network packet transmission rate. In other words, in the centralized architecture, a low-rate communication channel connects the remote user to the central virtual environment. The downsampling of forces and positions of the remote workstation adversely affects the maximum achievable stiffness in contact with a virtual wall. The distributed architecture ameliorates this negative effect by allowing local high rate rendering of static objects at the expense of imposing higher computational power requirements and a need for a synchronization method.

A thorough comparison of the stability and performance of distributed and centralized architectures were presented by Fotoohi *et al.* [31]. In the rest of this work the distributed architecture is selected as a base due to its superiority over the centralized architecture.

3.3 Modeling

Fig. 3.5 displays a block diagram representation of the two-user distributed architecture introduced in Section 3.1.2. It is composed of continuous-time and multi-rate discrete-time blocks. User and haptic device have continuous-time dynamics, whereas virtual environment, corresponding couplers and communication link are implemented in a computer and are therefore discrete-time elements. The maximum packet transmission rate over the network is usually well below what is required for haptic rendering. Distributed architecture solves this problem by allowing virtual environment simulations to run at a higher sampling rate (T_c) compared to the network transmission rate (T_t). This approach results in a multi-rate system in which local high rate feedback loops are not affected by the relatively slow communication rate.

The method of subsystem resampling was used by Fotoohi *et al.* [31] to derive the

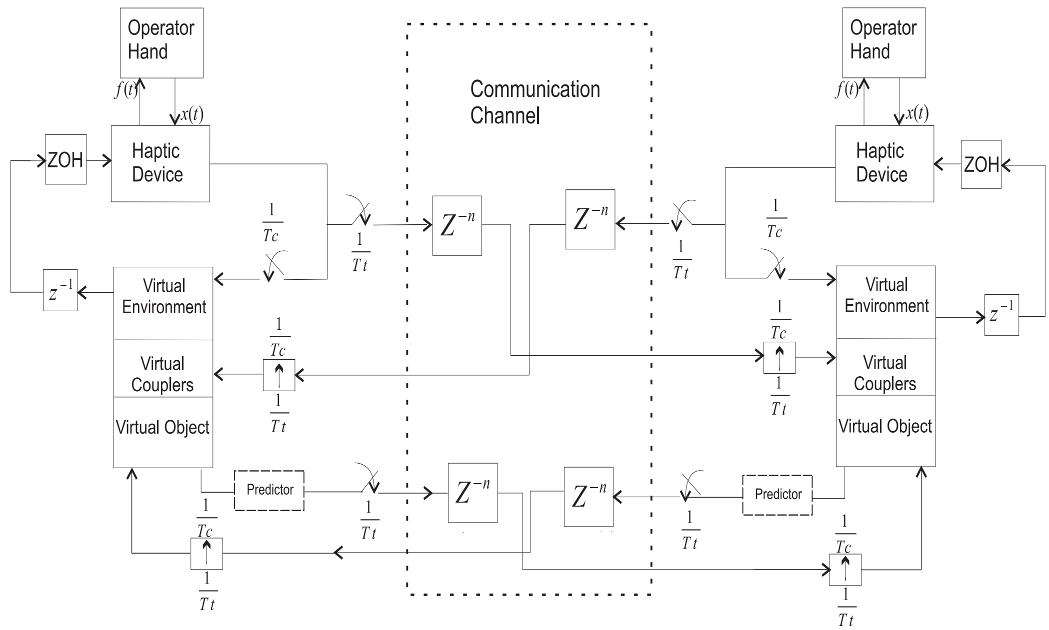

Figure 3.5: Distributed control architecture; state predictors, shown by dashed blocks, are added to improve performance.

dynamic equations of the multi-rate system of Fig. 3.5. In this method modeling and analysis are conducted in discrete-time. The method of subsystem resampling in modeling multi-rate discrete-time systems is founded on two assumptions. First, the two samplers are synchronized and second, the slower sampling rate is an integer multiple of the faster sampling rate, $T_{slow} = NT_{fast}$. The zero-order-hold blocks in Fig. 3.5 generate continuous-time signals by holding the sample value constant over the sampling interval (T_c). For this reason the continuous dynamics of the hand and haptic device are first discretized using a zero-order-hold (ZOH) continuous to discrete transformation. The system dynamics are divided into two subsystems on the basis of their sampling rate. These two sampling rates correspond to packet rate and control rate, and are denoted by T_t and T_c, respectively. State space difference equations of each subsystem are constructed at its sampling rate.

$$
\begin{aligned}
x_i[k_i+1] &= A_i x_i[k_i] + B_i u_i[k_i] \\
y_i[k_i] &= C_i x_i[k_i] + D_i u_i[k_i], \quad i = t, c
\end{aligned} \tag{3.1}
$$

At this stage the network/computation delay can be integrated into the system. In order to connect the two subsystems, they must operate at the same sampling rate. Therefore, the faster subsystem is downsampled to the slower sampling rate. The system matrices for the downsampled fast system are given by Fotoohi *et al.* [31]:

$$
\begin{aligned}
\widetilde{A}_t &= A_c^N \\
\widetilde{B}_t &= A_c^{N-1} B_c + A_c^{N-2} B_c + \cdots + A_c B_c + B_c \\
\widetilde{C}_t &= C_c \quad \widetilde{D}_t = D_c
\end{aligned} \tag{3.2}
$$

The two subsystems are then combined to form the overall dynamic equations of the system. The steps for modeling the distributed control architecture are shown in Fig. 3.6. The state-space matrices of each subsystem are presented in Appendix A.

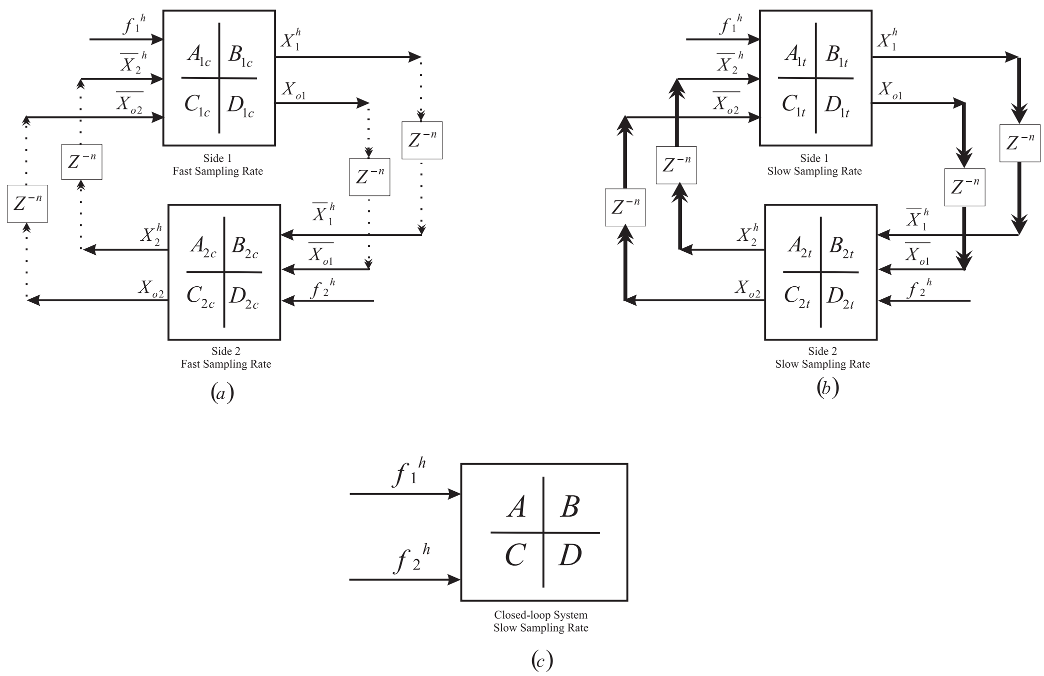

Figure 3.6: Subsystem resampling approach in modeling the distributed control architecture. (a) Constructing the state-space equations of each side at the fast sampling rate (T_c). (b) Resampling at the slow sampling rate (T_t). (c) Connecting the two subsystems and finding the closed-loop system dynamics. [35]

Chapter 4

Performance Measures and Parameter Optimization

4.1 Performance Measures

Fotoohi *et al.* [31] compared the perceived admittance by the users in centralized and distributed architecture to that of the object being simulated, in this case a pure mass in interaction with spring-damper. Obviously, in an ideal case the users should feel the combined dynamics of the virtual object and the environment with no distortion. To obtain the perceived admittance, given the linearity of the model, it can be assumed that only one user is in contact with the virtual object. The single-axis model of the resulting distributed architecture along with the ideal system are shown in Fig. 4.1(a) and (b), respectively. Here, k_w represents the stiffness and b_w denotes the damping of the virtual environment. The perceived admittance of the object, h_1, is defined as the ratio of the user hand velocity v_1^h to

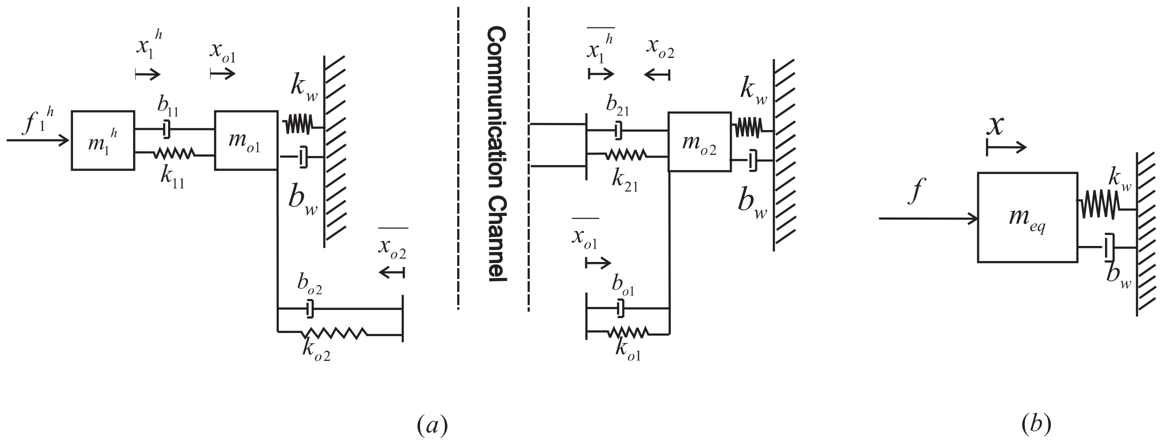

Figure 4.1: (a) Single user in distributed architecture; (b) The ideal system.

the user's input force f_1^h in Fig. 4.1(a).

$$h_1(j\omega) = \frac{v_1^h}{f_1^h}(j\omega) \tag{4.1}$$

Similarly, the ideal admittance , h_{ideal}, is defined as the ratio of the ideal velocity v to the input force f in Fig. 4.1(b).

$$h_{ideal}(j\omega) = \frac{v}{f}(j\omega) \tag{4.2}$$

In free motion k_w and b_w are set to zero. In contact, k_w can be assigned a very large value to simulate contact with a rigid wall. The subsystem resampling approach can be used to obtain the difference equations governing the dynamics of the system. The frequency responses of the system for a typical set of control coupling gains are displayed in Fig. 4.2. Table 4.1 contains the parameters used in obtaining these results.

Although the user's perceived admittance of the environment is an objective measure of performance, it is not sufficient in evaluating the effectiveness of the system. In the

$k_{11} = k_{22} = 3000$ N/m	$b_{11} = b_{22} = 20$ N·s/m
$k_{o1} = k_{o2} = 300$ N/m	$b_{o1} = b_{o2} = 10$ N·s/m
$k_{21} = 400$ N/m	$b_{21} = 2$ N·s/m
$m_1^h = m_2^h = 0.1$ kg	$m_o = 0.4$ kg
$T_c = 1/1000$ s	$N = 8$
$n = 1$	$m_{eq} = m_1^h + m_o = 0.5$ kg

Table 4.1: A set of system parameters used to produce the results in Fig. 4.2.

distributed control architecture the multiple (e.g. two) copies of the virtual object must be synchronized in order to ensure consistency among the local copies of the virtual environment. Therefore, another performance index should be defined that reflects the ability of the system to synchronize these virtual objects. Position tracking of the two copies of the virtual object can provide a measure for evaluating the synchronization ability of the distributed architecture. For this reason, g_i is defined as the ratio of the position of the virtual objects x_{oi} to the user's input force f_1^h in Fig. 4.1(a).

$$g_i\,(j\omega) = \frac{x_{oi}}{f_1^h}\,(j\omega) \tag{4.3}$$

Similarly, the ideal position response is defined as the ratio of the ideal position x to the user's input force f in Fig. 4.1(b).

$$g_{ideal}\,(j\omega) = \frac{x}{f}\,(j\omega) \tag{4.4}$$

Fig. 4.3 displays the frequency responses of g_i and g_{ideal} in free motion with the parameters in Table 4.1.

In the distributed architecture user's position is sent across the channel, where through

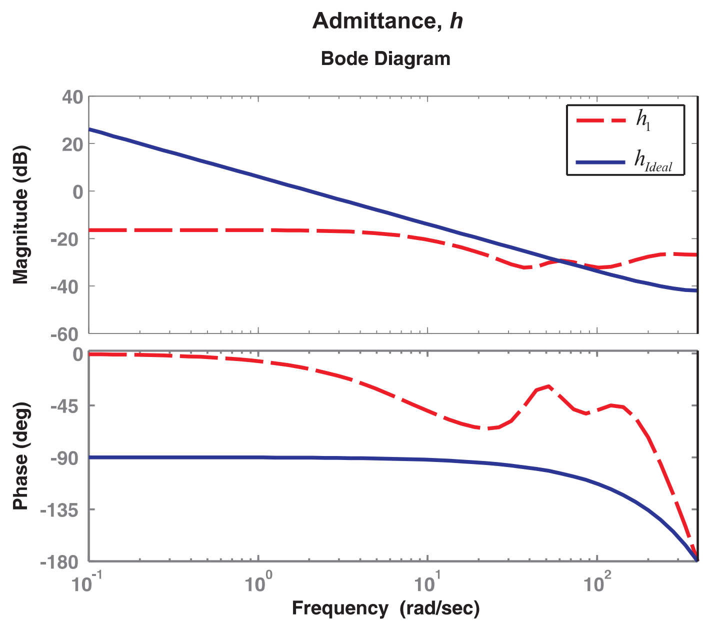

Figure 4.2: User's perceived admittance in distributed architecture $h_1(j\omega)$ and ideal admittance $h_{Ideal}(j\omega)$ in free motion

virtual couplings the appropriate force is calculated and applied to the object. These couplings reduce the discrepancy caused by the user's force. The other factor in evaluating the performance is the ability of the system to synchronize the virtual objects in the presence of a disturbance a model of which is shown in Fig. 4.4. In this figure, f_d is the disturbance on m_{o1}. d_i is defined as the ratio of the position of the virtual objects x_{oi} to the disturbance f_d.

$$d_i(j\omega) = \frac{x_{oi}}{f_d}(j\omega) \tag{4.5}$$

Fig. 4.5 displays the frequency response of $d_i(j\omega)$ in free motion.

The virtual objects position along with the perceived admittance of the environment and the ability of the system to synchronize the objects in the presence of disturbance provide a complete measure for evaluating the transparency of the distributed architecture and will

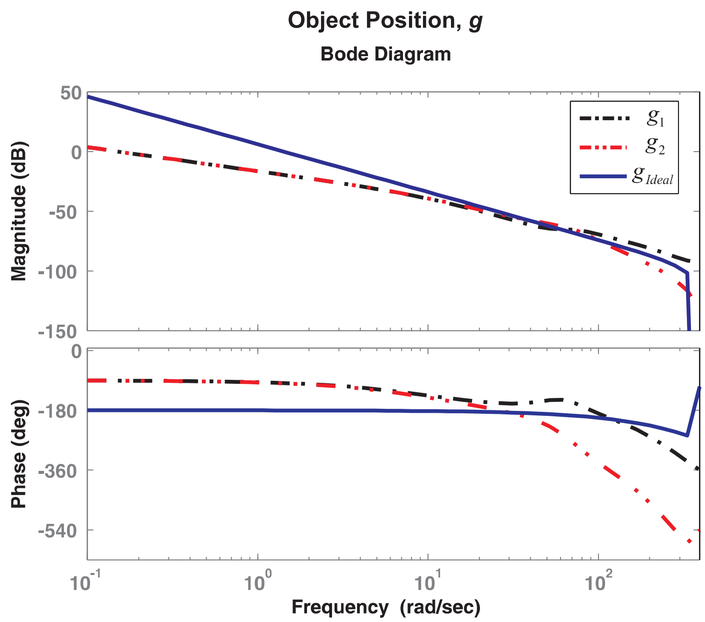

Figure 4.3: Object position $g_i\,(j\omega)$ and ideal position $g_{Ideal}\,(j\omega)$ in free motion

be used in the next section to formulate an optimization problem.

4.2 Parameter Optimization

The values of virtual coupling parameters, k_{11}, b_{11}, k_{o2}, b_{o2}, k_{12}, b_{12}, in the distributed control architecture can significantly affect the performance and stability of the system. While these parameters have been chosen empirically by Sankaranarayanan and Hannaford [76], and Fotoohi *et al.* [31], there is no guarantee that such selection would yield the best achievable response. In this section, an optimization problem will be formulated for choosing the best set of virtual coupling gains in terms of the response transparency and system stability. This provides an objective and clear mechanism for the control synthesis in the distributed network-based haptics.

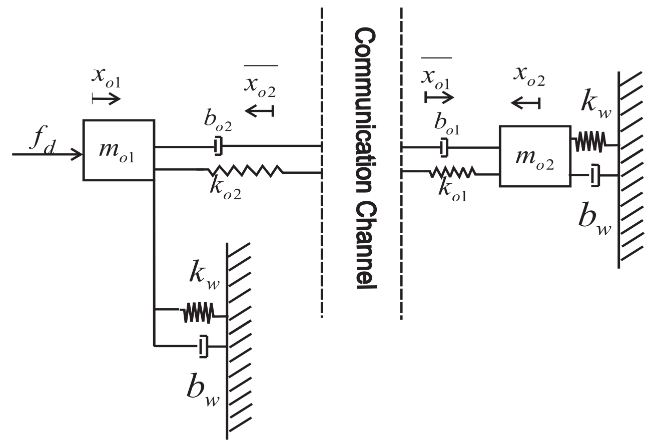

Figure 4.4: Effect of disturbance on the virtual object's position.

The frequency responses of admittance and virtual objects' position defined in Section 4.1 were used as a graphical tool for evaluating the performance in a distributed architecture. Equations (4.6), (4.7), and (4.8) convert the admittance error and position discrepancy due to user's force and disturbance into quantitative measures suitable for the use in a cost function.

$$f_1 = \int_0^{\pi} |h_1(j\omega) - h_{Ideal}(j\omega)|\, w_1(\omega)\, d\omega. \tag{4.6}$$

$$f_2 = \int_0^{\pi} |g_1(j\omega) - g_2(j\omega)|\, w_2(\omega)\, d\omega. \tag{4.7}$$

$$f_3 = \int_0^{\pi} |d_1(j\omega) - d_2(j\omega)|\, w_3(\omega)\, d\omega. \tag{4.8}$$

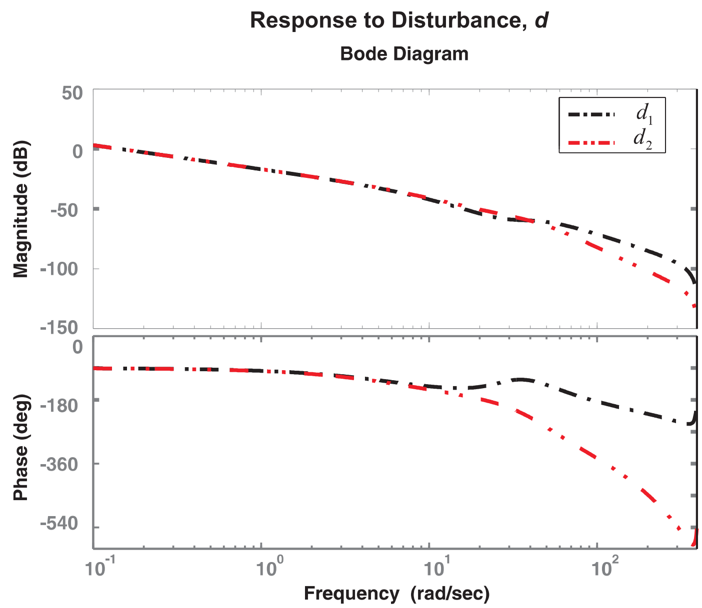

Figure 4.5: Response to disturbance $d_i\,(j\omega)$ in free motion

where f_1 represents the admittance error, f_2 is the virtual objects discrepancy and f_3 represents the discrepancy in response to disturbance, h_i, g_i and d_i have been defined in Section 4.1, and w_i's are normalizing factors chosen as:

$$
\begin{aligned}
w_1(\omega) &= |h_{Ideal}\,(j\omega)|^{-1}\,l(\omega) \\
w_2(\omega) = w_3(\omega) &= |g_{Ideal}\,(j\omega)|^{-1}\,l(\omega)
\end{aligned} \tag{4.9}
$$

where $l(\omega)$ is a frequency dependant gain used to give relevant importance to different frequencies.

The goal is to find the virtual coupling gains, $y = [k_{11}, b_{11}, k_{o2}, b_{o2}, k_{12}, b_{12}]$, that minimize the objective functions defined in equations (4.6)-(4.8). The system is assumed to be symmetric, i.e. $y = [k_{11},\ b_{11},\ k_{o2},\ b_{o2},\ k_{12},\ b_{12}] = [k_{22},\ b_{22},\ k_{o1},\ b_{o1},\ k_{21},\ b_{21}]$. The resulting multi-objective optimization problem can be formulated using the following methods.

A. Weighted Sum Method

$$\begin{array}{ll} \text{minimize} & f(y) = \sum_i \alpha_i f_i(y) \ , \quad i \in \Omega = \{1,2,3\ldots,k\} \\ \text{subject to} & \text{stability} \end{array}$$

where $f(y)$ is the weighted sum of objectives, f_i's. α_i's are weighting factors assigned to the corresponding objective functions. This method converts the multi-objective problem into a single-objective problem suitable for standard optimization algorithms. The solution should avoid gains that make the system unstable. Hence, stability is imposed as a constraint on the optimization problem. A drawback of this method is the need to find appropriate weighting factors, α_i, for each objective function.

B. Goal Programming Method

$$\begin{array}{ll} \text{minimize} & f(y) = \sum_i \alpha_i f_i(y) \ , \quad i \in \Omega_1 \\ \text{subject to} & \alpha_j f_j(y) < \delta_j \ \ \& \ \ \text{stability} \ , \quad j \in \Omega_2 \\ \multicolumn{2}{l}{\Omega_1 \cup \Omega_2 = \Omega = \{1,2,3\ldots,k\} \ \ \& \ \ \Omega_1 \cap \Omega_2 = \varnothing} \end{array}$$

In the goal programming method a set of objective functions, f_j, are selected and a target value is assigned to each. In order to satisfy the constraints, the selected objective functions must be less than the assigned target values. This method attempts to minimize a weighted sum of the remaining objective functions, f_i. The goal programming method simplifies the problem by constraining some of the objective functions and limiting the search region. However, care must be taken in selecting the appropriate target values to avoid over-constraining the problem (Gass [33]).

Chapter 5

Predictive Synchronization and Feedforward Scheme

In this chapter a discrete-time state predictor is employed to improve the stability and transparency of the haptic simulation. In addition a feedforward control scheme is proposed to improve coherency between copies of the virtual object.

5.1 Predictive Synchronization

The effect of communication delay on transparency and stability of haptic interactions were investigated by Alhalabi and Horiguchi [5], and Wang *et al.* [83]. In the distributed architecture, time delay causes inconsistency between the copies of the virtual object. A smith predictor was used by Cheong *et al.* [22] to compensate for the negative effects of delay on object synchronization in such environments. A discrete-time model-based predictor is proposed here to reduce inconsistency between the copies of the virtual objects. It is anticipated that the state predictor will improve the performance and stability of the haptic interaction by estimating the virtual objects' states ahead of time. At each site, the predictor provides an estimate of the future values of the local virtual object position and velocity over the prediction horizon. These values are then transmitted to the other side of the communication channel and are used in virtual couplers (k_{o1}, b_{o2}) to produce the synchronizing control action.

In order to construct a discrete-time state predictor we must first derive the state space equations of the virtual object. In this work, the virtual object is a box and is modeled as a mass-damper system as presented in Chapter 3. However, the approach is identical for designing a predictor for virtual object with different dynamics.

The state space dynamics of a mass-damper system are given by:

$$\begin{aligned} \dot{X}_o(t) &= A_o X_o(t) + B_o u_o(t) \\ y_o(t) &= C_o X_o(t) + D_o u_o(t) \end{aligned} \tag{5.1}$$

where

$$X_o = \begin{bmatrix} x_o \\ \dot{x}_o \end{bmatrix}, A_o = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{m_o} \end{bmatrix}, B_o = \begin{bmatrix} 0 \\ \frac{1}{m_o} \end{bmatrix} \tag{5.2}$$
$$C_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, u_o = [f_o]$$

In the above equations the state vector, x_o, is the vector of object position and velocity; f_o, the net force applied to the box, is the input and output is the state vector itself.

Discrete-time state-space dynamics of rigid single-body virtual objects can be derived using the ZOH continuous-to-discrete transformation in the following form:

$$\begin{aligned} x_{oD}[k+1] &= A_{oD}x_{oD}[k] + B_{oD}f_{oD}[k] \\ y_{oD}[k] &= C_{oD}x_{oD}[k] + D_{oD}u_{oD}[k] \end{aligned} \tag{5.3}$$

where

$$\begin{gathered} A_{oD} = e^{A_o T} \quad B_{oD} = \left(\int_0^T e^{A_o \eta} d\eta\right) B_o \\ C_{oD} = C_o \quad D_{oD} = D_o \end{gathered} \tag{5.4}$$

Assuming that the net force is constant over the prediction horizon, a prediction of the state vector over a time horizon of n sample times can be obtained by

$$\widetilde{x}_{oD}[k+n] = A_{oD}^{n} x_{oD}[k] + \left(A_{oD}^{n-1}B_{oD} + A_{oD}^{n-2}B_{oD} + \cdots + A_{oD}B_{oD} + B_{oD}\right) f_{oD}[k] \tag{5.5}$$

Although the net force on the object will not be constant over the prediction horizon, the assumption can provide reasonable results when the net force does not change abruptly over the prediction horizon. A block diagram representation of the predictor is shown in Fig. 5.1.

Fig. 5.2 displays the proposed predictive scheme for shared haptic interaction using a distributed control architecture. This is a modification of the control architecture in Fig. 3.4 where instead of the actual object states, x_{oi}, the predicted states, $\widetilde{x}_{oi}$, are calculated and transmitted over the channel. The measures defined in Section 4.2 can be similarly employed to evaluate and optimize the performance of the system in this case.

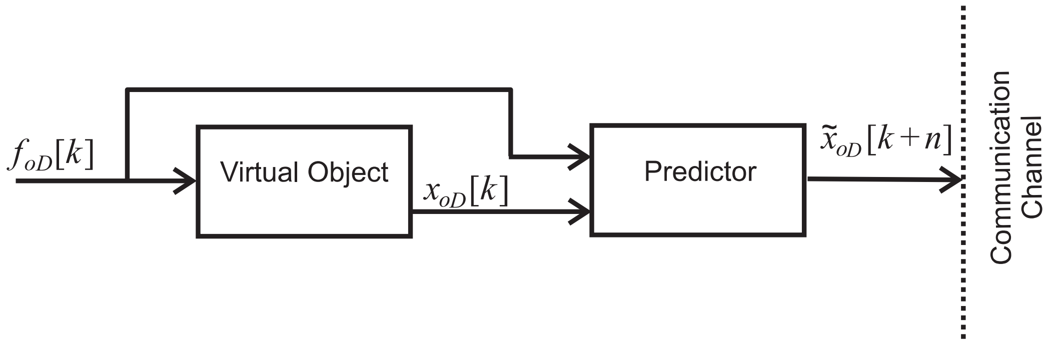

Figure 5.1: Predictor- Block diagram representation.

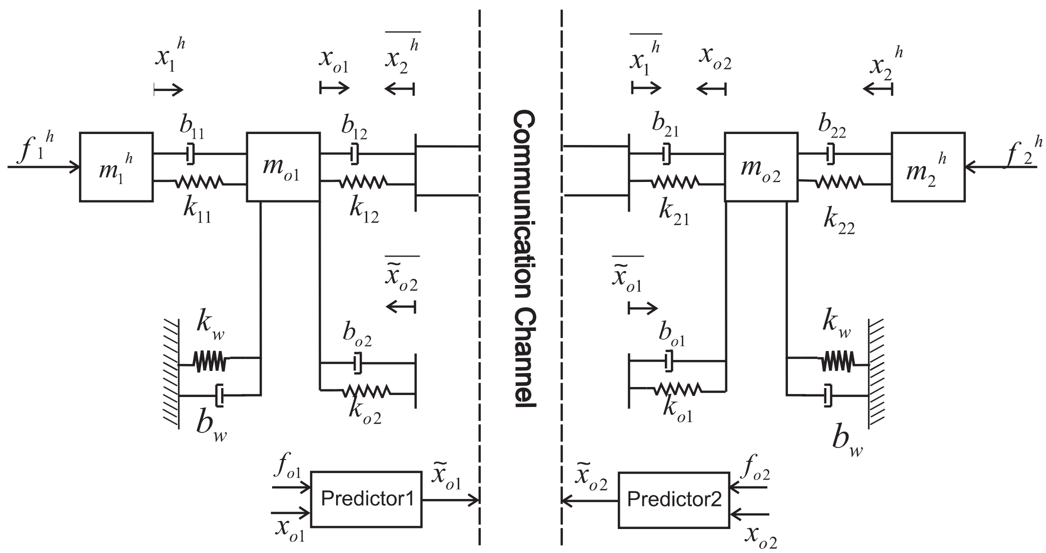

Figure 5.2: Predictive scheme.

5.2 Feedforward Predictive Scheme

Fig. 5.3(a) shows a single axis model of the distributed scheme with predictive synchronization. The dashed block indicates virtual object 2, m_{o2}, and the force exerted to it by user 1, m_1^h. f_{21} denotes this force which can be computed from one of the following schemes.

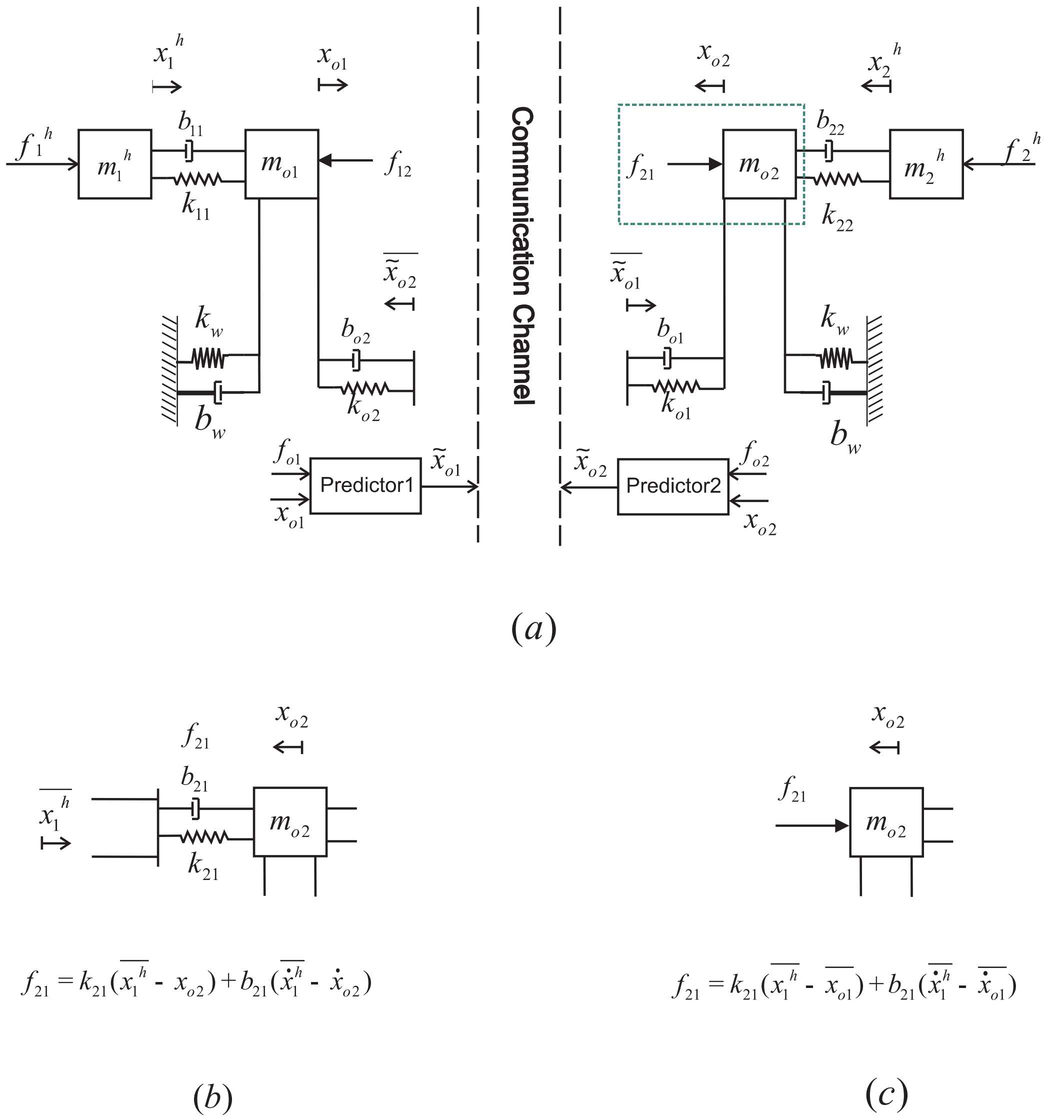

Figure 5.3: (a) The distributed architecture with state prediction. The dashed box displays the force applied to virtual object at side 2 from user at side 1. (b) Feedback scheme. (c) Feedforward scheme.

5.2.1 Feedback Scheme

Fig. 5.3(b) displays the force calculated by the feedback scheme. the spring-damper couplings k_{21} and b_{21} provide a coupling between the user at the remote side and the local copy of the object. The force, f_{21}, is directly dependant on the virtual object's state and is determined by the following equation:

$$f_{21} = f_{feedback} = k_{21}(\overline{x_1^h} - x_{o2}) + b_{21}(\overline{\dot{x}_1^h} - \dot{x}_{o2}) \tag{5.6}$$

The advantage of this method of force generation is that the feedback connection tends to reduce the error between haptic device's position, x_1^h, and virtual object's position, x_{o2}. The force is generated by a PD controller, spring-damper, between the virtual object's position and the haptic device position transmitted from side 1. Network communication delay imposes a restriction on how large the coupling can be chosen.

In this scheme copies of the virtual object may or may not be in contact with each user. Whether or not a virtual object is in contact with a user depends on the position of the virtual object and the user and is determined by the collision detection module. As shown in Fig. 5.4, small discrepancy between the copies of the virtual objects may lead to one object being in contact with a user while no contact has been detected between the user and the other copy of the object. Therefore, the error between the copies of the virtual object and nonlinearity of contact may result in a significant difference between the forces applied to the objects.

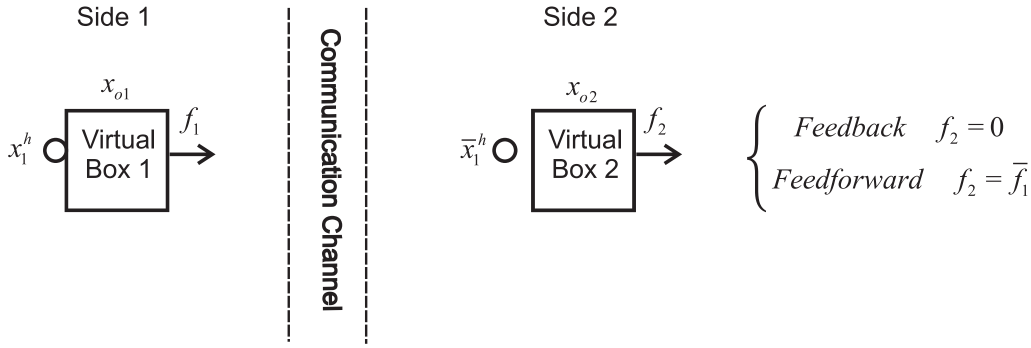

Figure 5.4: The force applied to virtual object 2 in feedback and feedforward methods as user 1 loses contact with one object due to discrepancy between the copies.

5.2.2 Feedforward Scheme

Fig. 5.3(c) displays the feedforward scheme. f_{21} denotes the force on a copy of the virtual object caused by the remote user and is calculated by:

$$f_{21} = f_{feedforward} = k_{21}(\overline{x_1^h} - \overline{x_{o1}}) + b_{21}(\overline{\dot{x}_1^h} - \overline{\dot{x}_{o1}}) \tag{5.7}$$

In this method the force caused by the local interaction of each user with its own copy of the virtual object is transmitted over the network and applied to all other copies of the virtual objects. Forces generated by this method are not directly dependant on the position of the virtual object on which the force is applied. Upon detection of contact between a user and its own local copy of the virtual object the local interaction forces are calculated and transmitted over the network. This force is then applied to all other copies of the virtual object, thereby, diminishing the need for a collision detection module between remote users and local copies of the virtual object. In addition discrepancy between copies of the virtual object will not result in a difference between the forces applied to them as shown in Fig. 5.4.

5.2.3 Feedforward/feedback Scheme

A feedforward/feedback scheme combines the two methods. The force applied to the remote virtual object is calculated by:

$$f_{21} = k_{feedforward} f_{feedforward} + k_{feedback} f_{feedback} \tag{5.8}$$

With a proper selection of the gains, $k_{feedforward}$ and $k_{feedback}$, it is possible to take advantage of both the feedforward and the feedback schemes. On one hand the feedback scheme reduces the error between each copy of the virtual object and remote users through the spring-damper coupling. On the other hand the feedforward scheme reduces the undesired effects caused by nonlinearity of contact. These gains can be added as an optimization parameter to the optimization problem formulated in Chapter 4 or can be chosen empirically. In this work we have chosen $k_{feedback} = 1$, and $k_{feedforward} = 0.5$.

Chapter 6

Design Examples

An example on designing a two-user haptic platform for a given communication delay is presented in this chapter. The virtual coupling gains in the distributed architecture of Fig. 3.4 and the predictive and feedforward predictive schemes of Chapter 5 are optimized for a round trip delay of 176ms ($n = 11$). This is approximately the delay encountered in our experiments between Mcmaster University and Orange County, CA, as will be seen in the next chapter. Furthermore, an analytical comparison between the optimized schemes with respect to the objective functions defined in Chapter 4 is provided here.

The goal programming method described in Section 4.2 is used to formulate the problem. The system parameters used in our analysis have been defined in Chapter 3 and Chapter 5. These parameters are presented in Table 6.1.

$m_1^h = m_2^h = 0.1$ kg	$m_o = 0.4$ kg
$T_c = 1/1000$ s	$N = 8$
$n = 11$	$m_{eq} = m_1^h + m_o = 0.5$ kg
$k_{feedback} = 1$	$k_{feedforward} = 0.5$

Table 6.1: Set of system parameters used in the optimization problem.

A. Objective Function: The goal programming method simplifies the objective function by constraining some of the objectives to be less than a target value. In our case, position discrepancy factors, $f_2(y)$ and $f_3(y)$, are kept in the objective function, whereas, the admittance measure, $f_1(y)$, is moved to the constraint. Although other configurations are possible, this selection is made because the position discrepancy factors and admittance measure are somewhat opposing objectives. Therefore, it is reasonable to constraint one objective and optimize the other one.

The solution of the optimization problem must reduce discrepancy between the two copies of the virtual object in free motion and in contact. The cases of free motion and contact are described as follows:

1. Free motion: This is represented by $k_w = 0$ N/m and $b_w = 1$ N.s/m. The small non-zero environment damping is employed to eventually stop the objects when the users are no longer in contact.

2. Contact: Users should be able to bring the virtual objects into contact with a static object such as a wall in the virtual environment. This case is modeled by setting the environment stiffness to $k_w = 1000$ N/m; the environment damping is chosen as $b_w = 10$ N.s/m.

The final objective function is the sum of the objective functions described above.

$$f(y) = f_{2free}(y) + f_{3free}(y) + f_{2contact}(y) + f_{3contact}(y) \tag{6.1}$$

where f_2 and f_3 correspond to position discrepancy between the copies of the virtual object due to user's force and disturbance, respectively, as defined in Chapter 4. The subscripts, $free$ and $contact$, respectively represent the cases of free motion and contact as defined above.

B. Constraints: The perceived admittance measure, $f_1(y)$, which was defined in Chapter 4, is included as a constraint in the optimization problem.

$$f_{1free}(y) < 0.6 \ \& \ f_{1contact}(y) < 0.5$$

Stability is the other requirement of the design and is included in the constraints. In order to have a realistic stability analysis the following cases are considered.

1. Users in contact: The system has to be stable in free motion ($k_w = 0$ and $b_w = 1$ N.s/m) and in contact with a wall ($k_w = 1000$ N/m and $b_w = 10$ N.s/m).

2. Users not in contact: The system has to be stable when users are not in contact with the virtual object. This condition is derived by setting the user and object virtual couplings in Fig. 3.4 and Fig. 5.2 to zero, i.e. $k_{11} = k_{22} = k_{12} = k_{21} = 0$ N/m, $b_{11} = b_{22} = b_{12} = b_{21} = 0$ N.s/m. The remaining virtual couplings are those between the two objects, i.e. $k_{o1}, b_{o1}, k_{o2}, b_{o2}$. Therefore, this requirement restricts the acceptable region for the virtual coupling gains between the two objects.

The discrete-time models of the systems are derived using the multi-rate approach described in Chapter 3. Ideally the optimized system should provide stable interaction over a range of user's dynamics and a range of virtual environments from free motion to contact with a stiff virtual wall. However, the resulting robust stability problem becomes extremely difficult to solve due to the nonlinear dependence of the system matrices on the optimization parameters. The nonlinearity is caused by the subsystem resampling method after downsampling of the fast subsystems, as described in Chapter 3.

In order to have a stable system all the poles of the resulting discrete-time system (eigenvalues of the state matrix) must lie inside the unit circle.

$$max(|eig(A)|) < 1$$

The nonlinear dependance of these poles (eigenvalues) on the optimization parameters results in a nonlinear constraint in the optimization problem. In order to provide stability robustness to changes in parameters and to avoid marginally stable systems the condition can be restricted to having all poles (eigenvalues) in a smaller circle inside the unit circle.

$$max(|eig(A)|) < \delta$$

where δ is the radius of the circle.

MATLAB® Optimization Toolbox™ was employed to solve the nonlinear constrained optimization problem. Due to the nonlinear dependance of constraints and objective function on the design parameters it is difficult to investigate the convexity of the problem and the resulting solutions might be local minimums. Random search methods such as those described by Brooks [17] ,and Price [72] can be used to search for the global optimum and are not considered here. The resulting virtual coupling gains and corresponding objective functions are shown in Table 6.2 and Table 6.3, respectively. Fig. 6.1 shows the frequency response of the admittance perceived by the user in all schemes in free motion and in contact. The frequency response of the virtual objects' positions due to user's input force is shown in Fig. 6.2. Fig. 6.3 shows the frequency response of the virtual objects' positions in response to a disturbance.

The optimization problem attempts to minimize the position discrepancy between the copies of the virtual object in the three schemes while constraining the error in admittance

$n=11$	**Virtual Couplings**					
	$k_{11}=k_{22}$ N/m	$b_{11}=b_{22}$ N.s/m	$k_{o1}=k_{o2}$ N/m	$b_{o1}=b_{o2}$ N.s/m	$k_{21}=k_{12}$ N/m	$b_{21}=b_{12}$ N.s/m
No Prediction	2259	30	12.6	1.3	81.9	3.1
Prediction	2000	29.8	47.1	1.9	105.7	9.1
FF + Prediction	1999	24.1	35.5	0	9.8	1

Table 6.2: Virtual coupling gains for round trip delay of 176ms.

$n=11$	**Free Motion**			**Contact**		
	f_1	f_2	f_3	f_1	f_2	f_3
No Prediction	0.60	0.38	0.87	0.41	0.87	0.96
Prediction	0.59	0.20	0.21	0.43	0.77	0.90
FF + Prediction	0.41	0.20	0.37	0.46	0.57	0.91

Table 6.3: Objective functions for round trip delay of 176ms.

perceived by the user. Based on a comparison of the position discrepancy measures between the first two schemes, it is seen that the predictive scheme significantly improves the position tracking measure, f_2, and disturbance measure, f_3 in free motion. In addition, the predictive scheme allows higher virtual coupling gains between the virtual objects without sacrificing stability.

The addition of the feedforward control further improves the position tracking measure, f_2, in contact. It also improves the admittance rendered to the user, f_1, in free motion by reducing the amount of damping felt by the user as seen in Fig. 6.1. The performance measures are derived from the linear model of the haptic system. Therefore, the ability of the feedforward scheme in reducing object discrepancy due to nonlinearity of contact, as discussed in Chapter 5, is not reflected in these performance measures. The optimization results show that the addition of the prediction and feedforward mechanisms significantly improve the performance of the cooperative haptic simulation in this design example.

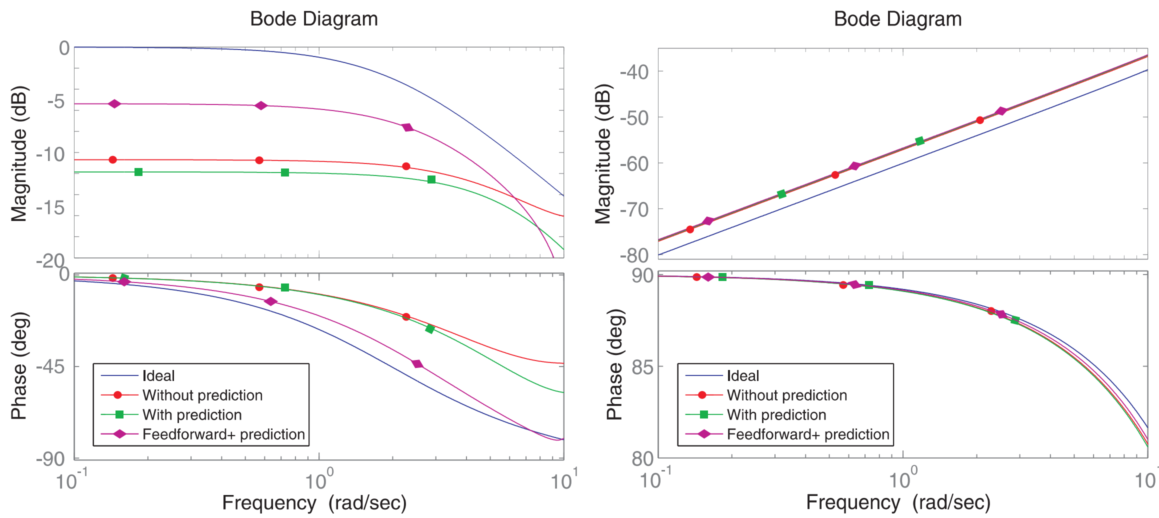

Figure 6.1: User's perceived admittance in distributed architecture $h_1(j\omega)$ and ideal admittance $h_{Ideal}(j\omega)$ in free motion (left figure) and contact (right figure).

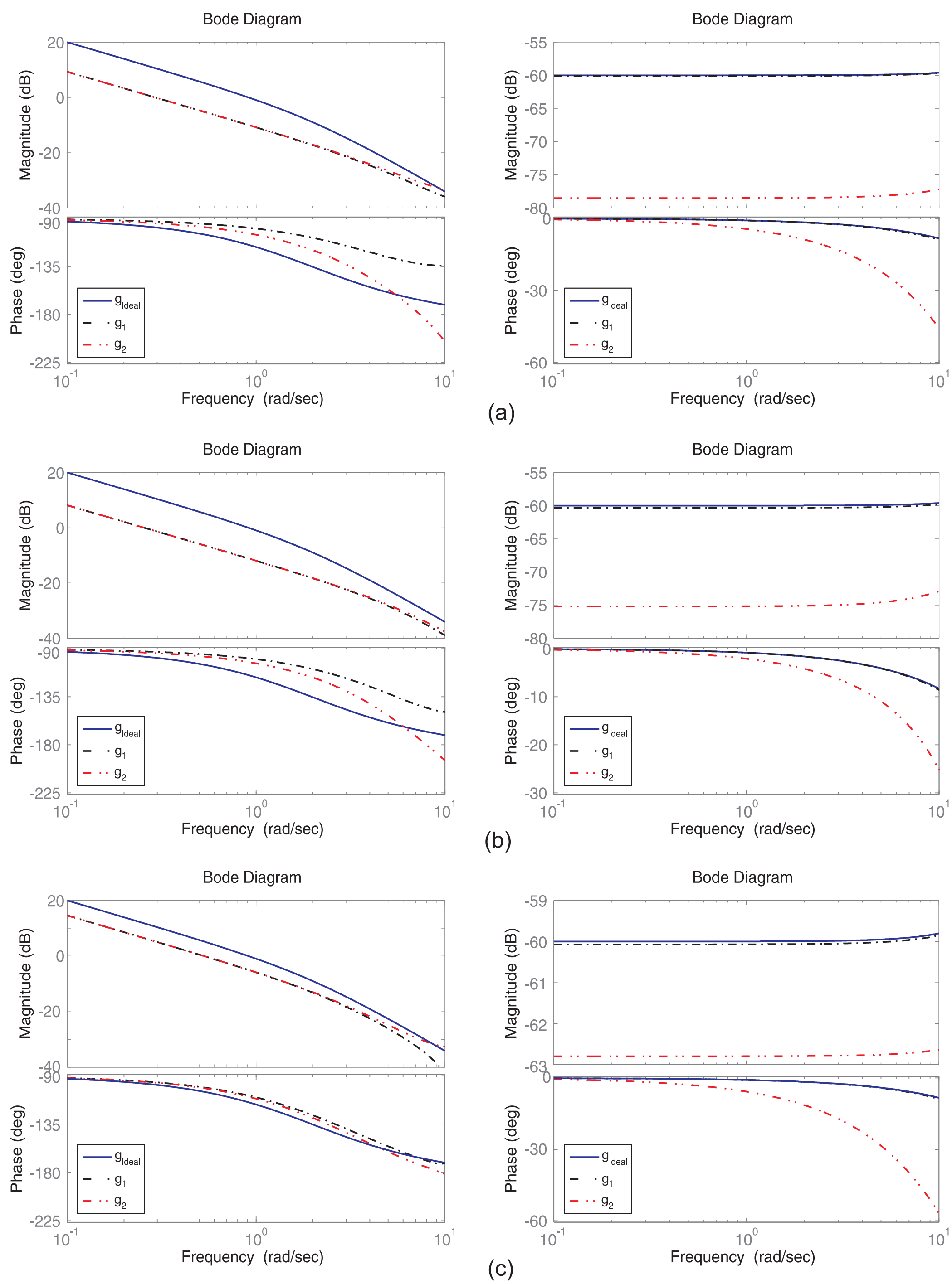

Figure 6.2: Object position $g_i(j\omega)$ and ideal position $g_{Ideal}(j\omega)$ in free motion (left column) and contact (right column). (a) Without prediction; (b) With prediction; (c) Feedforward + prediction.

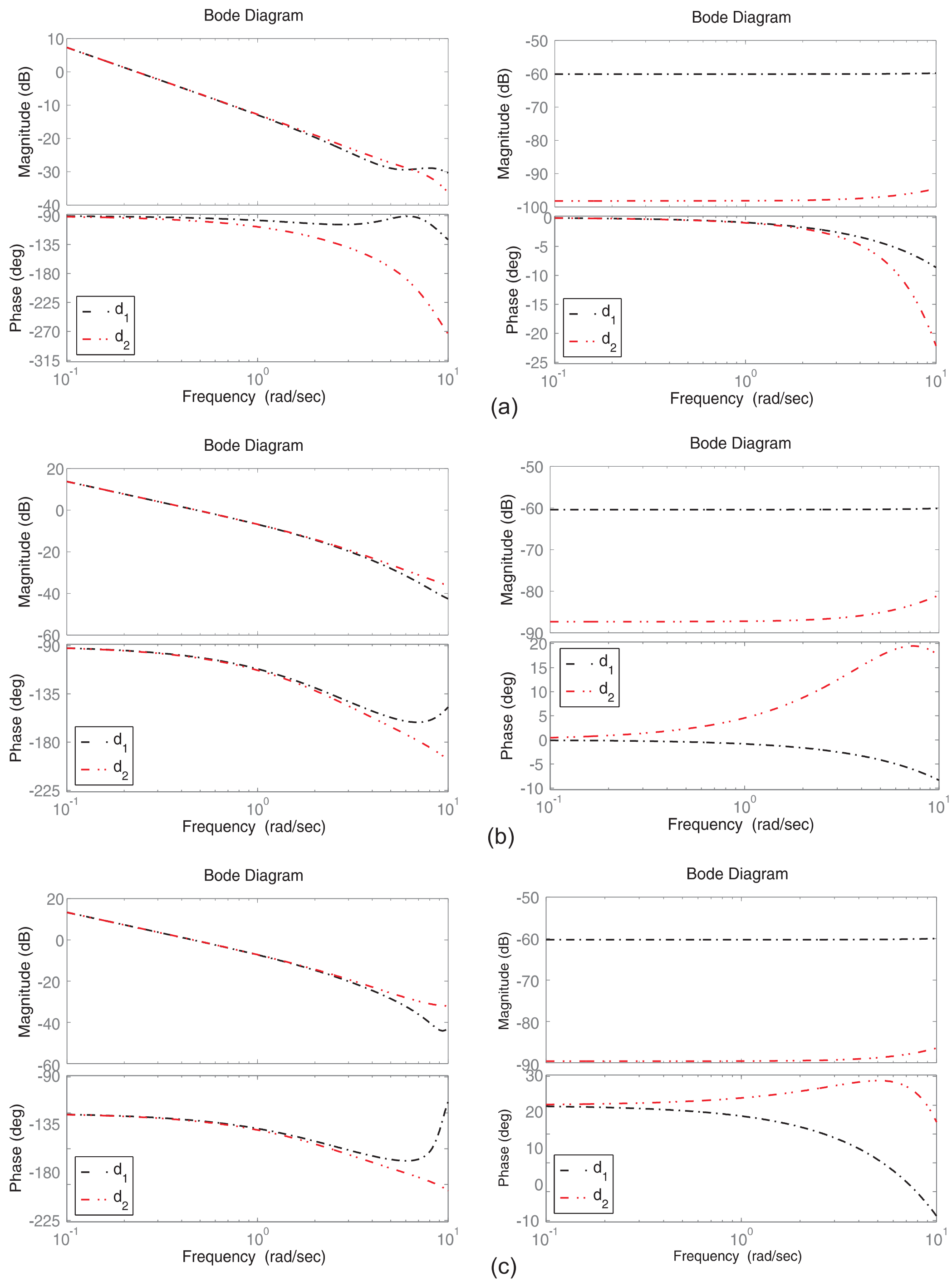

Figure 6.3: Response to disturbance $d_i\,(j\omega)$ in free motion (left column) and contact (right column). (a) Without prediction; (b) With prediction; (c) Feedforward + prediction.

Chapter 7

Experimental Setup and Results

A platform for two-user haptic interaction over a network is developed in this chapter. It is comprised of impedance-type haptic devices, virtual environment simulators for distributed haptic simulations, graphical display and network communication blocks. The platform provides an environment in which users from across a WAN can collaborate in a shared virtual environment with haptic feedback. The elements of this platform are described in this chapter.

The control schemes proposed in previous chapters are implemented in the two-user experimental setup to verify the effectiveness of our approach in practice. The results of the haptic experiments using this platform are also presented in this chapter.

7.1 Experimental Setup

Fig. 7.1 displays the building blocks of the haptic simulator at one side of the multi-user platform. Elements of this haptic simulator along with the network blocks used in our experiments are briefly described in this section.

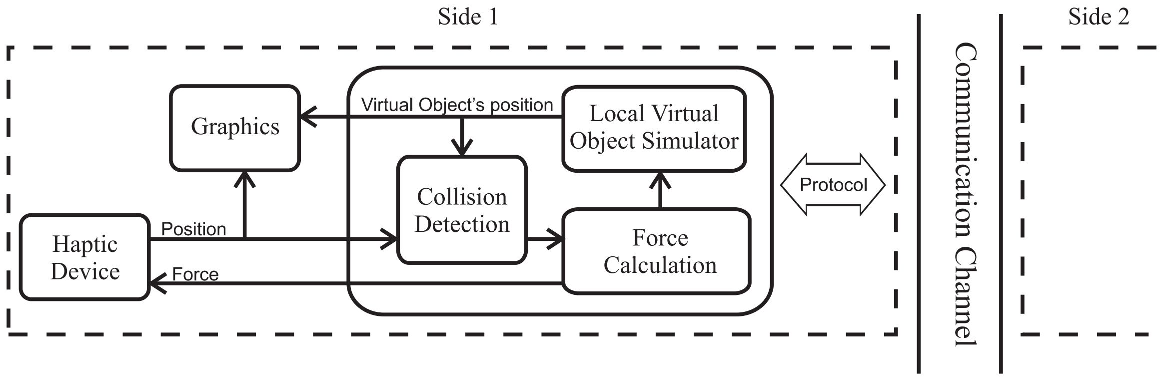

Figure 7.1: Building blocks of the multi-user distributed haptic simulator. The two sides of the communication channel are identical.

7.1.1 Haptic Devices

In the two-user experiments two planar two-degree-of-freedom Quanser pantographs provide haptic feedback to the users. Each pantograph is actuated by two DC motors. Two Quanser QPA linear current amplifiers power the motors. Motor shaft angles are measured by optical encoders with 20,000 counts per revolution. In order to block direct visual feedback from other users these haptic devices have been placed at different locations in our lab.

7.1.2 Virtual Environment Simulator

The Virtual Environment (VE) Simulator includes the dynamics of the virtual objects. It also performs rendering of forces to be applied to the users and virtual objects. The VE simulator updates the states of virtual objects based on the forces calculated. In the distributed architecture, the VE simulator is implemented locally on each workstation to provide high-rate haptic feedback to the users. Different modules of the VE simulator are briefly described here.

Collision Detection Module

The Collision Detection Module is responsible for detecting any contact between users and the virtual object. It also detects collision between the dynamic virtual objects, i.e. virtual box, and static virtual objects, i.e. virtual wall. Upon the detection of collision a flag is set and along with contact information, e.g. side and angle of contact, is sent to the force calculation module.

Force Calculation Module

The Force Calculation Module (FCM) calculates the forces to be applied to virtual objects and users. The FCM constantly monitors the contact information from the Collision Detection Module. When no contact has been detected the contact flag is not set and no force is calculated. Upon the detection of a contact between a user and a virtual object the FCM determines the forces to be applied to the virtual object and the user. According to Newton's third law the forces applied to the virtual object and the user are equal in magnitude and opposite in direction. A penalty based force generation method has been employed in this work. Forces are calculated based on the amount of penetration of the haptic device into the virtual object as shown in Fig. 2.3. Virtual spring-damper couplings convert this penetration to contact forces.

The maximum achievable stiffness in penalty based methods is limited due to stability constraints caused by sensor quantization, discrete-time implementation of the controller ,and limited computational update rate (Kuchenbecker *et al.* [53]. In order to provide a more realistic sense of rigid contact it has been suggested by Kuchenbecker *et al.* [54], Lawrence *et al.* [55] to apply large impulse forces to the haptic interface at the moment of contact for a short duration of time. The purpose of this force is to prevent the haptic device from penetrating deeper into the virtual object by rapidly reducing the velocity of the haptic device and therefore providing a sense of rigidness. From the conservation of momentum one can write

$$\int_0^T f(t)dt = m_h v_c \tag{7.1}$$

where m_h is the combined mass of user's hand and haptic device and v_c is the normal component of the contact velocity relative to the virtual object. In writing Eq. (7.1), it has been assumed that the user comes into contact with a rigid wall or a relatively large mass and that m_h is constant and known. Adaptive methods such as those proposed by Abdossalami and Sirouspour [2] eliminate the need for accurate knowledge of the user's hand and haptic device dynamics by replacing it with the dynamics of an adjustable mass-damper tool.

Assuming a constant force is applied for N_c sample times, Eq. 7.1 becomes

$$F_c T_c N_c = m_h v_c \Rightarrow F_c = \frac{m_h v_c}{T_c N_c} \tag{7.2}$$

Care must be taken in selecting N_c such that F_c does not exceed the maximum haptic device force. In our experiments $N_c = 2$.

Virtual Object Simulator Module

The Virtual Object Simulator Module is responsible for updating the states of dynamic virtual objects. The states are governed by the state space dynamics of the object and are updated based on the forces calculated in the Force Calculation Module and the current states of the object. Ode1 (Euler) has been used as the integration routine in Matlab/Simulink to

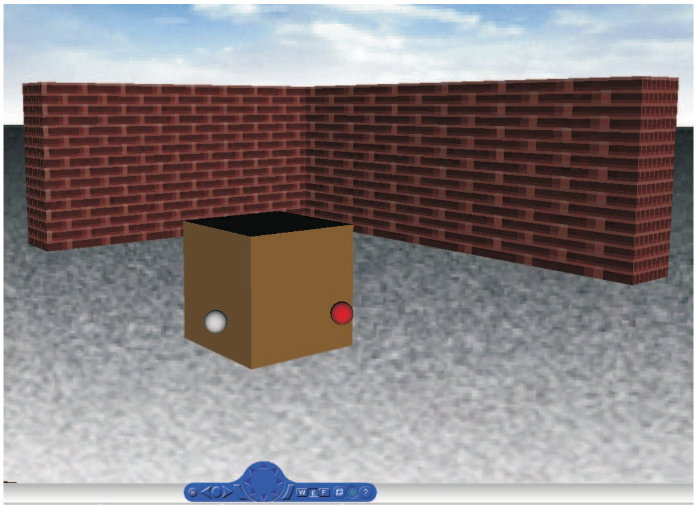

Figure 7.2: Virtual environment snapshot.

simulate the object dynamics.

7.1.3 Real Time Operating System

Quanser QuaRC© generates the real-time code from Simulink®. As suggested by Basdogan and Srinivasan [11] a 1kHz haptic update rate is maintained to provide a realistic haptic interaction with the VE. All elements of the haptic simulator described in this chapter except for Graphics and Network Communication run at update rate of 1kHz.

7.1.4 Graphics

MATLAB® Virtual Reality Toolbox™ provides visual feedback to the users. To trick the eye into perceiving motion an update rate of 32 frames per second is required for the graphic simulator. A snapshot of the virtual environment is shown in Fig. 7.2.

7.1.5 Network Communication

The network communication interface enables geographically distributed workstations to communicate over the Internet. The network interface must take into account the real-time requirements of the haptic simulation. This section describes the network communication elements used in our experiment to connect the workstations.

Protocol

Among the many Internet protocols User Datagram Protocol (UDP) is more suitable for real-time applications (Forouzan [30]) and is therefore used in our experiments as the communication protocol. The following properties makes UDP a suitable protocol for time-sensitive applications such as haptic simulation.

- Connectionless protocol
- No hand-shaking dialogue
- Small packet overhead by removing the checksums
- Compatibility with packet broadcast and packet multicast

However the low latency transmission comes at a cost. UDP cannot guarantee that packets reach their destination or that they arrive in the order that they were transmitted. It also has no strategy for resubmission of lost packets or for reordering out of order packets.

Communication Rate

The packet transmission rate depends on network conditions and remains around 100-200Hz for acceptable communication. In our experiments the communication rate was set to 125Hz to maintain reliable communication between the workstations.

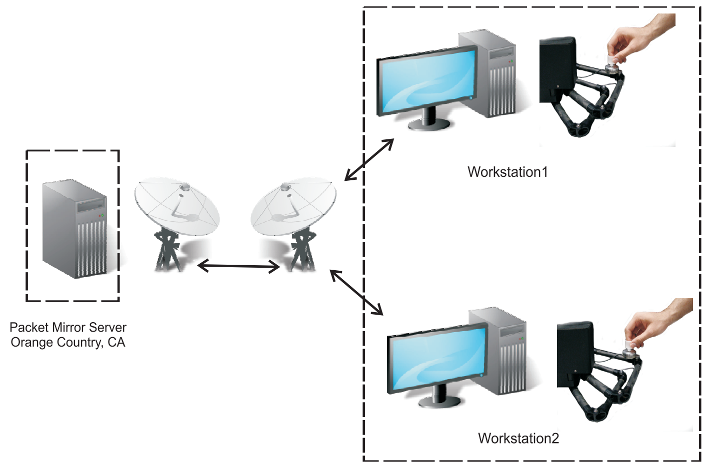

Figure 7.3: Experiment network setup. Icons from http://www.devcom.com/.

Packet Mirror Servers

A packet mirror server is a computer running our packet mirror program. It provides a communication path between our workstations. All communication between the two workstations must pass through one of the packet mirror servers. The packet mirror server can be chosen anywhere in the world to make arbitrary Internet conditions. The packet mirror program enables us to construct actual Internet connection conditions, i.e. delay, jitter, etc., between our lab and another location in the world without the need for the experiment setup to be present in that location. Fig. 7.3 shows the two-user network setup used in our experiments. Packets are transmitted over the Internet to a server in Orange County, CA where they are reflected back to the control workstations at McMaster University. The round-trip delay from our workstations to the packet mirror server was measured around 90ms with negligible jitter and data loss. Therefore, the total round-trip delay between the two workstations is approximately 180ms.

$n = 11$	**Virtual Couplings**					
	$k_{11} = k_{22}$ N/m	$b_{11} = b_{22}$ N.s/m	$k_{o1} = k_{o2}$ N/m	$b_{o1} = b_{o2}$ N.s/m	$k_{21} = k_{12}$ N/m	$b_{21} = b_{12}$ N.s/m
No Prediction	2259	30	12.6	1.3	81.9	3.1
Prediction	2000	29.8	47.1	1.9	105.7	9.1
FF + Prediction	1999	24.1	35.5	0	9.8	1

Table 7.1: Virtual coupling gains for round trip delay of 176ms.

7.2 Experimental Results

The effectiveness of the three proposed schemes have been examined in comparative experiments and the results are presented in this section. The virtual coupling gains have been optimized in Chapter 6 and are presented again in Table 7.1.

7.2.1 Free Motion

Position of the virtual objects along y-axis in free motion of the distributed, predictive and feedforward predictive schemes are shown in Fig. 7.4(a), (b) and (c), respectively. In order to compare position discrepancy among copies of the virtual object in free motion the users were asked to move the box toward each other. User forces are displayed on the same figure. The addition of feedforward and prediction mechanisms improved position tracking between the copies of the virtual object in free motion as was expected from the analysis presented in Chapter 6. In particular the prediction block used in the last two schemes allows larger coupling gains between the copies of the virtual objects. The system becomes unstable if the same coupling gains are used without prediction. By comparing the user force profile in Fig. 7.4(a), (b) and (c), it is seen that less force is required to move the object in the feedforward predictive scheme compared to the first two schemes.

7.2.2 Contact

The distributed, predictive and feedforward predictive schemes were examined in contact with a virtual wall with a stiffness of $k_w = 1000$ N/m and damping of $b_w = 10$ N.s/m. The position of the copies of the virtual object along the y-axis in the three schemes are shown in Fig.7.5. While all schemes were able to provide stable contact with the virtual wall, the addition of the prediction and feedforward mechanisms significantly improved the position tracking as was expected from the analysis presented in Chapter 6. In contact with a virtual wall the feedforward control applies larger forces to the remote object comparing to the forces that the feedback control can provide without causing instability.

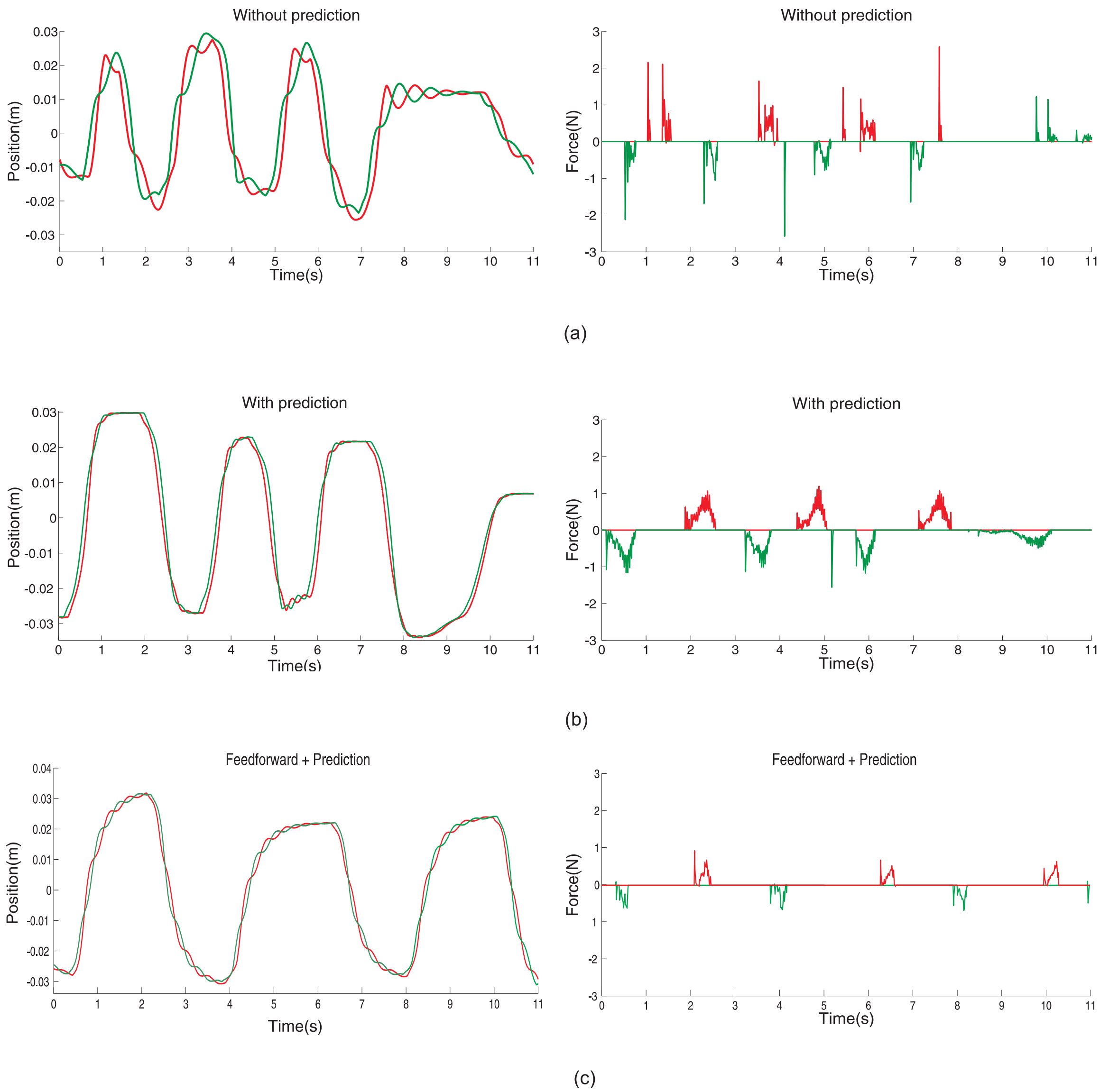

Figure 7.4: Position of the box in free motion. (a) Without prediction; (b) With prediction; (c) Feedforward + prediction.

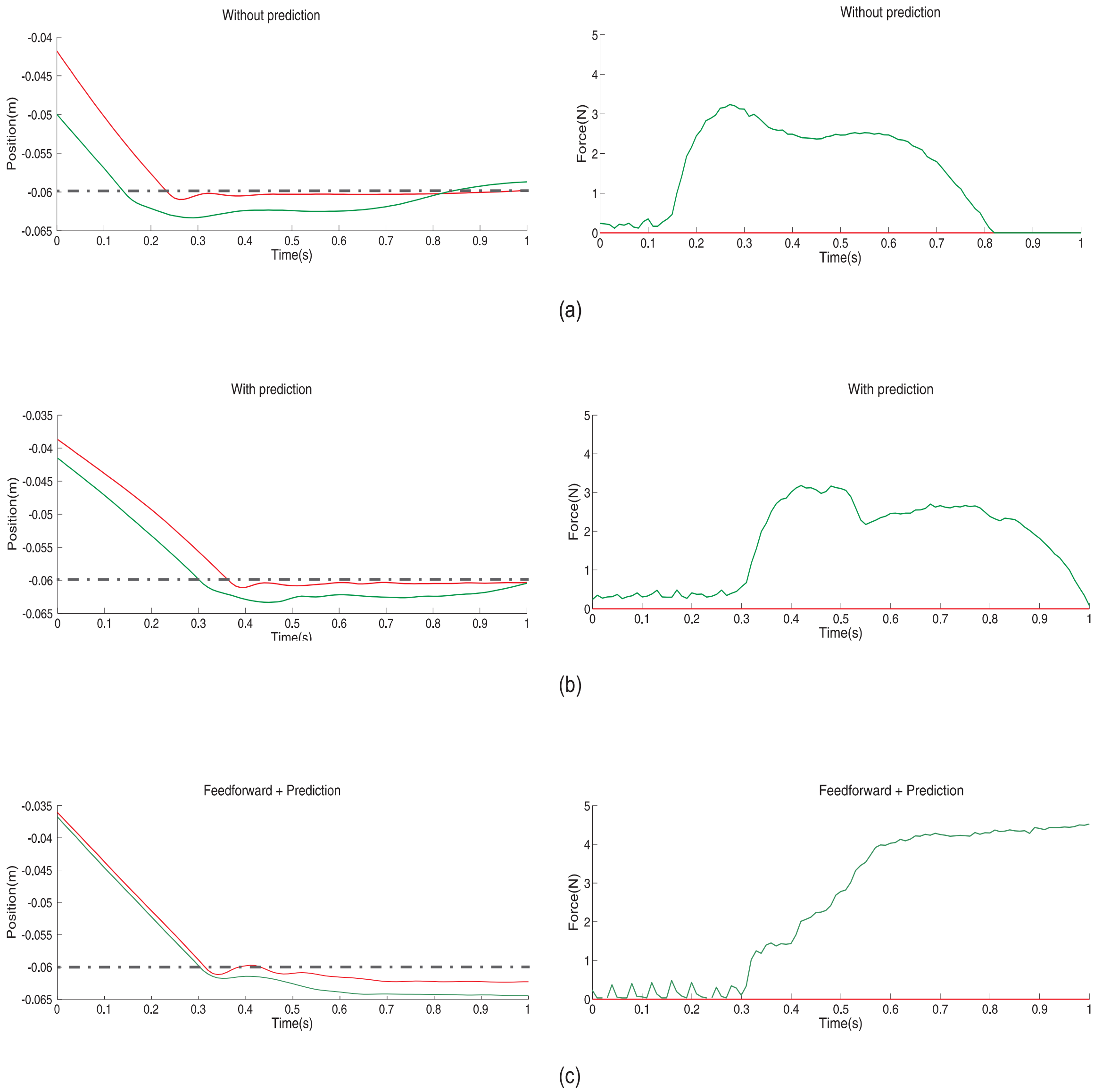

Figure 7.5: Position of the box in contact with a virtual wall at $y = -0.06m$. (a) Without prediction; (b) With prediction; (c) Feedforward + prediction.

Chapter 8

Extension to Three Users

This chapter builds upon the two-user architecture designed in the previous chapters and extends the results to a three-user platform. The addition of a third user to the existing two-user architecture is investigated. The goal is to show the effectiveness of the proposed methods in haptic interactions involving more than two users. Due to its superior performance and stability in the two-user haptic interaction, the feedforward predictive scheme has been employed in this chapter for extension to three users.

8.1 Extending the Architecture

In the distributed architectures, each user interacts with its local copy of the virtual object. Virtual spring-damper couplers synchronize the multiple copies of the virtual object. The addition of the third user to the existing two-user architecture requires another copy of the virtual object as well as virtual coupling gains to synchronize the three copies and to provide force feedback to the third user. Figs. 8.1(a), (b), and (c) display three possible topologies for adding Side 3 to the existing two-user architecture. n_1, n_2, and n_3 are the

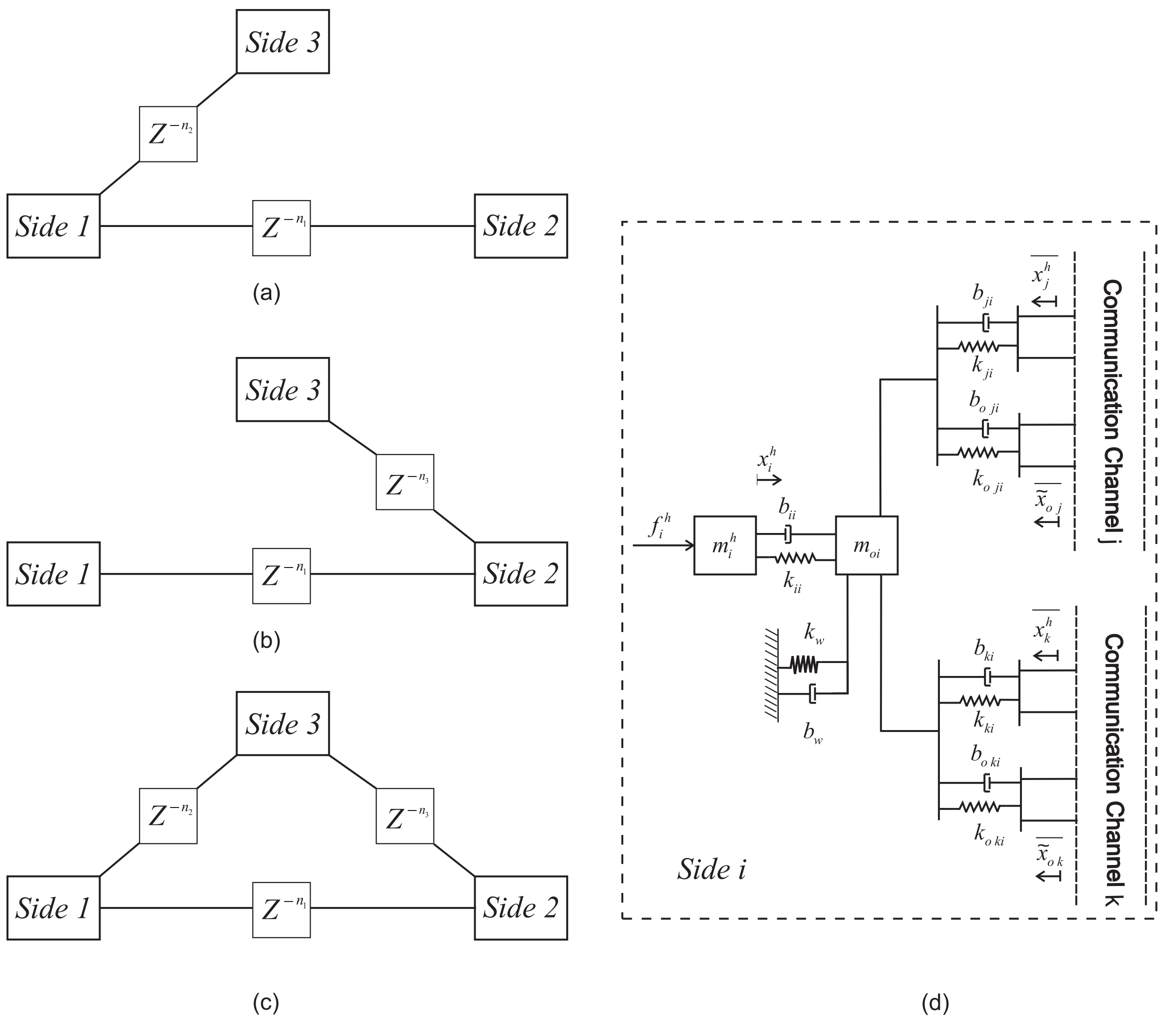

Figure 8.1: (a) Star topology with Side 1 as the central workstation. (b) Star topology with Side 2 as the central workstation. (c) Fully connected topology. (d) Single axis model of one side of the three-user setup in the fully connected topology.

number of sampling rate delays in the three communication channels.

Figs. 8.1(a) and (b) show the star topology in which all workstations are connected to a central workstation. Local copies of the virtual object are connected to the central workstation's copy by virtual spring-damper couplings. In these schemes the position of the haptic interfaces must pass through the central workstation to get to other workstations. In addition, there in no direct coupling between all copies of the virtual object. The central workstation plays an important role in the star topology and in the event of failure in any of the communication channels the associated workstation will be completely disconnected.

Figs. 8.1(c) shows the fully connected topology in which every two workstations are connected through a separate communication channel. The communication of data in this topology is not solely dependant on one communication channel and in the event of failure in one channel the workstations will be connected through the remaining channels. However the fully connected topology requires more communication channels between the workstations and will generate more network traffic. Due to the reliability of the fully connected topology it will be used in the rest of this chapter for extension to three users.

The addition of the third workstations necessities additional virtual couplings to connect the three virtual objects together. Fig. 8.1(d) displays the single-axis model of one side of the three-user architecture. In this figure m_i^h represents the combined mass of the user and haptic device at Side i; m_{oi} denotes the mass of the local copy of the virtual object at Side i; k's and b's are stiffness and damping of corresponding virtual couplers; x's, $\tilde{x}$'s, and $\overline{x}$'s are local, predicted, and network transmitted positions respectively; f_i^h is the user's exogenous force input at Side i; and $i,\ j,\ k\ \in \{1,2,3\},\ i \neq j \neq k.$ a single network sampling rate T_t has been assumed among all the workstations.

A symmetric architecture has been assumed, i.e. $k_{ij} = k_{ji}$ and $k_{o\ ij} = k_{o\ ji}$ for $i,\ j\ \in$

$\{1,2,3\}$, $i \neq j$. This is a reasonable assumption since the communication delay between every two workstations is the same in each direction. The multi-rate modeling approach represented in Chapter 3 can be used to derive the state-space equations of the three-user architecture.

8.2 Performance Measures and Parameter Optimization

Our goal in this section is to define a set of objective functions that can be used in optimizing the extra virtual coupling gains in the three-user architecture. Since the third workstation is being added to the existing two-user architecture, the optimization problem optimizes only the virtual coupling gains associated with the third workstation, i.e. $y = [k_{33}, b_{33}, k_{o13}, b_{o13}, k_{13}, b_{13}, k_{o23}, b_{o23}, k_{23}, b_{23}]$. The system is assumed to be symmetric, i.e. $[k_{o13}, b_{o13}, k_{13}, b_{13}, k_{o23}, b_{o23}, k_{23}, b_{23}] = [k_{o31}, b_{o31}, k_{31}, b_{31}, k_{o32}, b_{o32}, k_{32}, b_{32}]$. The rest of the virtual coupling gains are those found by the solving the optimization problem in Chapter 6.

Similar to the objective functions defined in Chapter 4, the following objective functions are employed in optimizing the three-user architecture.

Perceived Admittance The perceived admittance of the object at Side 3, h_3, is defined as the ratio of the user hand velocity v_3^h to the user's input force f_3^h , i.e. $h_3(j\omega) = \frac{v_3^h}{f_3^h}(j\omega)$. Similarly, the ideal admittance , h_{ideal}, is defined as the ratio of the ideal velocity v to the input force f, i.e. $h_{ideal}(j\omega) = \frac{v}{f}(j\omega)$. The admittance measure is formulated as

$$f_1 = \int_0^{\pi} |h_3(j\omega) - h_{Ideal}(j\omega)|\, w_1(\omega)\, d\omega. \tag{8.1}$$

Position Discrepancy The ability of the system to synchronize multiple copies of the virtual object is reflected in this measure, i.e. $g_i(j\omega) = \frac{x_{oi}}{f_3^h}(j\omega)$. Similarly, the ideal position response, g_{ideal}, is defined as the ratio of the ideal velocity x to the input force f, i.e. $g_{ideal}(j\omega) = \frac{x}{f}(j\omega)$. The virtual object discrepancy measure is formulated as

$$f_2 = \int_0^{\pi} \frac{1}{2}\left(|g_3(j\omega) - g_1(j\omega)| + |g_3(j\omega) - g_2(j\omega)|\right) w_2(\omega)\, d\omega. \tag{8.2}$$

Disturbance Response This measure represents the ability of the system to synchronize the virtual objects in the presence of a disturbance, i.e. $d_i(j\omega) = \frac{x_{oi}}{f_d}(j\omega)$. The disturbance measure is formulated as

$$f_3 = \int_0^{\pi} \frac{1}{2}\left(|d_3(j\omega) - d_1(j\omega)| + |d_3(j\omega) - d_2(j\omega)|\right) w_3(\omega)\, d\omega. \tag{8.3}$$

where w_i's are normalizing factors chosen as:

$$\begin{aligned} w_1(\omega) &= |h_{Ideal}(j\omega)|^{-1}\, l(\omega) \\ w_2(\omega) = w_3(\omega) &= |g_{Ideal}(j\omega)|^{-1}\, l(\omega) \end{aligned} \tag{8.4}$$

and $l(\omega)$ is a frequency dependant gain used to give relevant importance to different frequencies. The above objective functions can be used in formulating a multi-objective optimization problem similar to the one defined in Chapter 4.

8.3 Design Example

The control gains $y = [k_{33}, b_{33}, k_{o13}, b_{o13}, k_{13}, b_{13}, k_{o23}, b_{o23}, k_{23}, b_{23}]$ in the three-user architecture of Fig. 8.1 are optimized for round trip delays of $n_1 = 11$ (176ms), $n_2 = 7$ (112ms), $n_3 = 5$ (80ms). The rest of the system parameters are presented in Table. 8.1. The virtual coupling gains in Table. 8.1 are those found in Chapter 6 by optimizing the two-user architecture. The goal programming method described in Chapter 4 is used to formulate the problem. In this method $f_2(y)$ and $f_3(y)$, are kept in the objective function, whereas, the admittance measure, $f_1(y)$, is moved to the constraint.

$$\begin{aligned} \text{minimize} \quad & f(y) = f_{2free}(y) + f_{3free}(y) + f_{2contact}(y) + f_{3contact}(y) \\ \text{subject to} \quad & \text{stability} \quad \& \quad f_{1free}(y) < 0.6 \quad f_{1contact}(y) < 0.5 \end{aligned} \tag{8.5}$$

The resulting virtual coupling gains and corresponding objective functions are shown in Table 8.2 and Table 8.3, respectively. Fig. 8.2 shows the frequency response of the admittance perceived by the third user in free motion and in contact. The frequency response of the virtual objects' positions due to user's input force is shown in Fig. 8.3. Fig. 8.4 shows the frequency response of the virtual objects' positions in response to a disturbance.

$k_{11} = k_{22} = 2000$ N/m	$b_{11} = b_{22} = 24.1$ N·s/m
$k_{o12} = k_{o21} = 35.5$ N/m	$b_{o12} = b_{o21} = 0$ N·s/m
$k_{12} = k_{21} = 9.4$ N/m	$b_{12} = b_{21} = 1$ N·s/m
$m_i^h = 0.1$ kg	$m_{oi} = 0.4$ kg
$T_c = 1/1000$ s	$N = 8$

Table 8.1: Set of system parameters used in the three-user optimization problem.

$n_1 = 11$, $n_2 = 7$, $n_3 = 5$	**Virtual Couplings**									
	k_{33} N/m	b_{33} N.s/m	$k_{o13} = k_{o31}$ N/m	$b_{o13} = b_{o31}$ N.s/m	$k_{13} = k_{31}$ N/m	$b_{13} = b_{31}$ N.s/m	$k_{o23} = k_{o32}$ N/m	$b_{o23} = b_{o32}$ N.s/m	$k_{23} = k_{32}$ N/m	$b_{23} = b_{32}$ N.s/m
	1916	28	48.6	0.2	60	1.2	53.1	0.7	35.5	0.5

Table 8.2: Virtual coupling gains for $n_1 = 11$, $n_2 = 7$, $n_3 = 5$.

$n_1 = 11$, $n_2 = 7$, $n_3 = 5$	**Free Motion**			**Contact**		
	f_1	f_2	f_3	f_1	f_2	f_3
	0.49	0.06	0.13	0.45	0.43	0.86

Table 8.3: Objective functions for $n_1 = 11$, $n_2 = 7$, $n_3 = 5$.

8.4 Experimental Results

A three-user haptic system has been set up to examine the effectiveness of the proposed design. This three-user platform consists of one PHANTOM Premium 1.5A Haptic Device by SensAble and the two Quanser pantographs used in the two-user system. Although the PHANTOM provides three-degree-of-freedom force feedback, its movement along the z-axis has been disabled to be consistent with the two pantographs as shown in Fig. 8.5. Due to limitations of PHANTOM in rendering high forces the local coupling between the phantom and the virtual object have been reduced to, $k_{33} = 600\text{N/m}$, and $b_{33} = 10\text{N.s/m}$. The three workstations have been located in separate cubicles in our lab to avoid direct visual feedback between the users.

Although it was possible to use network packet reflectors as in Chapter 7 in the experiments, this setup requires three packet reflector servers and therefore constant artificial delay was employed to emulate the communication delays instead.

Fig. 8.6 displays the position of the three copies of the virtual objects in free motion and the forces applied to the users as the users pushed the box toward each other. The

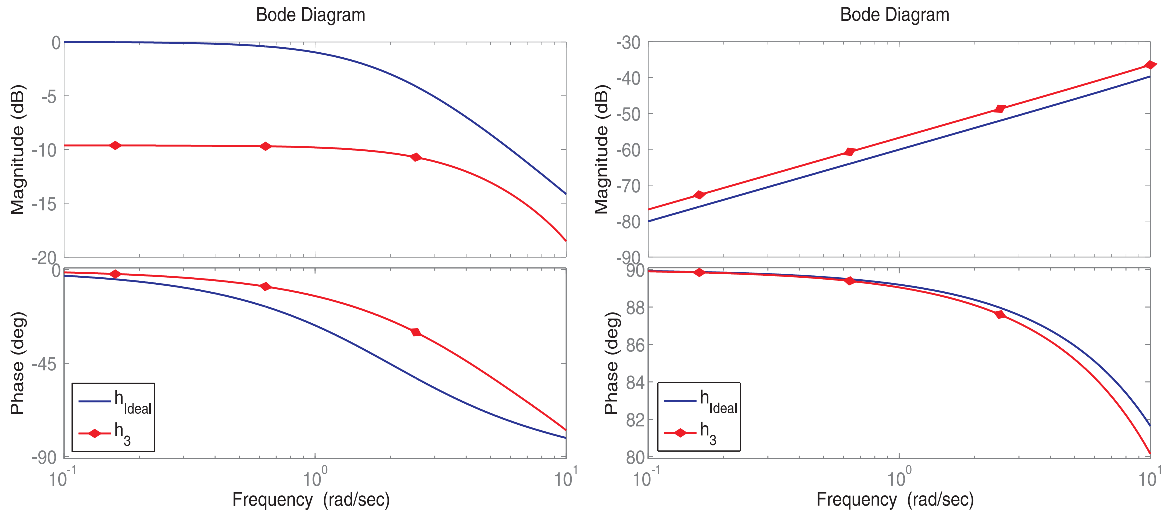

Figure 8.2: User's perceived admittance in the three-user setup $h_1(j\omega)$ and ideal admittance $h_{Ideal}(j\omega)$ in free motion (left figure) and contact (right figure).

positions of the copies of the virtual box along the y-axis in contact with a virtual wall with a stiffness of $k_w = 1000$ N/m and damping of $b_w = 10$ N.s/m are shown in Fig. 8.7. The dashed line shows the position of the virtual wall at $y = -0.06m$.

The three-user haptic system was able to provide stable haptic interaction with the virtual box in free motion and contact, as expected from the analysis presented in Section 8.3. The same approach can be employed to extend the architecture to more than three users. For instance, a fourth user can be added to the existing three-user architecture by keeping the virtual coupling gains used in this chapter and optimizing a set of virtual coupling gains associated with the fourth user.

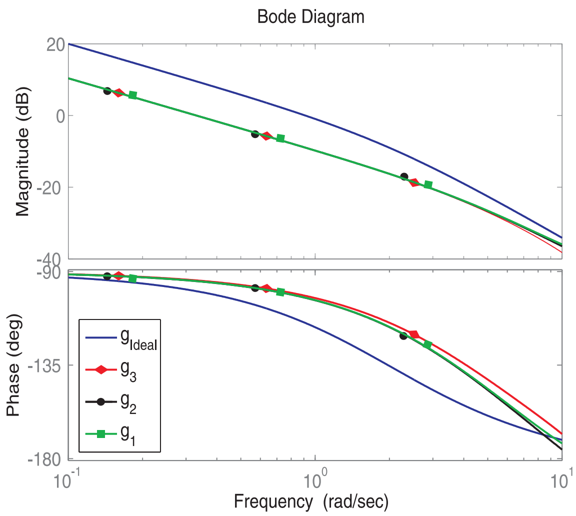

Figure 8.3: Object position $g_i\,(j\omega)$ and ideal position $g_{Ideal}\,(j\omega)$ in the three-user setup in free motion (left column) and contact (right column).

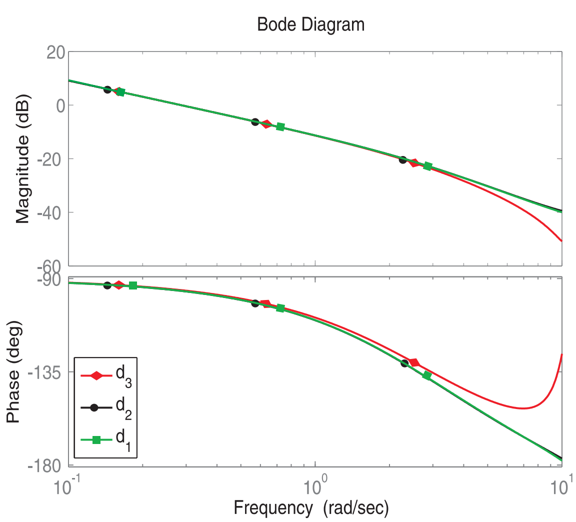

Figure 8.4: Response to disturbance $d_i\,(j\omega)$ in the three-user setup in free motion (left column) and contact (right column).

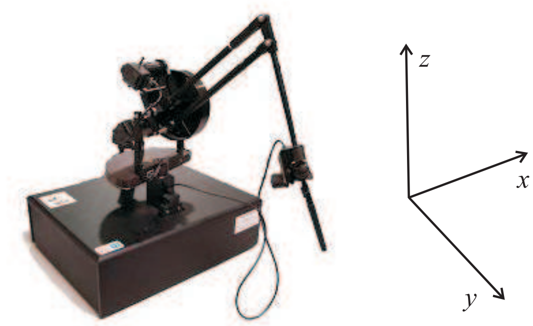

Figure 8.5: PHANTOM Premium 1.5A Haptic Device by SensAble and the coordinate frame assigned to it.

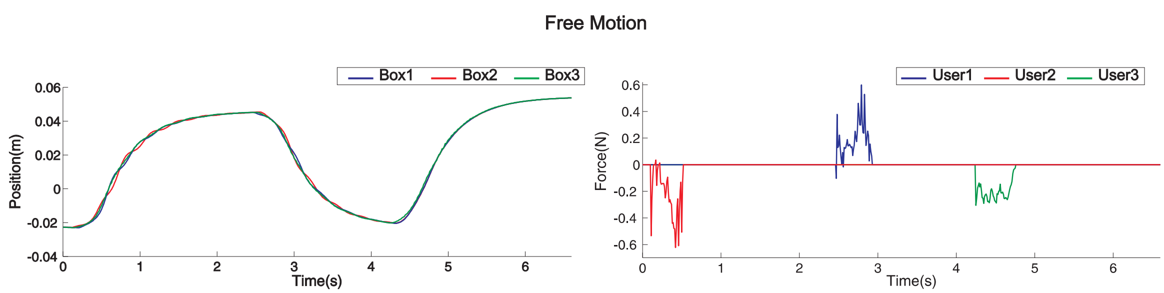

Figure 8.6: Position of the virtual objects (left figure) and force applied to the users (right figure) in free motion.

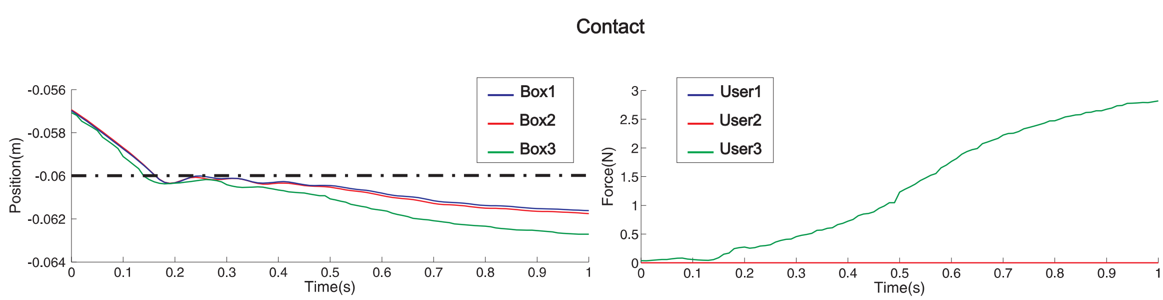

Figure 8.7: Position of the virtual objects (left figure) and force applied to the users (right figure) in contact with a virtual wall at $y = -0.06m$.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

The fidelity and stability of haptic interaction in network-based virtual environments degrade due to network impediments such as time delay, delay jitter, limited packet transmission rate, and packet loss. This work was primarily concerned with the effect of network communication delay on the performance and stability of cooperative haptic simulations over a WAN.

A recent study conducted in our group by Fotoohi *et al.* [31] compared the performance and stability of centralized and distributed control architectures over a Local Area Network (LAN). It was demonstrated in their work that a distributed control architecture can improve haptic fidelity in free motion and in contact with rigid environments over a centralized control architecture by allowing high-rate feedback with local copies of the virtual environment. A distributed control architecture was employed in our work due to its superiority in network-based haptic simulations. High-rate local simulation of the virtual objects and low-rate communication over the network result in a multi-rate discrete-time system. The subsystem resampling approach in modeling multi-rate systems was used to obtain the mathematical description of the system.

In the distributed architecture, virtual coupling gains synchronize multiple copies of the virtual object and provide feedback to the users. The values of these virtual coupling gains affect the performance and stability of the haptic simulation. In order to improve the interaction transparency, an optimization problem was formulated for selecting the virtual coupling gains. Quantitative measures of user's perceived admittance and position discrepancy between multiple copies of the virtual object were employed as objective functions in formulating the optimization problem.

The control architecture was modified by introducing a prediction mechanism to compensate for the negative effects of the time delay on performance and stability of the haptic simulation. Based on the virtual object's dynamics, the prediction mechanism predicts the position and velocity of each copy of the virtual object over the prediction horizon. This prediction is sent over the communication channel instead of the object's states. To further improve the transparency of the haptic simulation, a feedforward mechanism was designed to reduce the discrepancy between the multiple copies of the virtual object due to disturbances and contact nonlinearities. Numerical analysis as well as experiments conducted with a two-user haptic platform over the Internet demonstrated the effectiveness of the prediction and feedforward mechanisms in improving the user's perceived admittance as well as the position tracking between copies of the object in free motion and in contact with a virtual wall.

The predictive-feedforward architecture was extended to a three-user platform to show the effectiveness of the proposed methods in haptic simulations involving more than two users. An optimization problem similar to that defined for the two-user architecture was formulated for finding the new virtual coupling gains associated with the third user. Experiments carried out on a three-user platform verified stable haptic interaction in free motion and in contact with a virtual wall.

9.2 Suggestions for Future Work

The following topics are a number of potential directions for future research:

- A single network sampling rate was assumed in this research. In multi-user haptic interaction workstations can have different network rates based on their hardware and network characteristics. The effect of multiple network rates in cooperative haptic interactions can be investigated in future.

- In the proposed predictor, the net force applied to the object was assumed constant over the prediction horizon. This assumption is only valid when the net force does not change abruptly over the prediction horizon. More realistic prediction models have to be developed in future. In addition, a prediction of the user's hand position can improve the transparency of the haptic simulation.

- The state prediction and optimization of the gains were performed for a constant network delay and no packet loss. However, depending on network conditions, network delay may be subject to jitter and can change over time. The proposed control framework can be extended to the case of variable time delay. The amount of delay can be estimated by attaching time stamps to the packets or by a separate routine running at a lower rate than the communication rate. In addition, lost packets can be recovered through extrapolation of the previous packets. The effect of such mechanisms on the stability and transparency of the haptic simulation can be investigated.

- The extension to four or more users can be implemented using various network topologies, e.g. star, ring, line, fully connected, etc. The effect of different topologies on stability, performance and network traffic can be investigated.

Appendix A

State-space Representation

The state-space equations of the system of Fig. A.1 is derived in this section. Discrete-time and multi-rate nature of the system as well as the computation and communication delays are considered in deriving the equations. The state-space equations of the three-user architecture introduced in Chapter 8 can be derived using the same approach.

A.1 Single Mass

It is useful to derive the state-space equations of a single mass system of Fig. A.2. The matrices will be used in constructing the state matrices of the system of Fig. A.1.

A.1.1 Continuous-time

$$\begin{aligned} \dot{X}(t) &= AX(t) + Bu(t) \\ y(t) &= CX(t) + Du(t) \end{aligned} \tag{A.1}$$

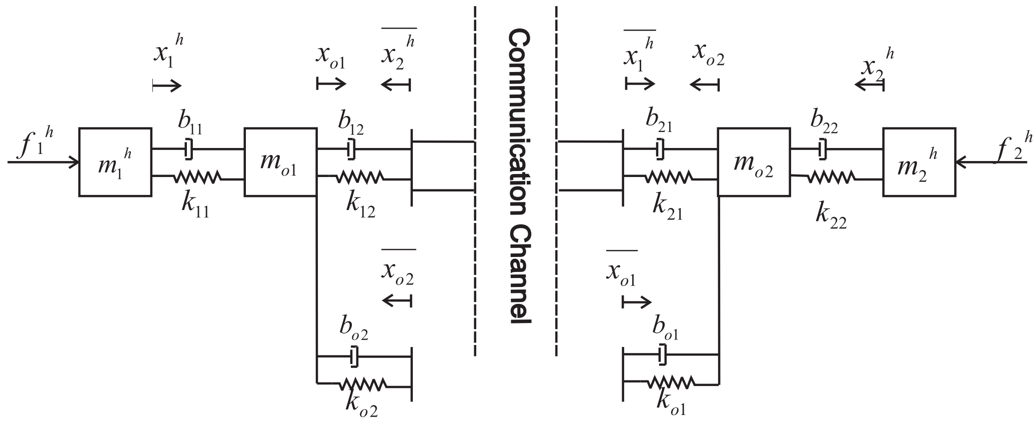

Figure A.1: Model of distributed single-axis cooperative haptics.

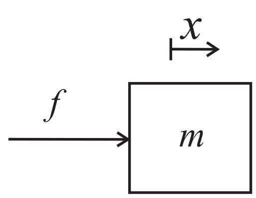

Figure A.2: Single mass.

where

$$X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \tag{A.2}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, u = [f]$$

where m is the mass of the object, x and $\dot{x}$ are position and velocity of the object, respectively, and f is the net force applied to the object.

A.1.2 Discrete-time

$$
\begin{aligned}
X(k+1) &= A_D X(k) + B_D u(k) \\
y(k) &= C_D X(k) + D_D u(k)
\end{aligned}
\tag{A.3}
$$

where

$$
\begin{aligned}
A_D = e^{AT} \quad & B_D = \left(\int_0^T e^{A\eta} d\eta\right) B \\
C_D = C \quad & D_D = D
\end{aligned}
\tag{A.4}
$$

Applying the above transformation to the system of Eq. (A.1), the discrete-time state-space representation of a single mass is

$$
\begin{aligned}
A_D &= \begin{bmatrix} 1 & T_c \\ 0 & 1 \end{bmatrix}, B_D = \begin{bmatrix} \frac{{T_c}^2}{2m} \\ \frac{T_c}{m} \end{bmatrix} \\
C_D &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D_D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{A.5}
$$

A.1.3 Augmenting Computation Delay

Computation delay can be augmented into the discrete-time model of the mass derived in SectionA.1.2. The new state transition matrices and state vectors for one sample time

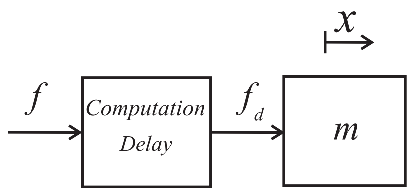

Figure A.3: Single mass with one sample computation delay.

computation delay are

$$\tilde{X}=\begin{bmatrix} x \\ \dot{x} \\ f_d \end{bmatrix},\tilde{A_D}=\begin{bmatrix} A_D & B_D \\ 0 & 0 \end{bmatrix}=\begin{bmatrix} 1 & T_c & \frac{T_c{}^2}{2m} \\ 0 & 1 & \frac{T_c}{m} \\ 0 & 0 & 0 \end{bmatrix},\tilde{B_D}=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.6}$$

$$\tilde{C_D}=\begin{bmatrix} C_D & D_D \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},\tilde{D_D}=\begin{bmatrix} 0 \\ 0 \end{bmatrix},\tilde{u}=[f]$$

where f_d is the one sampled delayed input.

A.2 Side 1

Considering one sample computation delay, the net force, f_1, applied to the user1, m_1^h, at side1 can be found from Fig. A.4.

$$f_{1d}=f_1^h-k_{11}(x_1^h-x_{o1})-b_{11}(\dot{x}_1^h-\dot{x}_{o1}) \tag{A.7}$$

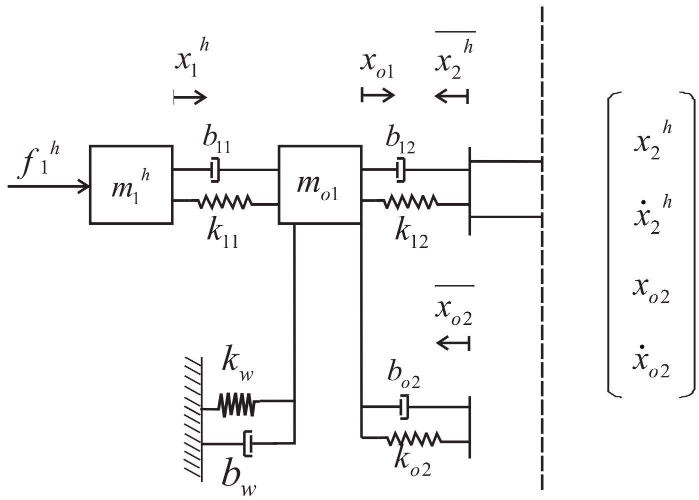

Figure A.4: Single-axis model of side1 of the two-user haptic system.

The net force, f_{o2}, applied to the object, m_1^h, at side1 can be found from the same figure. The opposite position references at the two sides must be considered in writing the equations.

$$\begin{aligned} f_{o1} = \quad & -k_{11}(x_{o1} - x_1^h) - b_{11}(\dot{x}_{o1} - \dot{x}_1^h) - k_{12}(x_{o1} + x_2^h) - b_{12}(\dot{x}_{o1} + \dot{x}_2^h) \\ & -k_{o2}(x_{o1} + x_{o2}) - b_{o2}(\dot{x}_{o1} + \dot{x}_{o2}) - k_w x_{o1} - b_w \dot{x}_{o1} \end{aligned} \tag{A.8}$$

Replacing Eq. (A.7) and (A.8) in discrete-time model of Section A.1.3 and A.1.2 and combining the dynamic equations of the two masses one can derive the state-space matrices of the system.

$$X_{side1} = \begin{bmatrix} x_1^h \\ \dot{x}_1^h \\ f_{1d} \\ x_{o1} \\ \dot{x}_{o1} \end{bmatrix}, \; u_{side1} = \begin{bmatrix} x_2^h \\ \dot{x}_2^h \\ x_{o2} \\ \dot{x}_{o2} \\ f_1^h \end{bmatrix}$$

$$A_{side1} = \begin{bmatrix} 1 & T_c & \frac{{T_c}^2}{2m_1^h} & 0 & 0 \\ 0 & 1 & \frac{T_c}{m_1^h} & 0 & 0 \\ -k_{11} & -b_{11} & 0 & k_{11} & b_{11} \\ \frac{{T_c}^2}{2m_{o1}}k_{11} & \frac{{T_c}^2}{2m_{o1}}b_{11} & 0 & 1-\frac{{T_c}^2}{2m_{o1}}(k_{11}+k_{12}+k_{o2}+k_w) & T_c-\frac{{T_c}^2}{2m_{o1}}(b_{11}+b_{12}+b_{o2}) \\ \frac{T_c}{m_{o1}}k_{11} & \frac{T_c}{m_{o1}}b_{11} & 0 & -\frac{T_c}{m_{o1}}(k_{11}+k_{12}+k_{o2}+k_w) & 1-\frac{T_c}{m_{o1}}(b_{11}+b_{12}+b_{o2}) \end{bmatrix}$$

$$B_{side1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{{T_c}^2}{2m_{o1}}k_{12} & -\frac{{T_c}^2}{2m_{o1}}b_{12} & -\frac{{T_c}^2}{2m_{o1}}k_{o2} & -\frac{{T_c}^2}{2m_{o1}}b_{o2} & 0 \\ -\frac{T_c}{m_{o1}}k_{12} & -\frac{T_c}{m_{o1}}b_{12} & -\frac{T_c}{m_{o1}}k_{o2} & -\frac{T_c}{m_{o1}}b_{o2} & 0 \end{bmatrix}$$

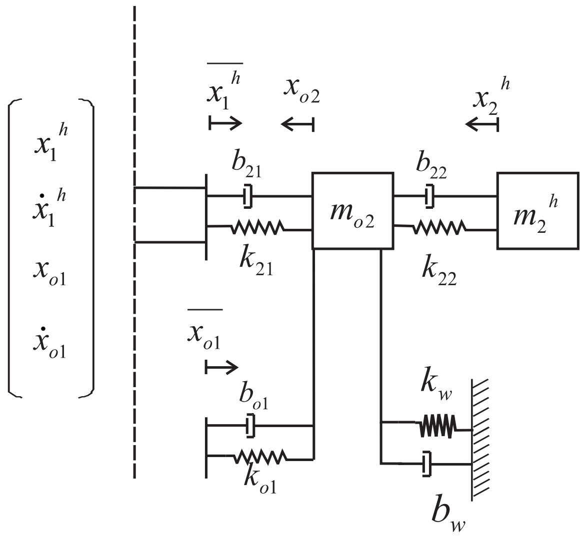

Figure A.5: Single-axis model of side2 of the two-user haptic system.

$$C_{side1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; D_{side1} = \begin{bmatrix} 0 \end{bmatrix}_{4\times 5}$$

A.3 Side 2

The system matrices of side2 can be written in the same way as side1 by placing the appropriate virtual coupling gains and removing the input force.

$$X_{side2} = \begin{bmatrix} x_2^h \\ \dot{x}_2^h \\ f_{2d} \\ x_{o2} \\ \dot{x}_{o2} \end{bmatrix}, \; u_{side2} = \begin{bmatrix} x_1^h \\ \dot{x}_1^h \\ x_{o1} \\ \dot{x}_{o1} \end{bmatrix}$$

$$A_{side2} = \begin{bmatrix} 1 & T_c & \frac{{T_c}^2}{2m_2^h} & 0 & 0 \\ 0 & 1 & \frac{T_c}{m_2^h} & 0 & 0 \\ -k_{22} & -b_{22} & 0 & k_{22} & b_{22} \\ \frac{{T_c}^2}{2m_{o2}}k_{22} & \frac{{T_c}^2}{2m_{o2}}b_{22} & 0 & 1-\frac{{T_c}^2}{2m_{o2}}(k_{22}+k_{21}+k_{o1}+k_w) & T_c-\frac{{T_c}^2}{2m_{o2}}(b_{22}+b_{21}+b_{o1}) \\ \frac{T_c}{m_{o2}}k_{22} & \frac{T_c}{m_{o2}}b_{22} & 0 & -\frac{T_c}{m_{o2}}(k_{22}+k_{21}+k_{o1}+k_w) & 1-\frac{T_c}{m_{o2}}(b_{22}+b_{21}+b_{o1}) \end{bmatrix}$$

$$B_{side2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{{T_c}^2}{2m_{o2}}k_{21} & -\frac{{T_c}^2}{2m_{o2}}b_{21} & -\frac{{T_c}^2}{2m_{o2}}k_{o1} & -\frac{{T_c}^2}{2m_{o2}}b_{o1} \\ -\frac{T_c}{m_{o2}}k_{21} & -\frac{T_c}{m_{o2}}b_{21} & -\frac{T_c}{m_{o2}}k_{o1} & -\frac{T_c}{m_{o2}}b_{o1} \end{bmatrix}$$

$$C_{side2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},\ D_{side2} = \begin{bmatrix} 0 \end{bmatrix}_{4\times 4}$$

A.4 Downsampling and Merging

The two subsystems can be downsampled by the following transformation

$$\begin{aligned}\widetilde{A}_t &= A_c^N \\ \widetilde{B}_t &= A_c^{N-1}B_c + A_c^{N-2}B_c + \cdots + A_cB_c + B_c \\ \widetilde{C}_t &= C_c \quad \widetilde{D}_t = D_c\end{aligned} \tag{A.9}$$

where $N = \frac{T_t}{T_c}$.

At this stage the communication delay can be augmented in the system matrices and the resulting subsystems will be merged.

$$A = \left[\begin{array}{c|c|c|c|c} \widetilde{A}_{side1} & 0 & 0 & 0 & \widetilde{B}_{side1}(:,1:4) \\ \hline 0 & \widetilde{A}_{side2} & 0 & \widetilde{B}_{side2}(:,1:4) & 0 \\ \hline \widetilde{C}_{side1} & 0 & 0 & 0 & 0 \\ \hline 0 & \widetilde{C}_{side2} & 0 & 0 & 0 \\ \hline 0 & 0 & I & 0 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} \widetilde{B}_{side1}(:,5) \\ \hline 0 \end{array}\right]$$

The state vector is consisted of the two sides' velocities and positions.

$$X = \begin{bmatrix} \widetilde{X}_{side1}(k) \\ \hline \widetilde{X}_{side2}(k) \\ \hline \hat{X}_{side1}(k-1) \\ \hline \hat{X}_{side2}(k-1) \\ \vdots \\ \hat{X}_{side1}(k-n) \\ \hline \hat{X}_{side2}(k-n) \end{bmatrix}, \widetilde{X}_{sidei} = \begin{bmatrix} x_i^h \\ \dot{x}_i^h \\ f_{id} \\ x_{oi} \\ \dot{x}_{oi} \end{bmatrix}, \hat{X}_{sidei} = \begin{bmatrix} x_i^h \\ \dot{x}_i^h \\ x_{oi} \\ \dot{x}_{oi} \end{bmatrix}, i = 1,2$$

where 0 and I are zero and identity matrices of appropriate dimensions. The force applied at side1, f, is the input to the system. Depending on the output of interest the matrix C can output any of the states ,and $D = [0]$.

Bibliography

[1] Abbott, J. and Okamura, A. (2005). Effects of Position Quantization and Sampling Rate on Virtual-wall Passivity. *IEEE Transactions on Robotics*, **21**(5), 952–964.

[2] Abdossalami, A. and Sirouspour, S. (2009). Adaptive Control for Improved Transparency in Haptic Simulations. *IEEE Transactions on Haptics*, **2**(1), 2–14.

[3] Acosta, E. and Temkin, B. (2005). Haptic Laparoscopic Skills Trainer with Practical User Evaluation Metrics. *Studies in Health Technology and Informatics*, **111**, 8–11.

[4] Alhalabi, M. O. and Horiguchi, S. (2003a). Haptic cooperative virtual workspace: Architecture and evaluation. *Virtual Reality*, **5**(3), 160–168.

[5] Alhalabi, M. O. and Horiguchi, S. (2003b). Network latency issue in cooperative shared haptic virtual environment. volume 4756, pages 199–205. SPIE.

[6] Anderson, R. and Spong, M. (1988). Bilateral Control of Teleoperators with Time Delay. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, 1988.*, volume 1.

[7] Anderson, R. and Spong, M. (1989). Asymptotic Stability for Force Reflecting Teleoperators with Time Delays. In *Proceedings of the IEEE International Conference on Robotics and Automation, 1989.*, pages 1618–1625.

[8] Andrews, S., Mora, J., Lang, J., and Lee, W. (2006). HaptiCast: A Physically-Based 3D Game with Haptic Feedback. *Proceedings of FuturePlay*.

[9] Arioui, H., Kheddar, A., and Mammar, S. (2002a). A Predictive Wave-based Approach for Time Delayed Virtual Environments Haptics Systems. In *11th IEEE International Workshop on Robot and Human Interactive Communication, 2002.*, pages 134–139.

[10] Arioui, H., Mammar, S., and Hamel, T. (2002b). A Smith-prediction Based Haptic Feedback Controller for Time Delayed Virtual Environments Systems. In *Proceedings of the 2002 American Control Conference.*, volume 5.

[11] Basdogan, C. and Srinivasan, M. (2002). Haptic Rendering in Virtual Environments. *Handbook of virtual environments: Design, implementation, and applications*, pages 117–134.

[12] Basdogan, C., Ho, C.-H., Srinivasan, M. A., and Slater, M. (2000). An experimental study on the role of touch in shared virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, **7**(4), 443–460.

[13] Basdogan, C., De, S., Kim, J., Muniyandi, M., Kim, H., and Srinivasan, M. (2004). Haptics in Minimally Invasive Surgical Simulation and Training. *IEEE Computer Graphics and Applications*, **24**(2), 56–64.

[14] Blanch, R., Ferley, E., Cani, M., and Gascuel, J. (2004). Non-realistic Haptic Feedback for Virtual Sculpture.

[15] Brewster, S. (2005). Impact of Haptic Touching Technology on Cultural Applications.

[16] Broeren, J., Georgsson, M., Rydmark, M., and Sunnerhagen, K. (2002). Virtual Reality in Stroke Rehabilitation with the Assistance of Haptics and Telemedicine. In *Proceedings of the 4th International Conference on Disability, Virtual Reality and Associated Technology*, pages 71–76.

[17] Brooks, S. (1958). A Discussion of Random Methods for Seeking Maxima. *Operations Research*, **6**(2), 244–251.

[18] Buttolo, P., Oboe, R., and Hannaford, B. (1997). Architectures for Shared Haptic Virtual Environments. *Computers and Graphics*, **21**(4), 421–429.

[19] Cao, Y., Zhang, Y., Ma, Y., and Wang, D. (2007). Design and Analysis of a New Six-DOF Haptic Device for Dental Training. In *Proceedings of the 16th IEEE International Symposium on Robot and Human interactive Communication, 2007. RO-MAN 2007.*, pages 263–267.

[20] Carignan, C. and Olsson, P. (1926). Cooperative Control of Virtual Objects over the Internet Using Force-reflecting Master Arms. In *IEEE International Conference on Robotics and Automation, 2004. ICRA'04.*, volume 2.

[21] Chang, B. and Colgate, J. (2002). *Real-time Impulse-based Simulation of Rigid Body Systems for Haptic Display*. Northwestern University.

[22] Cheong, J., Niculescu, S.-I., Annaswamy, A., and Srinivasan, M. A. (2005). Motion synchronization in virtual environments with shared haptics and large time delays. volume 0, pages 277–282, Los Alamitos, CA, USA. IEEE Computer Society.

[23] Cho, C., Kim, M., Hwang, C., Lee, J., and Song, J. (2005). Stable Haptic Display of Slowly Updated Virtual Environment with Multirate Wave Transform. In *Proceedings*

of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005., pages 2465–2470.

[24] Colgate, J., Stanley, M., and Brown, J. (1995). Issues in the haptic display of tool use. volume 3, pages 140–145 vol.3.

[25] Constantinescu, D., Salcudean, S., and Croft, E. (2005). Haptic Rendering of Rigid Contacts Using Impulsive and Penalty Forces. *IEEE Transactions on Robotics*, **21**(3), 309–323.

[26] Dai, X., Zhang, Y., Cao, Y., and Wang, D. (2008). Stable Multirate Control Algorithm for Haptic Dental Training System. In *Proceedings of the First International Conference on Intelligent Robotics and Applications: Part II*, pages 27–35. Springer.

[27] Diolaiti, N., Niemeyer, G., Barbagli, F., and Salisbury Jr, J. (2006). Stability of Haptic Rendering: Discretization, Quantization, Time delay, and Coulomb Effects. *IEEE Transactions on Robotics*, **22**(2), 256–268.

[28] El-Far, N., Shen, X., and Georganas, N. (2004). Applying Unison, a Aeneric Framework for Hapto-visual Application Developments, to an E-commerce Application. In *Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004.*, pages 93–98.

[29] Eom, K., Suh, I., and Yi, B. (2000). A Design Method of a Haptic Interface Controller Considering Transparency and Robust Stability. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.(IROS 2000).*, volume 2.

[30] Forouzan, B. (2003). *Data Communications and Networking*. McGraw-Hill Science/Engineering/Math.

[31] Fotoohi, M., Sirouspour, S., and Capson, D. (2007). Stability and Performance Analysis of Centralized and Distributed Multi-rate Control Architectures for Multi-user Haptic Interaction. *The International Journal of Robotics Research*, **26**(9), 977.

[32] Frank Dachille, I., Qin, H., Kaufman, A., and El-Sana, J. (1999). Haptic Sculpting of Dynamic Surfaces. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 103–110. ACM New York, NY, USA.

[33] Gass, S. (1986). A Process for Determining Priorities and Weights for Large-scale Linear Goal Programmes. *Journal of the Operational Research Society*, pages 779–785.

[34] Gerovichev, O., Marayong, P., and Okamura, A. (2002). The Effect of Visual and Haptic Feedback on Manual and Teleoperated Needle Insertion. *Lecture Notes in Computer Science*, pages 147–154.

[35] Ghiam, M. F., Sirouspour, S., Capson, D. W., of Electrical, M. U. D., and Engineering, C. (2006). *Multi-rate Control Architectures for Network-based Multi-user Haptics Interaction*. by Mahyar Fotoohi Ghiam.; xiii, 103 leaves : ill. ; 28 cm; Supervisor: Shahin Sirouspour, David Capson.; Thesis (M.A.Sc.) – McMaster University, 2006.; Includes bibliographical references.

[36] Gil, J., Avello, A., Rubio, A., and Florez, J. (2004). Stability Analysis of a 1 DOF Haptic Interface Using the Routh-hurwitz Criterion. *IEEE Transactions on Control Systems Technology*, **12**(4), 583–588.

[37] Gil, J., Sanchez, E., Hulin, T., Preusche, C., and Hirzinger, G. (2007). Stability Boundary for Haptic Rendering: Influence of Damping and Delay. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 124–129.

[38] Gillespie, R. and Cutkosky, M. (1996). Stable User-specific Haptic Rendering of the Virtual Wall. In *Proceedings of the ASME International Mechanical Engineering Congress and Exhibition*, volume 58, pages 397–406.

[39] Gourishankar, V., Srimathveeravalli, G., and Kesavadas, T. (2007). HapStick: A High Fidelity Haptic Simulation for Billiards. In *Proceedings of the Second Joint Euro-Haptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007.*, pages 494–500.

[40] Griffiths, P. (2008). Design and Analysis of Haptic Interface and Teleoperator Feedback Systems.

[41] Griffiths, P. and Gillespie, R. (2008). Characterizing Teleoperator Behavior for Feedback Design and Performance Analysis. In *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2007. HAPTICS 2007.*, pages 273–280.

[42] Gutwin, C. (2001). The Effects of Network Delays on Group wWrk in Real-time Groupware. In *Proceedings of the European Conference on CSCW*, pages 299–318.

[43] Hannaford (1991). Performance evaluation of a six-axis generalized force-reflecting teleoperator. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(3), 620.

[44] Hayward, V. and MacLean, K. (2007). Do it Yourself Haptics: Part I. *IEEE Robotics and Automation Magazine*, **14**(4), 88–104.

[45] Hikichi, K., Morino, H., Arimoto, I., Sezaki, K., and Yasuda, Y. (2002). The Evaluation of Delay Jitter for Haptics Collaboration over the Internet. In *Proceedings og the IEEE Global Telecommunications Conference, 2002. GLOBECOM'02*, volume 2.

[46] Hinterseer, P., Steinbach, E., and Chaudhuri, S. (2006). Perception-based Compression of Haptic Data Streams Using Kalman Filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, Toulouse, France*.

[47] Ho, C., Basdogan, C., and Srinivasan, M. (2000). Ray-based Haptic Rendering: Force and Torque Interactions between a Line Probe and 3D Objects in Virtual Environments. *The International Journal of Robotics Research*, **19**(7), 668.

[48] Hollerbach, J., Cohen, E., Thompson, W., Freier, R., Johnson, D., Nahvi, A., Nelson, D., Thompson, T., and Jacobsen, S. (1997). Haptic Interfacing for Virtual Prototyping of Mechanical CAD Designs. In *ASME Design for Manufacturing Symposium*, pages 201–207. Kluwer Academic Publishers.

[49] Houtsma, J. and Van Houten, F. (2006). Virtual Reality and a Haptic Master-Slave Set-Up in Post-Stroke Upper-Limb Rehabilitation. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, **220**(6), 715–718.

[50] Hu, T., Castellanos, A., Tholey, G., and Desai, J. (2002). Real-time Haptic Feedback in Laparoscopic Tools for Use in Gastro-intestinal Surgery. *Lecture Notes in Computer Science*, pages 66–74.

[51] Kim, L. and Park, S. (2006). Haptic Interaction and Volume Modeling Techniques for Realistic Dental Simulation. *The Visual Computer*, **22**(2), 90–98.

[52] Kretz, A., Huber, R., and Fjeld, M. (2005). Force Feedback Slider (FFS): Interactive Device for Learning System Dynamics. In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, 2005. ICALT 2005*, pages 457–458.

[53] Kuchenbecker, K., Fiene, J., and Niemeyer, G. (2006a). Improving Contact Realism through Event-based Haptic Feedback. *IEEE Transactions on Visualization and Computer Graphics*, **12**(2), 219–230.

[54] Kuchenbecker, K. J., Fiene, J., and Niemeyer, G. (2006b). Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, **12**(2), 219–230.

[55] Lawrence, D., Pao, L., Dougherty, A., Salada, M., and Pavlou, Y. (2000). Rate-hardness: A New Performance Metric for Haptic Interfaces. *IEEE Transactions on Robotics and Automation*, **16**(4), 357–371.

[56] Lécuyer, A., Mobuchon, P., Megard, C., Perret, J., Andriot, C., and Colinot, J. (2003). HOMERE: a Multimodal System for Visually Impaired People to Explore Vrtual Environments. *IEEE Virtual Reality, 2003. Proceedings*, pages 251–258.

[57] Levesque, V. (2005). Blindness, technology and Haptics. *Center for Intelligent Machines*.

[58] Macedonia, M. and Noll, S. (1997). A Transatlantic Research and Development Environment [3D virtualgraphics]. *IEEE Computer Graphics and Applications*, **17**(2), 76–82.

[59] Mali, U. and Munih, M. (2006). HIFE-haptic Interface for Finger Exercise. *IEEE/ASME Transactions on Mechatronics*, **11**(1), 93–102.

[60] Marsh, J., Glencross, M., Pettifer, S., and Hubbold, R. (2006). A Network Architecture Supporting Consistent Rich Behavior in Collaborative Interactive Applications. *IEEE Transactions on Visualization and Computer raphics*, **12**(3), 405–416.

[61] McLaughlin, M., Rizzo, A., Jung, Y., Peng, W., Yeh, S., and Zhu, W. (2005). Haptics-Enhanced Virtual Environments for Stroke Rehabilitation. *Proc. IPSI*.

[62] McNeely, W., Puterbaugh, K., and Troy, J. (1999). Six Degree-of-freedom Haptic Rendering Using Voxel Sampling. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 401–408. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA.

[63] Minogue, J. and Jones, M. (2006). Haptics in Education: Exploring an Untapped Sensory Modality. *Review of Educational Research*, **76**(3), 317.

[64] Minsky, M., Ming, O.-y., Steele, O., Brooks, Jr., F. P., and Behensky, M. (1990). Feeling and seeing: Issues in force display. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 235–241, New York, NY, USA. ACM.

[65] Mirtich, B. and Canny, J. (1995). Impulse-based Dynamic Simulation. *Algorithmic Foundations of Robotics*, pages 407–418.

[66] Morris, D. (2004). Haptic Battle Pong: High-Degree-of-Freedom Haptics in a Multi-player Gaming Environment. Experimental Gameplay Workshop, GDC. In *Proceedings of the Experimental Gameplay Workshop, GDC*.

[67] Niakosari, S. and Sirouspour, S. (2009). Improving transparency in network-based haptics. pages 547–552.

[68] Niemeyer, G. and Slotline, J.-J. (1991). Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering.*, **16**(1), 152–162.

[69] Orozco, M., Asfaw, Y., Shirmohammadi, S., Adler, A., and El Saddik, A. (2006). Haptic-based Biometrics: A Feasibility Study. In *Proceedings of the 14th Symposium*

on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006., pages 265–271.

[70] Ortega, A. (2001). Haptic Data Compression. *Touch in Virtual Environments, IMSC Series in Multimedia, Prentice-Hall, New York.*

[71] Park, K. and Kenyon, R. (1999). Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment. pages 104–111.

[72] Price, W. (1977). A Controlled Random Search Procedure for Global Optimisation. *The Computer Journal*, **20**(4), 367–370.

[73] Richards, C., Rosen, J., Hannaford, B., Pellegrini, C., and Sinanan, M. (2000). Skills Evaluation in Minimally Invasive Surgery Using Force/torque Signatures. *Surgical Endoscopy*, **14**(9), 791–798.

[74] Ruspini, D., Kolarov, K., and Khatib, O. (1997). The Haptic Display of Complex Graphical Environments. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 345–352. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA.

[75] Salisbury, K., Conti, F., and Barbagli, F. (2004). Haptic Rendering: Introductory Concepts. *IEEE Computer Graphics and Applications*, **24**(2), 24–32.

[76] Sankaranarayanan, G. and Hannaford, B. (2008). Experimental internet haptic collaboration using virtual coupling schemes. pages 259–266.

[77] Shaw, C. and Green, M. (1993). The MR Toolkit Peers Package and Experiment. In *Proceedings of the IEEE Virtual Reality Annual International Symposium, 1993.*, pages 463–469.

[78] Shirmohammadi, S. and Ho Woo, N. (2004). Evaluating decorators for haptic collaboration over internet. pages 105–109.

[79] Stansfield, S., Shawver, D., Miner, N., and Rogers, D. (1995). An Application of Shared Virtual Reality to Situational Training. In *Proceedings of the Virtual Reality Annual International Symposium, 1995.*, pages 156–161.

[80] Stramigioli, S., Secchi, C., vanderSchaft, A., and Fantuzzi, C. (2005). Sampled Data Systems Passivity and Discrete Port-hamiltonian Systems. *IEEE Transactions on Robotics*, **21**(4), 574–587.

[81] Van der Linde, R., Lammertse, P., Frederiksen, E., and Ruiter, B. (2002). The HapticMaster, a New High-performance Haptic Interface. In *Proc. Eurohaptics*, pages 1–5.

[82] Walker, S. and Salisbury, J. (2003). Large Haptic Topographic Maps: Marsview and the Proxy Graph Algorithm. In *Proceedings of the 2003 Symposium on Interactive 3D Graphics*, pages 83–92. ACM New York, NY, USA.

[83] Wang, D., Tuer, K., Rossi, M., Ni, L., and Shu, J. (2003). The effect of time delays on tele-haptics. pages 7–12.

[84] Wang, X. and Fenster, A. (2004). A Haptic-enhanced 3D Real-time Interactive Needle Insertion Simulation for Prostate Brachytherapy. In *Proceedings of SPIE*, volume 5367, page 781.

[85] Webster, R., Haluck, R., Zoppetti, G., Benson, A., Boyd, J., Charles, N., Reeser, J., and Sampson, S. (2003). A Haptic Surgical Simulator for Laparoscopic Cholecystectomy Using Real-time Deformable Organs. In *Proceedings of the IASTED International Conference on Biomedical Engineering*, pages 25–27.

[86] Williams, R., Chen, M., and Seaton, J. (2003). Haptics-augmented Simple-machine Educational Tools. *Journal of Science Education and Technology*, **12**(1), 1–12.

[87] Yu, W., Ramloll, R., and Brewster, S. (2001). Haptic Graphs for Blind Computer Users. *Lecture Notes in Computer Science*, pages 41–51.

[88] Zilles, C. and Salisbury, J. (1995). A Constraint-based God-object Method for Haptic Display. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots'.*, volume 3.