

Multi-rate Control Architectures for
Network-based Multi-user Haptics Interaction

MULTI-RATE CONTROL ARCHITECTURES FOR
NETWORK-BASED MULTI-USER HAPTICS INTERACTION

BY

MAHYAR FOTOOHI GHIAM, 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

© Copyright by Mahyar Fotoohi Ghiam, December 2006

All Rights Reserved

Master of Applied Science (2006)
(Electrical & Computer Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE: Multi-rate Control Architectures for Network-based
Multi-user Haptics Interaction

AUTHOR: Mahyar Fotoohi Gham
B.Sc., (Electrical Engineering)
Shiraz University, Shiraz, Iran

SUPERVISOR: Dr. Shahin Sirouspour and Dr. David Capson

NUMBER OF PAGES: xiii, 103

To Hamid and Shahin

Abstract

Cooperative haptics enables multiple users to manipulate computer simulated objects in a shared virtual environment and to feel the presence of other users. Prior research in the literature has mainly addressed single user haptic interaction. This thesis is concerned with haptic simulation in multi-user virtual environments in which the users can interact in a shared virtual world from separate workstations over Ethernet-based Local Area Networks (LANs) or Metropolitan Area Networks (MANs). In practice, the achievable real-time communication rate using a typical implementation of network protocols such as the UDP and TCP/IP can be well below the 1kHz update rate that is suggested in the literature for high fidelity haptic rendering. However by adopting a multi-rate control strategy as proposed in this work, the local control loops can be executed at 1kHz while the data packet transmission between the user workstations occur at a lower rate. Within such a framework, two control architectures, namely centralized and distributed are presented. In the centralized controller a central workstation simulates the virtual environment, whereas in the distributed controller each user workstation simulates its own copy of the virtual environment. Two different approaches have been proposed for mathematical modeling of the controllers and have been used in a

comparative analysis of their stability and performance. The results of such analysis demonstrate that the distributed control architecture has greater stability margins and outperforms the centralized controller. They also reveal that the limited network transmission rate can degrade the haptic fidelity by introducing viscous damping into the virtual object perceived impedance. This extra damping is compensated by active control based on the damping values obtained from the analytical results. Experimental results conducted with a dual-user/dual-finger haptic platform are presented for each of the proposed controller under various scenarios in which the user workstations communicate with UDP protocol subjected to a limited transmission rate. The results demonstrate the effectiveness of the proposed distributed architecture in providing a stable and transparent haptic simulation in free motion and in contact with rigid environments.

Acknowledgements

I am grateful to my supervisors Dr. Shahin Sirouspour and Dr. David Capson for providing me the support and guidance to pursue this research work. I want to thank Dr. Sirouspour for his tremendous effort and encouragement during all the stages of my M.A.Sc. program.

Sincere thanks go to my colleagues at the Telerobotics, Haptics and Computational Vision Lab for their kind help and friendship. I am also grateful to my friends for their kindness and help.

Last but by no means least, I'm deeply thankful to my family for they endless support, love and encouragement through out my life. Their understanding and patience have made it possible to pass all the steps. Without their support completion of this thesis would not have been possible.

Contents

| | |
|--|-----------|
| Abstract | iv |
| Acknowledgements | vi |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Problem Statement | 2 |
| 1.3 Thesis Contributions | 5 |
| 1.4 Organization of the Thesis | 6 |
| 1.5 Related Publications | 7 |
| 2 Literature Review | 8 |
| 2.1 Haptic devices | 8 |
| 2.2 Single-user Haptic Interaction | 11 |
| 2.2.1 Force Calculation | 11 |
| 2.2.2 Friction Force Calculation | 13 |
| 2.2.3 Stability Analysis | 14 |
| 2.2.4 Multi-rate Haptic Control | 14 |
| 2.2.5 Haptic Applications | 15 |

| | | |
|----------|---|-----------|
| 2.3 | Multi-user Haptic Interaction | 17 |
| 3 | Modelling of Single-user Haptic Interaction | 20 |
| 3.1 | Continuous-time Dynamics of Haptic Interaction | 20 |
| 3.2 | System Discretization | 27 |
| 4 | Cooperative Haptics: Architectures and Modeling | 30 |
| 4.1 | Architectures for Cooperative Haptics | 31 |
| 4.1.1 | Centralized Cooperative Haptics | 31 |
| 4.1.2 | Distributed Cooperative Haptics | 32 |
| 4.2 | Modelling of Multi-rate Cooperative Haptics | 34 |
| 4.2.1 | Subsystem Resampling | 35 |
| 4.2.2 | Direct State-space Representation | 43 |
| 5 | Stability Analysis | 51 |
| 5.1 | Centralized Controller | 52 |
| 5.2 | Distributed Controller | 53 |
| 6 | Performance Analysis | 56 |
| 6.1 | Free Motion | 56 |
| 6.2 | Rigid Contact | 62 |
| 7 | A Platform for Two-finger Haptic Interaction in Cooperative Virtual Environments | 66 |
| 7.1 | Haptic Device | 67 |
| 7.2 | Virtual Environment | 68 |
| 7.2.1 | Collision Detection | 68 |

| | | |
|----------|---|-----------|
| 7.2.2 | Force Calculation | 69 |
| 7.2.3 | Virtual Environment Dynamics | 73 |
| 7.3 | Graphics | 73 |
| 7.4 | Real-time Control Software | 73 |
| 7.5 | Network Communication | 74 |
| 8 | Experimental Results | 75 |
| 8.1 | Stability | 75 |
| 8.2 | Performance | 77 |
| 9 | Conclusions and Future Work | 81 |
| 9.1 | Conclusions | 81 |
| 9.1.1 | Suggestions for Future Research | 82 |
| A | State Transition Matrices for Discrete-time Multi-rate Dynamics with ZOH | 84 |
| B | Viscous Damping Calculation Using the Original Model | 87 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Experimental setup to measure communication channel characteristics for two remote workstations on an Ethernet-based LAN. | 3 |
| 2.1 | An impedance control scheme for single-user haptic interaction. | 9 |
| 2.2 | (a) Phantom from Sensible. (b) Wand from Quanser. (c) Delta form Force Dimension. (d) CyberForce from Immersion | 10 |
| 2.3 | Penalty-based approach for force calculation | 12 |
| 3.1 | Single-user haptic: (a) control architecture, (b) single-axis model. | 21 |
| 3.2 | Block diagram of continuous-time system with discrete-time feedback loop. The continuous-time system is discretized using ZOH approximation explained in the next section. | 22 |
| 3.3 | State-space presentation of single-user haptic interaction (a) Continuous system with ZOH feedback, (b) Discretized system, (c) Closed loop system. | 27 |
| 4.1 | Centralized control architecture. | 31 |
| 4.2 | Distributed control architecture. | 32 |
| 4.3 | Model of centralized single-axis cooperative haptics. | 33 |
| 4.4 | Model of distributed single-axis cooperative haptics. | 34 |

| | | |
|-----|---|----|
| 4.5 | Block diagram representation of multi-rate feedback control system in subsystem resampling approach. | 35 |
| 4.6 | Block diagram illustration of resampling approach for centralized architecture: (a) local and remote user are running under two different sample rates. (b) local user is resampled to the slower sample rate. (c) The interconnection is closed and the system is reduced to a MIMO/single-rate system. | 36 |
| 4.7 | Block diagram illustration of resampling approach for distributed architecture: (a) First and second user are running under the fast computation rate. (b) Both subsystems are resampled at the slower sample rate. (c) The interconnection is closed and the system is reduced to a MIMO/single-rate system. | 41 |
| 4.8 | Block diagram representation of Multi-rate feedback control system in state-space approach. | 43 |
| 5.1 | The region of stability for the centralized control architecture for different communication channel delays. | 52 |
| 5.2 | The region of stability for the distributed control architecture when $k_{o1} = k_{o2} = 1000$ N/m are fixed for different communication channel delays. | 53 |
| 5.3 | The region of stability for the distributed control architecture when $k_{12} = k_{21} = 300$ N/m are fixed for different communication channel delays. | 54 |

| | | |
|-----|--|----|
| 5.4 | The region of stability for the distributed control architecture when $k_{11} = k_{22} = 2000$ N/m are fixed for different communication channel delays. | 55 |
| 6.1 | Model of haptic interaction with $k_w = 30000$ N/m and $b_w = 20$ N.s/m in hard contact and $b_w = k_w = 0$ in free motion: (a) Local user in centralized controller; (b) Remote user in centralized controller; (c) Distributed controller; (d) The ideal system. | 57 |
| 6.2 | Object perceived admittance $\frac{V_i^h}{F_i^h}(j\omega)$ in free motion. | 58 |
| 6.3 | Simplified model of distributed architecture in which the user force is directly applied to the object. | 60 |
| 6.4 | The user perceived admittance of the virtual object in the distributed controller before and after compensation compared with that of a pure mass. | 63 |
| 6.5 | The perceived environment admittance $\frac{X_i^h}{F_i^h}(j\omega)$ when the box is in contact with a rigid wall. | 64 |
| 7.1 | The experimental setup. | 67 |
| 7.2 | Basic elements for single-finger haptic rendering. | 68 |
| 7.3 | Interference test for collision detection. | 69 |
| 7.4 | Contact states and their transitions. | 71 |
| 8.1 | Comparison of distributed user and remote user in centralized architectures in experiment: (a) finger-box interaction force; (b) box's path on the x-y plane. | 76 |
| 8.2 | The virtual environment in performance evaluation experiments: (a) free motion; (b) rigid contact; grey cylinders represent the user fingers. 78 | |

| | | |
|-----|--|----|
| 8.3 | The net force exerted on the object for free motion along a sinusoidal path. | 79 |
| 8.4 | Finger position when the user pushes the object against a stiff wall. . | 80 |

Chapter 1

Introduction

1.1 Motivation

Haptic interaction in shared virtual environments (VEs) is an emerging area of research with promising applications [1,2]. These include but are not limited to the training of surgical tasks [3,4], haptic-enabled rehabilitation [5], teaching writing skills to children [6], sports training, as well as gaming and entertainment. For instance, a cooperative haptic simulator can permit an experienced surgeon to virtually guide a student trainee through various surgical procedures by providing corrective force feedback in real time. In such a scenario, the trainer and trainee may use identical haptic interfaces to collaboratively operate on a common virtual patient. In gaming applications, new series in group games can be developed by adding force feedback capability to multi-player games. The experimental study by Basdogan et al. in [7] has shown that the addition of haptic communication to visual feedback would significantly improve the sense of togetherness and task performance in shared virtual environments.

In its simplest form, cooperative haptics can involve multiple users with their haptic interfaces directly connected to a single host computer running the simulation. This streamlines the control design but is only feasible for a very limited number of users at the same workstation. Arguably, the more promising applications of shared virtual environments are in network-based haptics where the users can interact across communication links such as Ethernet-based Local Area Networks (LANs), Metropolitan Area Networks (MANs) or more broadly Wide Area Networks (WANs). Such configurations remove physical barriers and permit users to interact over long distances. The problem, however, is that network-based data communication is generally nondeterministic and suffers from delay, jitter, packet loss, and limited packet transmission rate. These can all adversely affect the performance and stability of cooperative haptics and pose a serious control design challenge.

1.2 Problem Statement

Due to the above-mentioned limitations and in particular relatively long nondeterministic delays, achieving a robustly stable high-performance haptic interaction over a public communication medium such as the Internet is immensely difficult with the current technology. Instead, the focus of the present study is on cooperative haptic simulation over LANs and high-speed MANs. Such fast networks can cover a single building, several buildings across a university campus, and even geographical areas as large as cities. They provide small and large companies, universities, hospitals, and other organizations with a fast and reliable means of communication. The target applications of this work are primarily in cooperative

| | |
|------------------------|--------------------------------------|
| Ave. Round-trip delay | 2.4395 ms |
| RMS Jitter | 0.4871 ms |
| Packet loss | 2 packets out of 2.7 million packets |
| Achievable packet rate | 1/128 s |

Table 1.1: Characteristics of a mixed 100Mbps/1Gbps LAN under normal traffic. The experiment was carried out on two VxWorks workstations communicating with UDP protocol over a period of 36 minutes.

haptic-enabled training. For instance, in hospitals or medical schools connected via LANs or MANs, students and expert surgeons can haptically interact in training sessions without all being present at the same location.

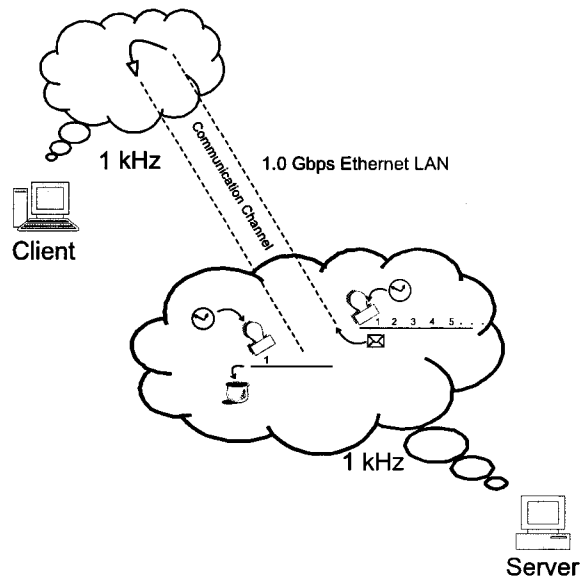


Figure 1.1: Experimental setup to measure communication channel characteristics for two remote workstations on an Ethernet-based LAN.

In Table 1.1, the results of a network communication experiment between two PCs running the VxWorks real-time operating system with the UDP protocol over a mixed 100Mbps/1Gbps LAN are presented (see Fig. 1.1). The computers in this

experiment were located at two separate locations in the Information Technology Building at McMaster University with a 1Gbps Ethernet network. However, one of the switches in the path of the workstations and the computer network cards were based on the 100Mbps standard. In the experiment, packets of incrementally increasing integer numbers tagged with the transmission time were sent from the first to the second computer and were immediately relayed back to the first computer. Upon their arrivals, packets were also tagged with the arrival time and stored for analysis. The entire experiment lasted about 36 minutes. Although some of the numbers in table 1.1 may change depending on the network traffic condition, they are representative of the typical characteristic of a LAN and to some extent a fast MAN. The data suggest that for the applications considered in this thesis, the data packet loss and jitter are negligible and the network communication delay between the user workstations is a fraction of the network sampling time. However as can be seen in the table, network protocols such as the UDP/IP and TCP/IP have limited update rates for real-time transmission of data packets. This limitation must be distinguished from that of the network bandwidth as in haptic applications the size of data packets is often small. The packet transmission rate can affect haptic fidelity, particularly in rendering of rigid objects and contacts. In our experience, the update rate of 1kHz for real-time control of haptic interfaces suggested in the literature, is well above that achievable over LANs and MANs using standard UDP/IP and TCP/IP implementations with a typical traffic load.

There have been prior attempts at achieving high-rate data communication for real-time control over an Ethernet connection. For instance in [8], a streamlined implementation of the UDP/IP has been reported for high-rate communication

between a haptic device running on an embedded controller and a host workstation through a dedicated Ethernet link. However, it is unclear whether such an approach would be applicable to a general Ethernet-based LAN or MAN with several general-purpose workstations that host the haptic devices in a cooperative environment. Obviously, attaining a high-rate real-time communication over the Internet is even more demanding. The development of new protocols for reliable high-packet-rate communication over computer networks suitable for real-time control will likely eliminate existing limitations in future. Nevertheless in the meantime, control algorithms for cooperative haptics must be devised for improved stability and performance within the confinement of the existing protocols.

1.3 Thesis Contributions

In this thesis, the effect of a limited packet transmission rate and a relatively small communication delay on the stability and performance of a centralized and a distributed haptic control architecture is investigated. In the centralized framework, the user workstations send the haptic device positions to a server computer which detects collisions, simulates the virtual environment dynamics, and generates force-feedback signals. The results are returned to the local workstations for display to the users. The second approach is based on a distributed control architecture in which each workstation executes a local copy of the VE and simultaneously communicates with other workstations to coordinate the VEs and users. The stability margins of the two multi-rate control architectures as well as their transparency are analyzed and compared. It is shown that the downsampling and delay in the

communication link cause a viscous friction effect that can be compensated by active control. An experimental platform for dextrous dual-user/dual-finger haptic interaction is developed and used to validate the analytical results. The main contributions of this thesis can be summarized as

- Presenting a mathematical framework for the modelling and analysis of cooperative multi-rate haptic control systems for rigid object manipulation.
- Comparative study of stability and performance of proposed centralized and distributed architectures for cooperative haptics using the developed models.
- Proposing a method for compensating the undesirable effect of network down-sampling on cooperative haptic rendering.

1.4 Organization of the Thesis

The thesis is organized as follow. The haptics literature will be reviewed in Chapter 2. Mathematical modelling of single-user haptic interaction is proposed in Chapter 3. The control architectures and mathematical modelling of multi-rate cooperative haptics will be discussed in Chapter 4. Results of stability analysis for the centralized and distributed control architectures will be presented in Chapter 5. These will be followed by a transparency analysis in Chapter 6. In Chapter 7, a dextrous cooperative haptic platform will be introduced. Experimental results will be given in Chapter 8. The thesis will be concluded in Chapter 9 with a discussion of possible directions for future work.

1.5 Related Publications

- M. Fotoohi, S. Sirouspour and D. Capson, "Multi-rate control architectures for dextrous haptic rendering in cooperative virtual environment", to appear in the Proceedings of the IEEE Conference on Decision and Control, December 2006, San Diego, USA.
- M. Fotoohi, S. Sirouspour and D. Capson, "A Multi-rate Control Approach to Haptic Interaction in Multi-user Virtual Environments", submitted to the IEEE International Conference on Robotics and Automation, 2007.
- M. Fotoohi, S. Sirouspour and D. Capson, "Stability and performance analysis of centralized and distributed multi-rate control architectures for multi-user haptic interaction" International Journal of Robotics Research (under review).

Chapter 2

Literature Review

The haptics literature contains several haptic rendering control algorithms, methods for their stability analysis and their applications. In the following sections a brief review of the literature will be presented.

2.1 Haptic devices

Haptic devices have been widely used in teleoperation applications in which a user can remotely operate a robot and interact with an environment while receiving force feedback [9]. One may think of haptic simulation system as a teleoperation framework in which the slave robot and environment are virtual, i.e. generated based on computer models. Haptic devices are instrumented/actuated as robotic arms customized for interaction with human anatomical limbs.

Haptic devices enable the user to interact with a computer simulated object by applying opposing forces in up to 6 degrees of freedom (DOF) to the user's

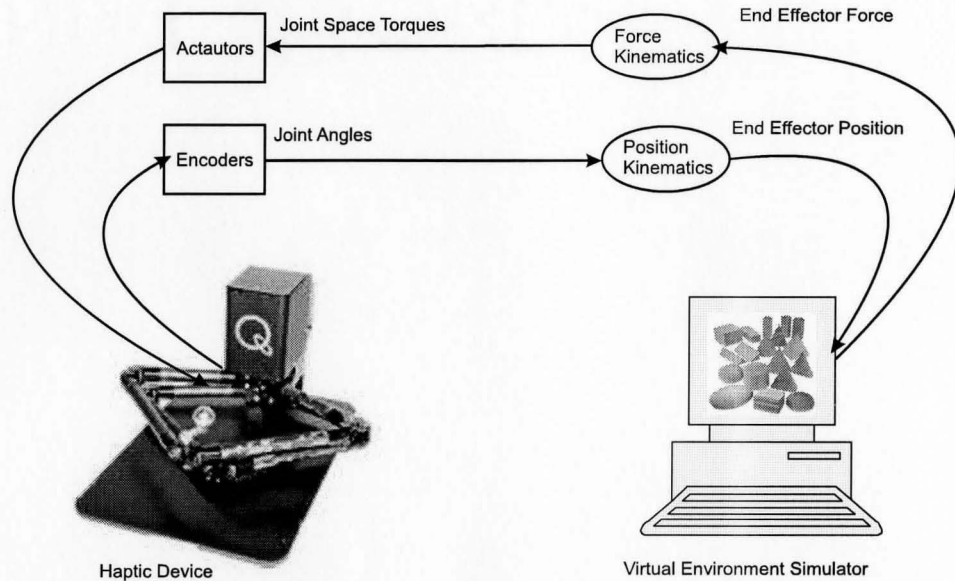


Figure 2.1: An impedance control scheme for single-user haptic interaction.

hand. Fig. 2.1 shows a general scheme for the impedance control of a haptic interface. As shown, the haptic device output is joint angle measurements and its input is the torque to be applied to the active joints. The device position kinematics routine maps the measured joint variables to the end-effector position in haptic device workspace. This represents the user hand position in the virtual world. The kinematics depend on the haptic device geometry and can be derived using conventional methods such as the Denavit-Hartenberg techniques [10]. The virtual environment simulator runs the virtual environment dynamics and calculates the reaction force that the user should experience based on the virtual world model and the user hand position in the workspace. A force kinematics routine maps the workspace force to corresponding joint space torque values applied to the joint actuators (e.g. DC motors) such that the user holding the device would feel the reaction force

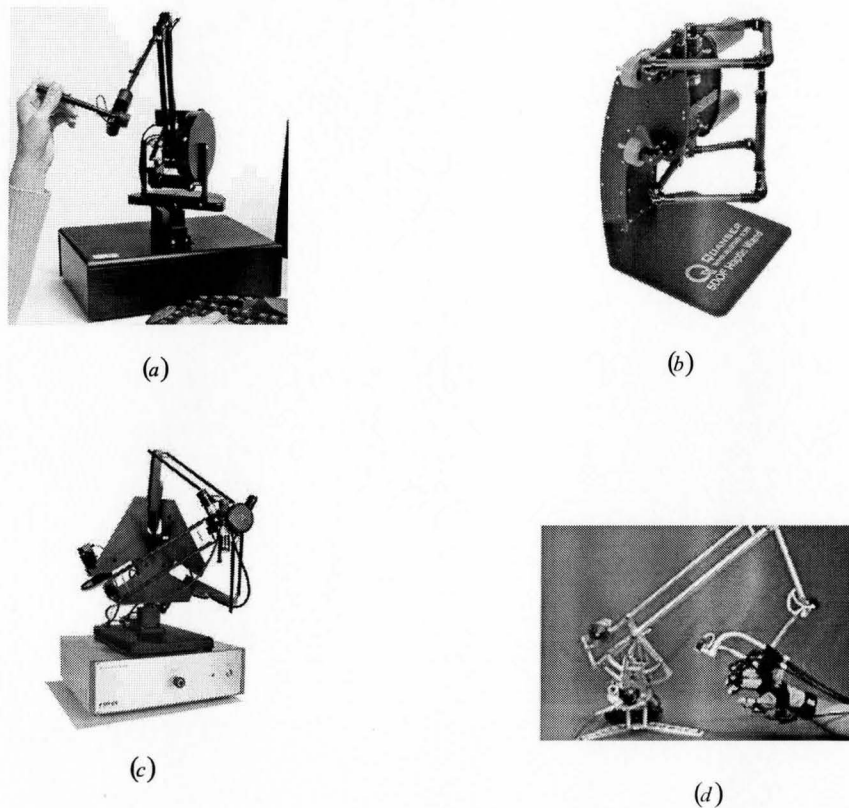


Figure 2.2: (a) Phantom from Sensible. (b) Wand from Quanser. (c) Delta form Force Dimension. (d) CyberForce from Immersion

Several haptic devices are commercially available. These are mainly designed based on the requirements of generic haptic applications but in some cases are tailored for specific applications. Hayward et al. [11] have a survey on design components, distinct properties and application of current haptic devices. Phantom (Fig. 2.2(a)) is a popular single-point interaction device used in research and commercial applications. The capstan drive mechanism enables the user to apply high forces without noticeable backlash or friction [12]. Parallel linkages have been used in the Wand (Fig. 2.2(b)) and some planar haptic devices such as the

Pantograph [13, 14]. Parallel arms are capable of applying large force while bearing simple structure and low friction. Also, the capstan mechanism has been used in combination with a parallel structure in Delta (Fig. 2.2(c)). Joystick force feedback [15] and rotatory knob force feedback devices [16] have been made for several applications in gaming or automobile industries.

In the real world, the human-object interaction happens at more than one contact point. Therefore, to attain a more realistic haptic simulation, multi-finger haptic devices are required. Haptic gloves are a more intricate field of haptic devices addressed in [17–20]. Such haptic gloves have one or more force degrees of freedom per finger with forces grounded in the palm or on the back of the hand and a virtual hand immerses the users in the virtual environment (see Fig. 2.2(d)).

2.2 Single-user Haptic Interaction

A large body of research in modelling of virtual environments and control of haptic interfaces has been dedicated to single-user applications. These efforts have already yielded numerous haptic rendering techniques and stability results for such applications.

2.2.1 Force Calculation

In virtual environments, a haptic interface point (HIP) or avatar plays the role of human limbs/tool in real world. Once the HIP is in contact with virtual object, the virtual reality simulator is responsible for applying proper force, giving the user the sensation of interacting with a real object. Penalty-based methods have been

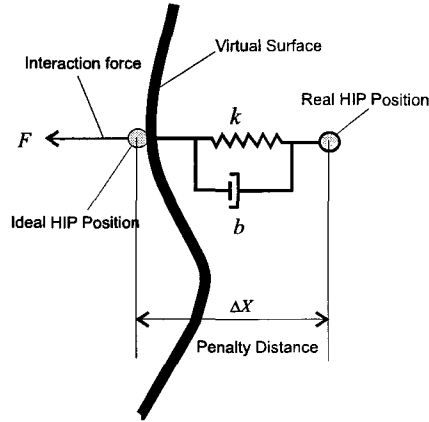


Figure 2.3: Penalty-based approach for force calculation

commonly used for rigid object force calculation. In this method applied forces are proportional to the amount of penetration into the virtual object. As shown in Fig. 2.3, one possible approach is using a spring-damper coupler to convert the penetration value to the interaction force. In this case the force is calculated from the following equation.

$$F = k\Delta X + b\Delta \dot{X} \quad (2.1)$$

For simple shapes such as sphere and cube the value and direction of the force can be easily found. For general shapes, Massie and Salisbury utilized penalty-based technique for a global force calculation algorithm. They divided the inner volume into subvolumes and related each subvolume with a surface which exerts repulsive forces [12]. The same method has also been used to render haptic interactions with volumetric data in [21, 22]. The concept of god object and proxy are introduced in [23] and [24] both based on the same principle; the algorithms

track the physical contact on the surface while HIP has penetrated into the object. A spring between the surface contact point and HIP creates the interaction force. A different approach has been taken in [25] for force rendering of rigid contacts based on infinite stiffness upon contact and limited stiffness during contact. The proposed algorithm does not increase the kinetic energy of the virtual body and improves the contact stability compared to classic methods such as the penalty method. A technique called "ray-based haptic rendering" proposed by Basdogan et al. takes into account haptic instrument shape [26]. The haptic interface is approximated as a line and the collision detection and force calculation are discussed. They have used a shading technique to provide a smooth interaction force. Sirouspour [13] et al. adopted a four-channel teleoperation framework, augmented with adaptive damping for stable transparent haptic rendering.

2.2.2 Friction Force Calculation

Friction modeling is essential for a realistic haptic display. Chen et al. [27] rendered friction and compliance based on the bristle model of friction [28], in which virtual bristles are attached to surfaces and break to a lower strain as a function of relative velocity. Armstrong and Hayward [29] have successfully used a Dahl-based model [30] to simulate friction on a haptic device. In [31] the Karnopps friction model [32] has been employed to simulate friction forces including Coulomb and viscous friction. Haptic texturing has been discussed in [33–36] with the aim of generating texture using mathematical functions. These functions usually return the height of virtual surface and its gradient as a function of HIP position.

2.2.3 Stability Analysis

One of the problems in controlling haptic interfaces is the presence of a human being in the control loop. In practice, it is difficult to propose a simple model for human hand dynamics. To address this problem, researchers have largely used passivity-based techniques to guarantee the stability of the loop consisting of human, haptic device and virtual environment. Colgate and Schenkel [37] and Miller et al. [38] have used the idea of passivity to derive fixed parameter virtual couplings that guarantee stable haptic interaction. Anderson and Spong [39] and Neimeyer and Slotine [40] utilized the passivity idea in the related area of stable control of force-feedback teleoperation with time delay. In [41] the passivity theorem is used to design an update strategy for calculating the interaction force between tool and deformable tissue to ensure stability and fidelity of the operation. Adams and Hannaford proposed an algorithm that guarantees the stability of a haptic interface coupled with a passive virtual environment [42]. Consequently, the haptic control algorithm can be designed independent of the virtual environments dynamic. Time domain passivity and virtual coupling have also been used for stability analysis of haptic control systems in [43–45]. It should be mentioned that in the passivity approach the haptic device parameters are not required for control design, at the expense of a conservative stability analysis.

2.2.4 Multi-rate Haptic Control

Complex virtual environments are computationally intensive and it is difficult to run the haptic loop at suggested rate of 1kHz or above. To tackle this problem multi-rate frameworks have been suggested in literature. In [46] a local model

bridges the discrepancy between the high frequency haptic simulation rate and the slowly run physical model to ensure a reliable haptic rendering. Also Barbagli et al. employ the concept of local models for multi-rate haptic rendering of complex deformable objects that can be touched by multiple users [47]. In [48] a multi-rate wave transformer is adopted to stabilize the zero-order-hold delay in a multi-rate control scheme for haptic simulation.

2.2.5 Haptic Applications

Recent advances in computer technology have enabled high-fidelity haptic simulations and have expanded haptic applications to several fields. For example, there has been a growing interest in the medical and computer science fields around the simulation of medical procedures.

Due to the deformable nature of human organs, several techniques have been developed to model soft tissues. Spring models have been extensively used for simulating the elasticity of soft tissue. The mass-spring model represents an object as points of mass joined by springs [49,50]. Waters [51] has defined springs on regular lattices for modeling facial tissue. Similarly, Delingette et al. [52] represented fat-tissue elasticity as a network of springs on a three-simplex mesh. Kuhnappel et al. [53,54] have implemented mass-spring models in their KISMET simulation system. They have added interior pattern nodes to the surface nodes in order to model the volumetric behavior of the whole tissue. Koch et al. [55] have used the combination of finite element and mass-spring for modelling of the skin surface, with a spring model to represent the fat tissue. A Mass-spring model is easy to

implement and is computationally inexpensive compared to the alternative methods such as finite element. On the other hand, a finite element model yields a more accurate deformation and visualization compared to that of the mass-spring model [56].

Finite element models for haptic simulation of elastic tissues have been investigated in the work of Bro-Nielsen et al. [57–59] and Cotin et al. [60, 61]. In [59] the so called Fast Finite Element (FFE) model is presented; mesh condensation techniques [62] reduce the intricacy of a finite element system. In addition, a matrix-vector multiplication algorithm is introduced to attain a higher speed. In [63–65] the accuracy of soft tissue simulation is increased at the expense of using a non-linear finite element model and hence more computation load. Wu et al. [65] have used the mass lumping method to ensure a diagonal mass matrix which speeds up the real-time force calculation. The anisotropic behavior of deformable tissue is discussed in [64] and incompressibility constraints are proposed to enhance the model. The model is implemented on a liver laparoscopic surgery platform. The authors in [66] have addressed the interaction between deformable tissues, considering soft tissue collisions and real-time haptic force calculation.

Cutting elastic tissue requires mesh regeneration and changing corresponding terms in stiffness matrix. Considering the real-time requirements of haptic simulation, the problem can become challenging. Song and Reddy proposed a technique for cutting linear deformable objects defined as finite element models and applied it for a simple 2-D case [67]. Bro-Nilsen also models the cutting by eliminating

tetrahedral elements and updating the pre-calculated inverse of the stiffness matrix [68]. Tanaka et al. simulate the cutting interaction force as viscous friction between tool tip and virtual surface [69]. In [70] a hybrid model is discussed which is a combination of a modified mass-spring model and a pre-computed deformation finite element model. The model is employed to simulate cutting anatomical structures. Mahvash et al. apply the energy-based concept of fracture mechanics to display external cutting forces of a single blade in surgical procedures.

A newly addressed application for haptic is dental training [71–73]. Wang et al. [46] implemented a Phantom-based training system for dental cutting. To ensure a stable and high-fidelity haptic interaction, a local model technique is adopted within a multi-rate cutting simulation architecture and hence the haptic device is isolated from the simulation loop.

The idea of force feedback also has been used in molecular docking [74] and nanomanipulators [75,76] to add the sense of touch to visual aspect of the work. Haptic simulators have significantly contributed to enhancing surgical skills and consequently improving the outcome of surgical procedures without risking casualties. Examples of such applications are in training simulators for catheter insertion [77], lumbar puncture [78], epidural blocks [79, 80], endoscopic surgeries [81,82], laparoscopic surgeries [83,84], and prostate biopsies [85,86].

2.3 Multi-user Haptic Interaction

Multi-user haptic interaction is necessary for applications in which the interaction of each user needs to be observed by the other users. An example of such interactions is when two surgeons are operating on the same virtual patient. Another

application of shared virtual environment is in teleoperation where the combination of real and virtual manipulation improves the teleoperation performance [87]. In sports training, the instructor can provide corrective kinesthetic forces for the trainee. Adding force feedback to networked computer games can open a new dimension in multi-user video games.

Takemura and Kishino [88] implemented a virtual workspace using a head-tracking and hand gesture input devices. However, in their system a mutual exclusion prohibits the users from simultaneous manipulating of the same object. In order to compare different cooperative haptic systems, Alhalabi and Horiguchi [89] proposed an evaluation technique in which the contact force is used as performance criterion. A collaborative nanomanipulator haptic system used for interfacing atomic force microscopes has been simulated by Hadson et al. in [90]. They used a "model view controller" scheme and an "optimistic concurrent control method" to analyze and design the system.

In [91], the authors investigated three different implementations for shared haptic environments depending on how the virtual environment is manipulated by the users. Accordingly, in static applications, users can explore but not modify the shared environment. In collaborative haptics, although users can modify the environment by touching and moving objects, they cannot manipulate the same object simultaneously. Finally, cooperative environments permit simultaneous handling of shared objects by two or more users.

Shared virtual environments are often implemented over the Internet because of its existing infrastructure and global reach that make it readily accessible by a large group of users. Unfortunately, data communication over the Internet is

nondeterministic and constrained by limited transmission rate, latency, delay jitter and data loss, all of which can negatively impact the performance and stability of haptic rendering. In [92], Yoshikawa and Ueda proposed a general structure for providing force-feedback to operators without addressing the above-noted communication issues.

The effect of communication time delay and jitter on the performance of shared haptic virtual environments has been experimentally studied by previous researchers [93–95]. In [96] it has been shown that jitter has significant negative effect on cooperative virtual environments. In [97], an audio/video media adaptation was developed for transferring data in distributed VEs, and the effects of data loss and delay jitter have been experimentally investigated. In [98] and [99], model-based controllers and wave variable-based techniques have been proposed for delay compensation in multi-user haptic rendering. In [100] a linear controller and smith predictor are used to improve the consistency of dynamic objects in shared virtual environment subjected to large communication delay. In the Transatlantic project [101], two users in remote sites lift a virtual box collaboratively across the Atlantic ocean under four different experimental scenarios. Three layers of damping in combination with a predictive algorithm are used to achieve stable haptic interaction under a peer-to-peer connection with delay. Gutwin et al. [102] used a “decorator” approach as a visual cue to display the amount of communication channel delay to the users. The cursor color indicates the delay in received data and users can use this information to secure the synchronization of dynamic objects by pausing or slowing the manipulation.

Chapter 3

Modelling of Single-user Haptic Interaction

In the simplest case of haptic rendering, only one user is manipulating the virtual environment running on the haptic interface workstation. The absence of the communication channel element in this case results in a single-rate discrete control system, assuming that the virtual environment simulation can be executed at the haptic control rate. This chapter briefly describes modelling of a single-user single-rate haptic control system as a modular component of the general form of cooperative haptics that will be studied in the following chapters.

3.1 Continuous-time Dynamics of Haptic Interaction

Fig. 3.1(a) shows the control architecture and corresponding single-axis model for a single-user haptic interaction. In this figure, $x(t)$ and $f(t)$ are continuous-time position and force signals exchanged between user and haptic device; $x(k)$ and

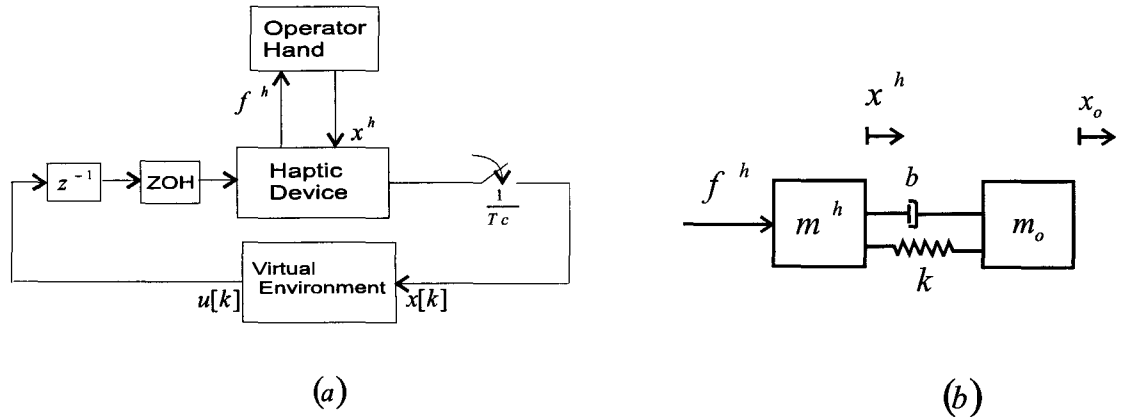


Figure 3.1: Single-user haptic: (a) control architecture, (b) single-axis model.

$u(k)$ are discrete-time position and control force signals exchanged between the haptic device and the virtual environment.

In general, the virtual environment consists of a typical haptic rendering algorithm described in Chapter 7. The user position is sampled at the control rate and provided to the haptic simulation algorithm. The haptic simulation engine calculates the user/environment interaction force and updates the virtual object states using a numerical integration routine. The discrete-time control signal is passed through a zero-order-hold (ZOH) block and is applied to the haptic device. This ZOH block and haptic simulation engine introduce one sample computation delay shown by the delay block z^{-1} .

In order to analyze the haptic architecture presented in Fig. 3.1(a), a mathematical formulation is developed based on the single-axis model of Fig. 3.1(b), assuming a penalty-based haptic simulation. Although realistic haptic simulations often involve multiple degrees of freedom, motion in these axes can be reasonably decoupled, at least in the case of interaction with rigid single-body objects and

single-point contacts. Therefore to simplify the modelling and analysis, a single-axis simulation with linear elements is considered here. The analysis could become involved if coupling among axes, e.g. due to multi-point contact between object and environment, or multi-body objects are considered. However the results of experiments in chapter 8 demonstrate that a single-axis analysis can accurately predict system behavior in multi-axis scenarios.

In Fig. 3.1(b), m^h is the operator hand and haptic device masses lumped together and b and k are virtual coupler spring and damping values which are intended to implement a penalty-based force calculation rendering. In practice, it is possible to estimate the hand/haptic device mass with an acceptable accuracy. It is assumed that the HIP is always in contact with the rigid object to avoid complexities associated with the modeling of switching control systems. As the experimental results in Chapter 8 will illustrate this assumption is a good approximation of the real system while significantly simplify the modelling and analysis.

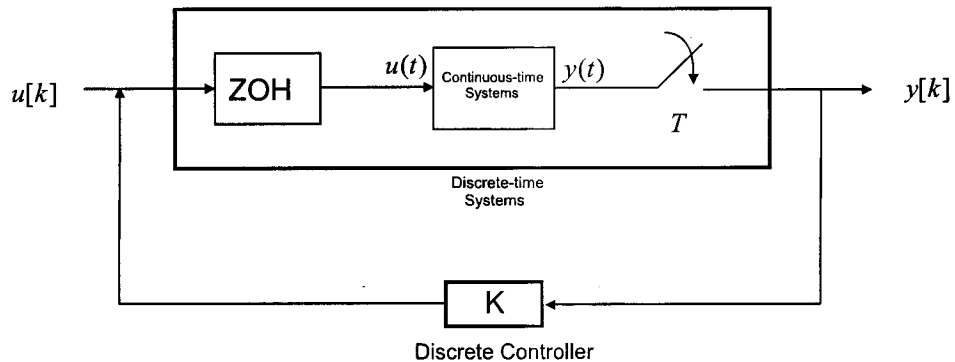


Figure 3.2: Block diagram of continuous-time system with discrete-time feedback loop. The continuous-time system is discretized using ZOH approximation explained in the next section.

In a haptic control system both the haptic/hand and the virtual object have

continuous-time dynamics. It should be mentioned that the virtual objects are usually simulated by invoking an integration routine with the goal of emulating a continuous-time system. On the other hand, the interaction between the virtual world and the operator occurs in discrete time as the user hand position is sampled and used to calculate the user/environment interaction force. Fig. 3.2 shows this discrete/continuous interconnection of the two subsystems. It is assumed that the input to the continuous-time systems is the output of a ZOH which generates a continuous time signal $u(t)$ by holding each sample value $u(k)$ constant over sampling interval T . Accordingly, the ZOH operator is defined as

$$\text{ZOH}[u(kT)] = u(kT) \quad kT \leq t \leq (k+1)T \quad (3.1)$$

The continuous-time plant dynamics are driven by the input $u(t)$ and the continuous-time output $y(t)$, usually user hand position and velocity, are sampled with sample time T . The single-user haptic interaction shown in Fig. 3.1 is composed of two continuous-time subsystems, i.e. hand and object masses and a discrete-time controller, i.e. the spring-damper virtual coupler. By applying Newton's law to the model shown in Fig. 3.1(b), one may write

$$\begin{aligned} f^h(t) - f_o(t) &= m^h \ddot{x}^h(t) \\ f_o(t) &= m_o \ddot{x}_o(t) \end{aligned} \quad (3.2)$$

The continuous time signal $f_o(t)$ is calculated using the defined operator ZOH

as follow

$$f_o(t) = \text{ZOH}[f_o(kT)] \quad (3.3)$$

where

$$f_o(kT) = k(x^h(kT) - x_o(kT)) + b(\dot{x}^h(kT) - \dot{x}_o(kT)) \quad (3.4)$$

The systems described in Eqs. (3.2) and (3.3), are linear, constant coefficient continuous systems and can be represented by a set of first order matrix differential equations in state-space form. To develop a state-space model for the proposed single-axis model of Fig. 3.1, the following subsystems are defined.

- **Hand Dynamics:** Position and velocity of the hand are chosen as states and f_o and f^h are considered as the input vector. In practice, both hand and object states are needed in order to calculate the interaction force, therefore the desired output is simply the state vector.

$$\begin{aligned} \dot{X}_h(t) &= A_h X_h(t) + B_h f^h(t) \\ y_h(t) &= C_h X_h(t) + D_h f^h(t) \end{aligned} \quad (3.5)$$

where

$$\begin{aligned} X_h &= \begin{bmatrix} x^h \\ \dot{x}^h \end{bmatrix}, A_h = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_h = \begin{bmatrix} 0 & 0 \\ \frac{1}{m^h} & -\frac{1}{m^h} \end{bmatrix} \\ C_h &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D_h = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, u_h = \begin{bmatrix} f^h \\ f_o \end{bmatrix} \end{aligned} \quad (3.6)$$

- **Object Dynamics:** In this case f_o is chosen as the input and object position and velocity as states. Hence

$$\begin{aligned} \dot{X}_o(t) &= A_o X_o(t) + B_o u_o(t) \\ y_o(t) &= C_o X_o(t) + D_o u_o(t) \end{aligned} \quad (3.7)$$

where

$$\begin{aligned} X_o &= \begin{bmatrix} x_o \\ \dot{x}_o \end{bmatrix}, A_o = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_o = \begin{bmatrix} 0 \\ \frac{1}{m_o} \end{bmatrix} \\ C_o &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, u_o = [f_o] \end{aligned} \quad (3.8)$$

and the discrete control signal $f_o(kT)$, calculated from Eq. (3.4), can be written as matrix form of

$$f_o(k) = \text{ZOH}[K (X_h - X_o)] = K \text{ZOH}[X_h - X_o] \quad (3.9)$$

where the elements of vector K consist of virtual coupler stiffness and damping.

$$K = [k \ b] \quad (3.10)$$

To derive a compact form for the above mentioned continuous-time dynamics, Eqs.(3.7) and (3.5) can be combined together. The resulting state-space dynamics are governed by

$$\begin{aligned} \dot{X}(t) &= AX(t) + Bu(t, kT) \\ y(t) &= CX(t) + Du(t, kT) \end{aligned} \quad (3.11)$$

where

$$\begin{aligned} X &= \begin{bmatrix} X_h(t) \\ X_o(t) \end{bmatrix}, \quad A = \begin{bmatrix} A_h & 0 \\ 0 & A_o \end{bmatrix}, \quad B = \begin{bmatrix} B_h(:, 1) & B_h(:, 2) \\ 0 & B_o \end{bmatrix} \\ C &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad u(t, k) = \begin{bmatrix} f^h(t) \\ f_o(kT) \end{bmatrix} \end{aligned} \quad (3.12)$$

where $B(:, i)$ denotes the i th column of matrix B . Fig. 3.3 illustrates the continuous-time system and its discrete feedback loop. Due to the presence of the ZOH elements, continuous-time analysis of the above system is rather cumbersome. Instead, a discrete-time model will be developed below that will streamline subsequent stability and performance analysis.

3.2 System Discretization

To convert a continuous-time system to its discrete counterpart several discretization/interpolation methods such as zero-order hold (ZOH), first-order hold (FOH), Tustin approximation and matched poles and zeros are proposed [103]. In haptic control systems due to the presence of the ZOH elements, the ZOH-based continuous-to-discrete transformation yields an exact description of the system in discrete-time. Assume that the continuous plant dynamics are given in the general form of Eq. (3.11), the goal is to derive the discrete-time dynamics under the zero-order hold assumption. It can be shown that [103] the solution to Eq. (3.11) is:

$$X(t) = e^{A(t-t_0)}X(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (3.13)$$

where $X(t_0)$ is the initial condition of the system. It is possible to use Eq. (3.13) to obtain a discrete-time state-space formulation of the system dynamics. Assuming the input $u(t)$ passes through a ZOH, $u(\tau)$ will be constant for $kT < \tau < (k+1)T$. By letting $t = (k+1)T$, $t_0 = kT$, and simple mathematical manipulations, one may

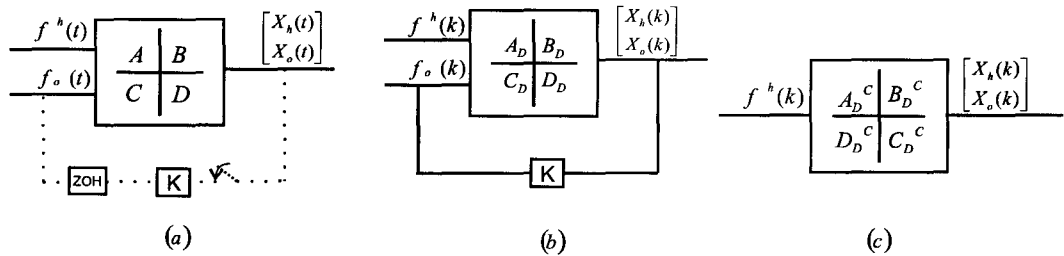


Figure 3.3: State-space presentation of single-user haptic interaction (a) Continuous system with ZOH feedback, (b) Discretized system, (c) Closed loop system.

obtain

$$\begin{aligned} X(k+1) &= A_D X(k) + B_D u(k) \\ y(k) &= C_D X(k) + D_D u(k) \end{aligned} \quad (3.14)$$

where

$$\begin{aligned} A_D &= e^{AT} & B_D &= \left(\int_0^T e^{A\eta} d\eta \right) B \\ C_D &= C & D_D &= D \end{aligned} \quad (3.15)$$

The following Taylor's series expansion is useful in evaluating the matrix exponential expression.

$$e^{AT} = I + AT + \frac{A^2 T^2}{2!} + \frac{A^3 T^3}{3!} + \dots \quad (3.16)$$

and consequently,

$$B_D = \sum_{k=0}^{\infty} \frac{A^k T^k}{(k+1)!} B \quad (3.17)$$

Replacing the continuous transition matrices from Eq. (3.11) yields the corresponding discrete-time matrices (see Fig. 3.3). In this case, the control signal f_o is a constant gain and hence, the discretized model of controller is the same as its continuous-time equivalent. Because both controller and system are discretized

with the same rate, which is the haptic simulation rate, the feedback interconnection can now be closed in discrete time (see Fig. 3.3(c)). Assuming that the subscript D denotes discretized transition matrices and superscript C denotes the closed loop transition matrices, the transition matrices for the closed loop system of Eqs. (3.11) are given by

$$A_D^C = \begin{bmatrix} A_{hD} + B_{hD}(:, 2)K & -B_{hD}(:, 2)K \\ B_{oD}K & A_{oD} - B_{oD}K \end{bmatrix} \quad (3.18)$$

$$B_D^C = \begin{bmatrix} B_{hD}(:, 1) \\ 0_{2 \times 1} \end{bmatrix}, \quad C_D^C = I_{4 \times 4}, \quad D_D^C = 0_{1 \times 4}$$

The stability of the closed system is guaranteed if and only if

$$if \quad \det(A_D - \lambda_i I) = 0 \Rightarrow |\lambda_i| \leq 1 \quad (3.19)$$

where $|\lambda_i|$ is the magnitude of λ_i .

Chapter 4

Cooperative Haptics: Architectures and Modeling

In shared virtual environments, users are connected through a network of computers that enable manipulation of objects in collaborative tasks. In the first section, two controller architectures, namely centralized and distributed, are proposed. The next section is dedicated to modelling techniques for such multi-rate multi-input/multi-output (MIMO) systems. Although the arguments presented here are based on a dual-user configuration, they can be easily extended to multi-user configurations.

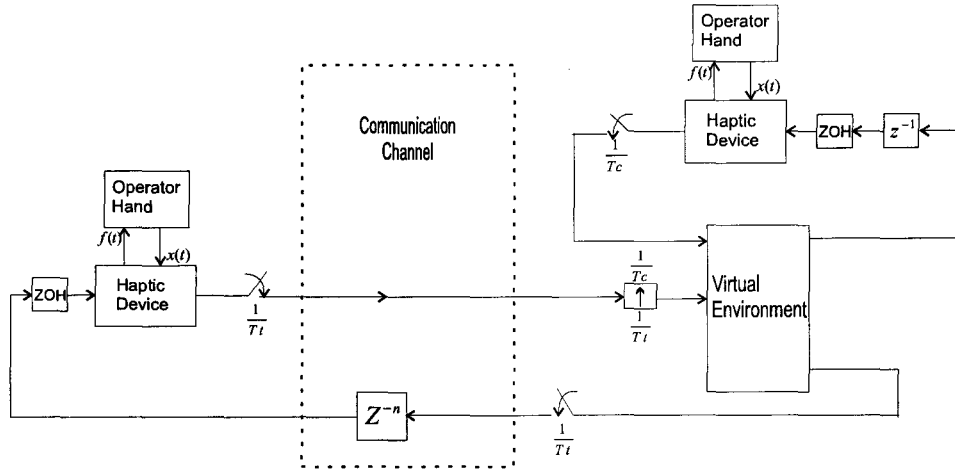


Figure 4.1: Centralized control architecture.

4.1 Architectures for Cooperative Haptics

4.1.1 Centralized Cooperative Haptics

In a centralized control architecture, a server workstation that is host to the VE simulator collects and processes the information acquired by all user workstations and returns the calculated interaction forces along with objects and other users' states for display to the users. Fig. 4.1 illustrates a centralized architecture for a dual-user cooperative haptic environment. Due to the discrete nature of the simulation, the combined network and computation delay is assumed to be a multiple of the packet communication period and is lumped in a single round-trip element represented by z^{-n} . In most LAN/MAN-based applications, $n = 1 - 2$ samples as the actual network delay is small compared to the packet transmission time interval and the delay is mainly due to the commutation cycles. In simple applications, all elements of the VE simulation can run at the high haptic update rate while the

network communication rate is limited. As will be seen later in the thesis, the maximum achievable stiffness in the centralized framework for the users across the network can be fairly restricted due to this limited network packet transmission rate.

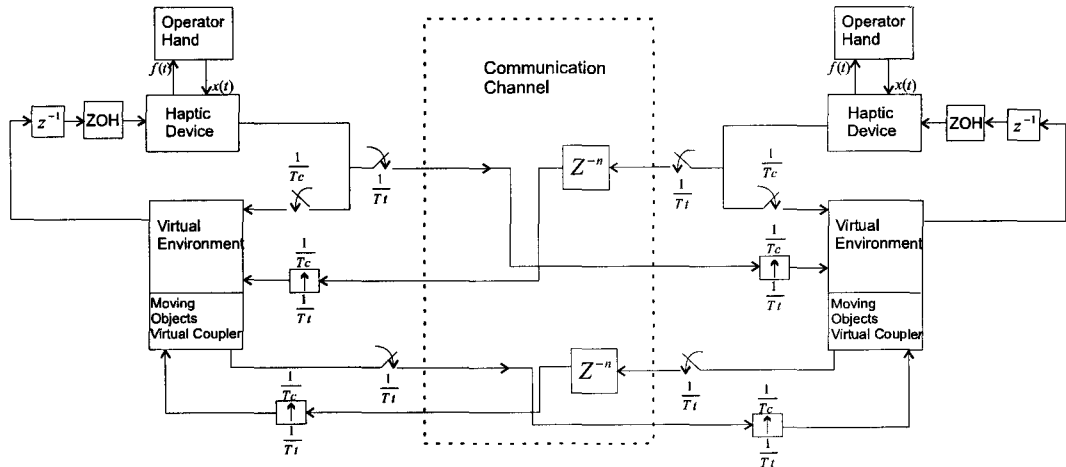


Figure 4.2: Distributed control architecture.

4.1.2 Distributed Cooperative Haptics

A distributed control architecture can mitigate the network-related stability and performance degradation of the centralized architecture at the expense of imposing higher computational power requirements⁹ on individual workstations. In this configuration, as shown in Fig. 4.2 for a two-user system, each workstation runs a complete copy of the shared virtual environment. To convey the presence of other users in the environment, each workstation receives position and velocity information of the other users as well as those of shared virtual moving objects. The

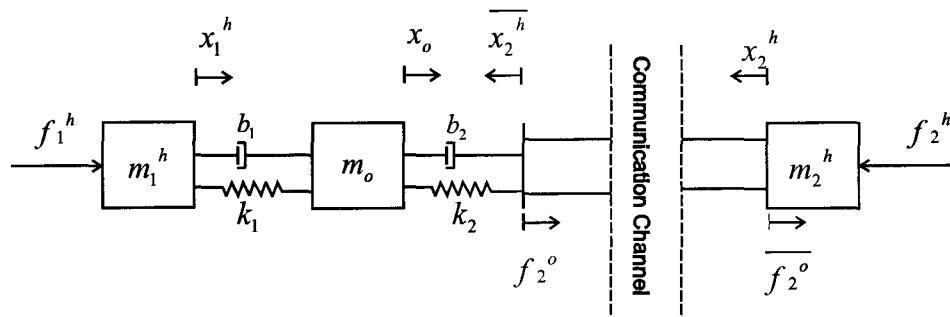


Figure 4.3: Model of centralized single-axis cooperative haptics.

corresponding objects will be connected through spring-damper type virtual couplers to avoid position drift. As in the case of centralized architecture, any data transmission over the network is subject to a limited packet communication rate and the network round-trip/computation delays are lumped as z^{-n} , with $n = 1 - 2$ network samples. The advantage of using the distributed architecture over the centralized one is rather evident in the case of contact with static rigid objects. In the distributed framework, local high-rate feedback loops allow for the rendering of rigid contacts whereas in the centralized framework, the low network packet transmission rate and delay restrict the maximum achievable contact stiffness. Some researchers have previously alluded to the potential benefit of using a distributed architecture for delay mitigation in cooperative haptics [104]. However, there has been no analytical work concerning the effect of network packet transmission rate, delay, and the control architecture on the stability and performance of LAN/MAN-based multi-rate cooperative haptic simulation.

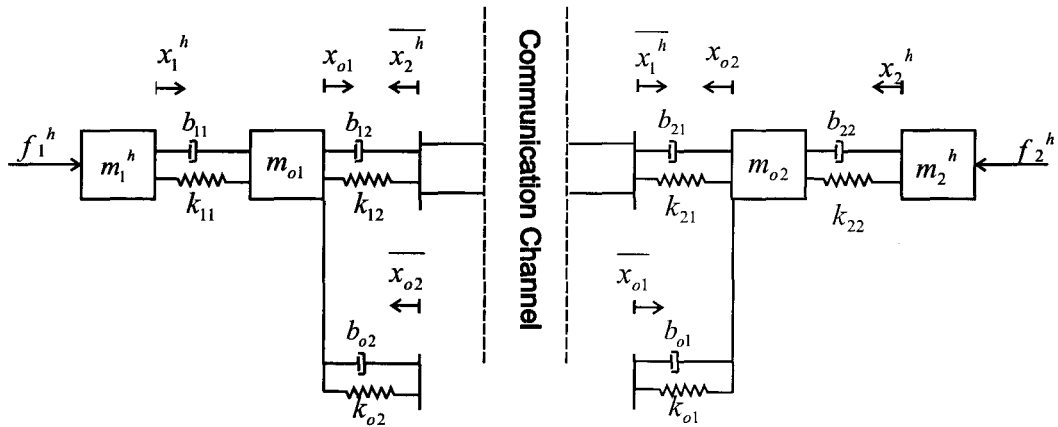


Figure 4.4: Model of distributed single-axis cooperative haptics.

4.2 Modelling of Multi-rate Cooperative Haptics

Mass-spring-damper models of the centralized and decentralized configurations are displayed in Figs. 4.3 and 4.4, respectively. In these figures, m_h^1 and m_h^2 are the combined masses of the users and haptic devices, $m_o = m_{o1} = m_{o2}$ is the mass of the virtual object, ks and bs are stiffness and damping of corresponding virtual couplers, xs and $\bar{x}s$ are local and network transmitted positions, and f_h^1 and f_h^2 are users' exogenous force inputs. The additional virtual couplers between the virtual objects in the distributed controller, represented by k_{o1} and k_{o2} in Fig. 4.4 are intended to prevent position drift between the two copies of the shared object. Note that these systems involve multi-rate discrete-time and continuous-time states due to the discrete-time nature of the controller, presence of the network element and zero-order-hold (ZOH) circuits, and the continuous-time dynamics of haptic devices. However, the modelling, analysis and synthesis will be conducted in discrete time. The following two MIMO representations of the system dynamics

in the discrete-time domain are proposed in this thesis.

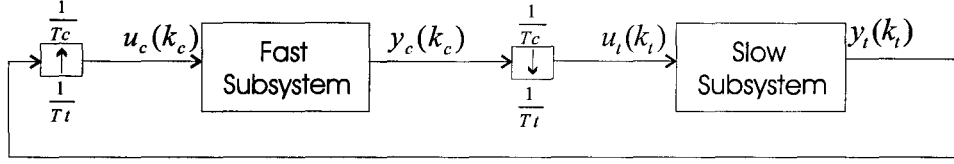


Figure 4.5: Block diagram representation of multi-rate feedback control system in subsystem resampling approach.

4.2.1 Subsystem Resampling

In this approach, first the continuous-time dynamics of the haptic devices and the virtual object are discretized using a ZOH continuous-to-discrete transformation at their corresponding sample rates as described in Chapter 3. The system dynamics including those of the controllers are then rearranged into two subsystems operating at sampling rates T_t and T_c corresponding to data transmission and control computation rates, respectively (see Fig. 4.5). It is Assumed that the samplers are synchronized and $T_t = NT_c$. The difference equations governing the evolution of the states are given by

$$X_{ij}[k+1] = A_{i,j}X_{ij}[k] + B_{ij}u_{ij}[k] \quad (4.1)$$

$$y_{ij}[k] = C_{ij}X_{ij}[k] + D_{ij}u_{ij}[k], \quad i = t, c \quad j = l, r, d$$

where the index i indicates the sampling rate of discrete system. In the succeeding analysis for the centralized controller, the users on the server and remote workstations are denoted as local (l subscript) and remote (r subscript) user, respectively. The distributed controller is symmetric with respect to the two users and denoted

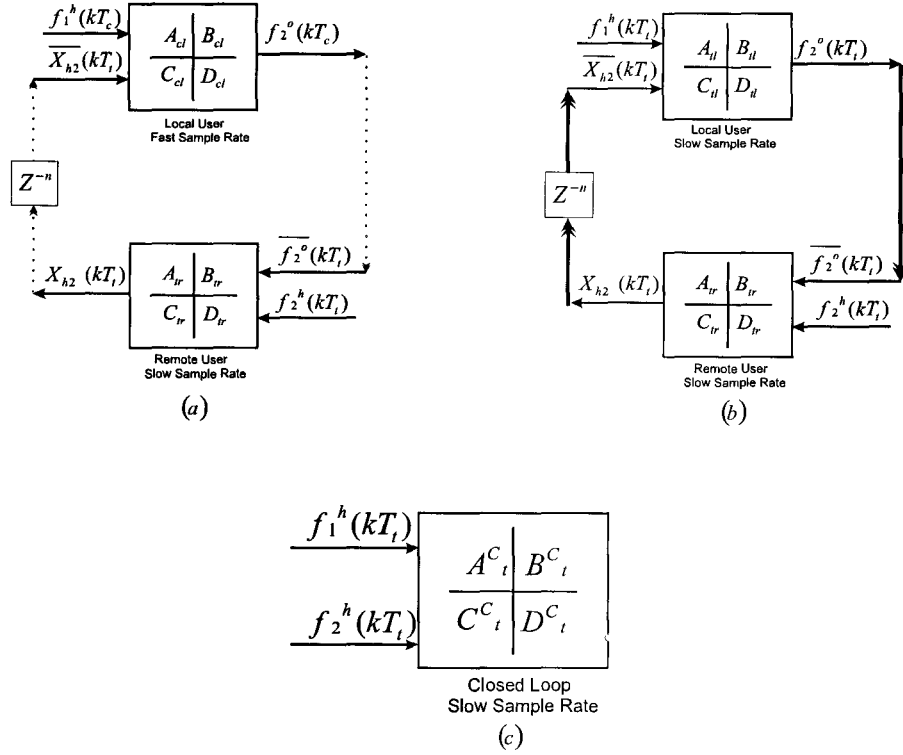


Figure 4.6: Block diagram illustration of resampling approach for centralized architecture: (a) local and remote user are running under two different sample rates. (b) local user is resampled to the slower sample rate. (c) The interconnection is closed and the system is reduced to a MIMO/single-rate system.

by subscript d . The state vectors X_{ij} contain positions and velocities of the haptic interfaces and the virtual object where applicable. The state transition matrices in Eq. (4.1) for the simplified models in Figs. 4.3 and 4.4 can be derived using the same approach discussed in Chapter 3 as follows.

- **Centralized Architecture:**

Fig. 4.6 shows the three steps required to formulate the discrete state-space

representation using a resampling technique. The local user dynamics consist of object/hand masses and two virtual couplers. As Fig. 4.3 illustrates the inputs to the system are f_1^h and the remote user states i.e. $[x_2^h \quad \dot{x}_2^h]^T = \overline{X_{h2}}$. The calculated force-feedback signal f_2^o is sent to the remote user over the communication channel and hence is subjected to limited transmission rate. Using the same ZOH system discretization approach of Chapter 3, the discrete state-space equations for local user are

$$X_{cl}((k+1)T_c) = A_{cl}X_{cl}(kT_c) + [B_{cl}(:,1) \quad B_{cl}(:,2:3)] \begin{bmatrix} f_1^h \\ \overline{X_{h2}} \end{bmatrix} \quad (4.2)$$

$$f_2^o(kT_c) = C_{cl}X_{cl}(kT_c) + [D_{cl}(:,1) \quad D_{cl}(:,2:3)] \begin{bmatrix} f_1^h \\ \overline{X_{h2}} \end{bmatrix}$$

The notation $B(:, i : j)$ represents the i th to j th columns of matrix B and the subscript c implies that the system has the fast sample time, i.e. computation sample rate. Also the subscript l refers to the local user.

The hand/mass position and velocity are chosen as states of remote user model. Note that the remote user has the communication channel sample rate which is denoted by subscribe t . The state-space equations for the remote user are (see Fig. 4.6(a))

$$\begin{aligned}
 X_{tr}((k+1)T_t) &= A_{tr}X_{tr}(kT_t) + [B_{tr}(:,1) \quad B_{tr}(:,2)] \begin{bmatrix} \frac{f_2^h}{f_2^o} \\ \frac{f_2^o}{f_2^o} \end{bmatrix} \\
 X_{h2}(kT_t) &= C_{tr}X_{tr}(kT_t) + [D_{tr}(:,1) \quad D_{tr}(:,2)] \begin{bmatrix} \frac{f_2^h}{f_2^o} \\ \frac{f_2^o}{f_2^o} \end{bmatrix}
 \end{aligned} \tag{4.3}$$

Assuming that the samplers are synchronized and $T_t = NT_c$, the fast discrete system can be resampled at the slower rate T_t . This is possible since $u_c[k_c]$ in Fig. 4.5 (or $X_{h2}(kT_t)$ in Fig. 4.6 (a)) is constant for N samples between the sampling instants of $y_t[k_t]$ (or $\overline{X_{h2}}(kT_c)$ in Fig. 4.6 (a)). It can be shown that the state transition matrices after resampling are given by

$$\begin{aligned}
 A_t &= A_c^N \\
 B_t &= A_c^{N-1}B_c + A_c^{N-2}B_c + \dots + A_c B_c + B_c \\
 C_t &= C_c \quad D_t = D_c
 \end{aligned} \tag{4.4}$$

Hence, the local transition matrices in 4.2 can be resampled using the relationships in 4.4 to the lower sample rate, i.e. transition rate.

$$\begin{aligned}
X_{ul}((k+1)T_t) &= A_{ul}X_{ul}(kT_t) + [B_{ul}(:,1) \quad B_{ul}(:,2:3)] \begin{bmatrix} f_1^h \\ X_{h2} \end{bmatrix} \\
f_2^o(kT_t) &= C_{ul}X_{ul}(kT_t) + [D_{ul}(:,1) \quad D_{ul}(:,2:3)] \begin{bmatrix} f_1^h \\ X_{h2} \end{bmatrix}
\end{aligned} \tag{4.5}$$

The computation and communication delays can also be incorporated into the state-space models by augmenting the state vectors with the delayed input signals. For one sample delay, i.e. $n = 1$, the new state transition matrices and state vector are

$$\begin{aligned}
\tilde{x}_i(k) &= \begin{pmatrix} x_i(k) \\ u_i(k) \end{pmatrix} \\
\tilde{A}_i &= \begin{pmatrix} A_i & B_i \\ 0 & 0 \end{pmatrix} \quad \tilde{B}_i = \begin{pmatrix} 0 & I \end{pmatrix} \\
\tilde{C}_i &= \begin{pmatrix} C_i & D_i \end{pmatrix} \quad \tilde{D}_i = \begin{pmatrix} 0 \end{pmatrix} \quad i = t, c
\end{aligned} \tag{4.6}$$

where 0 and I are zero and identity matrices with appropriate dimensions, respectively. At this stage, all difference equations describing the evolution of system states have the same sample rate and therefore, the feedback interconnection between the two subsystems can be closed (see Fig. 4.6(b)). Considering that the D_{tr} matrix in Eq. (4.3) is a zero matrix, and rewriting

the Eqs. in (4.3) and (4.5) results in the following state-space representation of closed loop system shown in Fig. 4.6(c).

$$\begin{aligned} X_t^C((k+1)T_t) &= A_t^C X_t^C(kT_t) + B_t^C u_t^C(kT_t) \\ y_t^C &= C_t^C X_t^C(kT_t) + D_t^C u_t^C(kT_t) \end{aligned} \quad (4.7)$$

where

$$\begin{aligned} X_t^C &= \begin{bmatrix} X_{ul} \\ X_{tr} \end{bmatrix}, \quad A_t^C = \begin{bmatrix} A_{ul} & B_{ul}(:, 2:3)C_{tr} \\ B_{tr}(:, 2:3)C_{ul} & A_{tr} + B_{ul}(:, 2:3)D_{ul}(:, 2:3)C_{tr} \end{bmatrix} \\ B_t^C &= \begin{bmatrix} B_{ul}(:, 1) & 0 \\ B_{tr}(2, :)D_{ul}(:, 1) & B_{tr}(1, :) \end{bmatrix}, \quad C_t^C = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \\ D_t^C &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad u_t^C = \begin{bmatrix} f_1^h \\ f_2^h \end{bmatrix} \end{aligned} \quad (4.8)$$

- **Distributed Architecture:**

The same procedure can be taken to model the distributed architecture. As Fig. 4.7 shows, both user simulate the virtual environment at the computation rate of T_c while the inputs $\overline{X_{h2}}(kT_t)$, $\overline{X_{h1}}(kT_t)$, $\overline{X_{o2}}(kT_t)$ and $\overline{X_{o1}}(kT_t)$ are updated at the slow rate of T_t due to the limited network packet rate. The discrete-time state-space equations of first and second users are given by

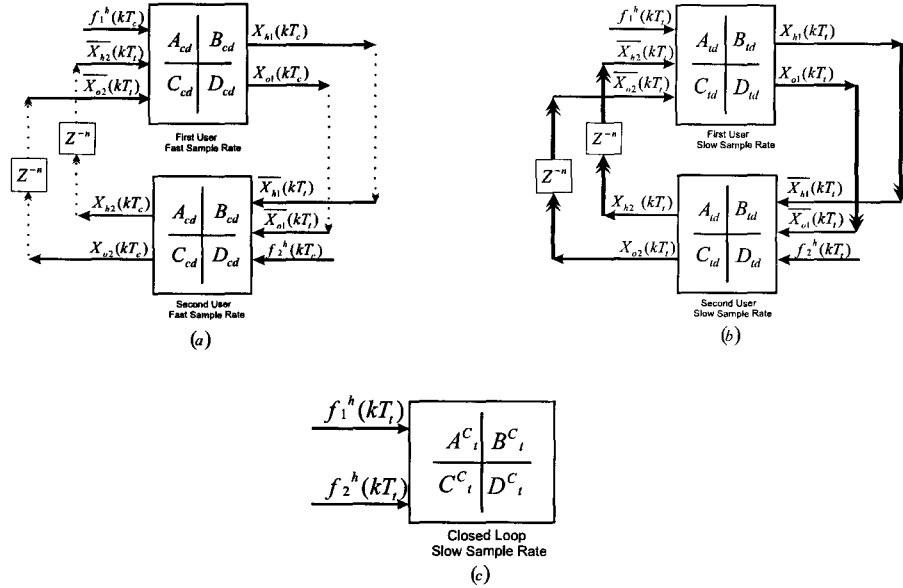


Figure 4.7: Block diagram illustration of resampling approach for distributed architecture: (a) First and second user are running under the fast computation rate. (b) Both subsystems are resampled at the slower sample rate. (c) The interconnection is closed and the system is reduced to a MIMO/single-rate system.

$$\begin{aligned}
 X_{cdm}((k+1)T_c) &= A_{cd}X_{cdm}(kT_c) + \\
 & [B_{cd}(:,1) \quad B_{cd}(:,2:3) \quad B_{cd}(:,4:5)] \begin{bmatrix} f_m^h \\ \bar{X}_{h(3-m)} \\ \bar{X}_{o(3-m)} \end{bmatrix} \\
 \begin{bmatrix} X_{hm}(kT_c) \\ X_{om}(kT_c) \end{bmatrix} &= \begin{bmatrix} C_{cd}(1:2,:) \\ C_{cd}(3:4,:) \end{bmatrix} X_{cdm}(kT_c) \quad m = 1, 2 \quad (4.9)
 \end{aligned}$$

where subscribe m denotes the first and second user. The notation $C(i:j,:)$ represents the i th to j th rows of matrix C . Note that the matrix D_{cdm} is zero in

this case because the outputs are simply the states of the system. In addition the transition matrices for both users are identical due to the symmetrical structure of the distributed architecture. The resampled transition matrices A_{td} , B_{td} , C_{td} and D_{td} (see Fig. 4.7(b)) can be computed using Eq.(4.4). The final transition matrices for the combined system after the two subsystems are connected are given by (see Fig. 4.7(c))

$$\begin{aligned} X_t^C((k+1)T_t) &= A_t^C X_t^C(kT_t) + B_t^C u_t^C(kT_t) \\ y_t^C &= C_t^C X_t^C(kT_t) + D_t^C u_t^C(kT_t) \end{aligned} \quad (4.10)$$

where

$$\begin{aligned} X_t^C &= \begin{bmatrix} X_{td1} \\ X_{td2} \end{bmatrix}, \quad A_t^C = \begin{bmatrix} A_{td} & M \\ M & A_{td} \end{bmatrix} \\ M &= B_{td}(:, 2:3)C_{td}(1:2, :) + B_{td}(:, 4:5)C_{td}(3:4, :) \\ B_t^C &= \begin{bmatrix} B_{td}(:, 1) & 0 \\ 0 & B_{td}(:, 1) \end{bmatrix}, \quad C_t^C = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \\ D_t^C &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad u_t^C = \begin{bmatrix} f_1^h \\ f_2^h \end{bmatrix} \end{aligned} \quad (4.11)$$

Despite its simplicity, this method normally is inapplicable to systems with more than two sample rates or when one rate is not a multiple of the other. Another drawback of this technique is that the design parameters, e.g. the virtual coupler

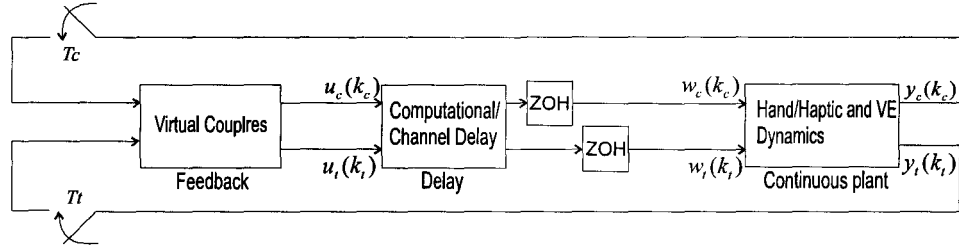


Figure 4.8: Block diagram representation of Multi-rate feedback control system in state-space approach.

parameters, cannot be separated from the rest of the system dynamics. Therefore, such a modelling approach has limited utility for control synthesis purposes.

4.2.2 Direct State-space Representation

To overcome the shortcomings of the previous method, the state-space modelling approach for multi-rate sampled systems given in [105] will be utilized here. In the rest of this section, it is demonstrated that how this technique can be employed for deriving the discrete-time model of the multi-rate cooperative haptic control system.

The control system diagram is shown in Fig. 4.8. The open-loop continuous-time model of the system, including the dynamics of the users, haptic interfaces and the virtual object, is given by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bw(t) \\ y(t) &= Cx(t)\end{aligned}\tag{4.12}$$

where $x(t) \in R^{n_0}$ is the vector of system states, i.e. positions and velocities,

$w(t) = (w_t(t), w_c(t))^T$ is the vector of inputs and $y(t) = (y_t(t), y_c(t))^T$ is the vector of measurements. The parameters in Eq. (4.12) are define below in each case.

- **Centralized Controller**

$$\begin{aligned}
 x(t) &= \begin{bmatrix} x_1^h(t) \\ \dot{x}_1^h(t) \\ x_o(t) \\ \dot{x}_o(t) \\ x_2^h(t) \\ \dot{x}_2^h(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 B &= \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{m_1^h} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{m_o} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m_2^h} \end{bmatrix}, \quad C = I_{6 \times 6}, \quad w(t) = \begin{bmatrix} f_{m_1^h}(t) \\ f_{m_o}(t) \\ f_{m_2^h}(t) \end{bmatrix}
 \end{aligned} \tag{4.13}$$

where $f_{m_1^h}(t)$, $f_{m_o}(t)$ and $f_{m_2^h}(t)$ are the net forces applied to the masses m_1^h , m_o and m_2^h , respectively.

- **Distributed Controller**

$$\begin{aligned}
 x(t) = \begin{bmatrix} x_1^h(t) \\ \dot{x}_1^h(t) \\ x_{o1}(t) \\ \dot{x}_{o1}(t) \\ x_2^h(t) \\ \dot{x}_2^h(t) \\ x_{o2}(t) \\ \dot{x}_{o2}(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.14) \\
 B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{m_1^h} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m_{o1}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m_2^h} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m_{o2}} \end{bmatrix}, \quad C = I_{8 \times 8}, \quad w(t) = \begin{bmatrix} f_{m_1^h}(t) \\ f_{m_{o1}}(t) \\ f_{m_1^h}(t) \\ f_{m_{o2}}(t) \end{bmatrix}
 \end{aligned}$$

where $f_{m_1^h}(t)$, $f_{m_{o1}}(t)$, $f_{m_{o2}}(t)$ and $f_{m_2^h}(t)$ are the net force applied to the masses m_1^h , m_{o1} , m_{o2} and m_2^h , respectively.

As before, subscripts t and c refer to the network and control sampling rates. It is assumed that all sample times are synchronized and are multiples of a base sample time t_{lcm} , i.e. $T_i = N_i t_{lcm}$, and the least common multiple of N_i 's is denoted by N_{lcm} . For the discrete-time realization of the system, an augmented state vector

is defined as

$$x_D[k] = \begin{pmatrix} x((k-1)T_{lcm} + t_{lcm}) \\ \vdots \\ x((k-1)T_{lcm} + (N_{lcm} - 1)t_{lcm}) \\ x(kT_{lcm}) \end{pmatrix} \quad (4.15)$$

with $T_{lcm} = N_{lcm}t_{lcm}$. The augmented output vector y_D is

$$y_D[k] = \begin{pmatrix} y_{D_t}[k] \\ y_{D_c}[k] \end{pmatrix} \quad (4.16)$$

where

$$y_{D_i}[k] = \begin{pmatrix} y_i(kT_{lcm}) \\ y_i(kT_{lcm} + T_i) \\ \vdots \\ y_i(kT_{lcm} + (N_i - 1)T_i) \end{pmatrix} \quad i = t, c \quad (4.17)$$

The augmented discrete input vector u_D can be defined similarly. Using the above definitions, it can be shown that [105]

$$\begin{aligned} x_D[k+1] &= A_D x_D[k] + B_D u_D[k] \\ y_D[k] &= C_D \{U_1 x_D[k+1] + U_2 x_D[k]\} \end{aligned} \quad (4.18)$$

The expressions for A_D , B_D , and C_D are given in Appendix A and U_1 , U_2 are

block diagonal matrices given by

$$\begin{aligned} U_1 &= \text{diag}(I_{n_0}, I_{n_0}, \dots, I_{n_0}, 0) \\ U_2 &= \text{diag}(0, 0, \dots, 0, I_{n_0}) \end{aligned} \quad (4.19)$$

By replacing $x_D[k+1]$ in second line of (4.18) from the first line of (4.18), one may obtain a standard state-space representation as follows

$$\begin{aligned} x_D[k+1] &= A_D x_D[k] + B_D u_D[k] \\ y_D[k] &= \hat{C}_D x_D[k] + \hat{D}_D u_D[k] \end{aligned} \quad (4.20)$$

where $\hat{C}_D = C_D U_1 A_D + C_D U_2$ and $\hat{D}_D = C_D U_1 B_D$.

The delay elements associated with computation and data transmission can be incorporated into the discrete-time model, the delayed inputs are added as new states to the state vector and the transition matrices are modified as below.

$$\begin{aligned} \tilde{x}_D(k) &= \begin{pmatrix} x_D(k+1) \\ u_D(kT_{lem} - T_i) \end{pmatrix} \\ \tilde{A}_D &= \begin{pmatrix} A_D & B_D(:,1) \\ 0 & 0 \end{pmatrix} \quad \tilde{B}_D = \begin{pmatrix} B_D(:,2\dots) & 0 \\ 0 & I \end{pmatrix} \\ \tilde{C}_D &= \begin{pmatrix} \hat{C}_D & \hat{D}_D(:,1) \end{pmatrix} \quad \tilde{D}_D = \begin{pmatrix} \hat{D}_D(:,2\dots) & 0 \end{pmatrix} \end{aligned} \quad (4.21)$$

The notation $A(:,j)$ denotes the j th column of the matrix A and $0, I$ are zero

and unit matrices with appropriate dimensions.

Once the open-loop discrete-time difference equations are obtained, the closed-loop dynamics can be formed using the feedback law

$$u_D = F_D * y_D \quad (4.22)$$

where F_D is a $p \times p$ block matrix and p is the number of distinct sample rates in the system (2 in this case). The $i - j$ th block of F_{Dij} is the N_i by N_j matrix given by

$$F_{Dij} = (f_{Dij,\mu\nu}), \quad \mu = 0, \dots, N_i - 1; \quad \nu = 0, \dots, N_j - 1 \quad (4.23)$$

$$f_{Dij,\mu\nu} = \begin{cases} f_{ij} & \nu l_j \leq \mu l_i < (\nu + 1)l_j \\ 0 & \text{otherwise} \end{cases} \quad (4.24)$$

Elements of F_D are constant and consist of the stiffness and damping parameters of all virtual couplers present in the system. In order to calculate f_{ij} elements, one may utilize the following relations for the net forces applied to the masses. (see Figs. 4.3 and 4.4)

- **Centralized Architecture:**

$$f_{m_1^h} = f_1^h - K_1(X_{h1} - X_o) \quad (4.25)$$

$$f_{m_o} = K_1(X_{h1} - X_o) - K_2(X_{h2} + X_o)$$

$$f_{m_2^h} = f_2^h - K_2(X_{h1} + X_o) \quad (4.26)$$

where

$$K_1 = [k_1 \ b_1], K_o = [k_o \ b_o], K_2 = [k_2 \ b_2] \quad (4.27)$$

- **Distributed Architecture:**

$$f_{m_1^h} = f_1^h - K_{11}(X_{h1} - X_{o1}) \quad (4.28)$$

$$f_{m_{o1}} = K_{11}(X_{h1} - X_{o1}) - K_o(X_{o1} + X_{o2}) - K_{12}(X_{h2} + X_{o1})$$

$$f_{m_2^h} = f_2^h - K_{22}(X_{h2} - X_{o2})$$

$$f_{m_{o2}} = K_{22}(X_{h2} - X_{o2}) - K_o(X_{o2} + X_{o1}) - K_{21}(X_{h1} + X_{o2})$$

where

$$K_{11} = [k_{11} \ b_{11}], K_{o1} = [k_o \ b_o], K_{22} = [k_{22} \ b_{22}] \quad (4.29)$$

$$K_{12} = [k_{12} \ b_{12}], K_{21} = [k_{21} \ b_{21}]$$

Finally, the close-loop space transition matrix A_D^c can be computed as

$$A_D^c = \tilde{A}_D + \tilde{B}_D F_D (I - \tilde{D}_D F_D)^{-1} \tilde{C}_D \quad (4.30)$$

The closed-loop system is stable if and only if all eigenvalues of this matrix lie inside the unit circle (condition in Eq. 3.19).

Chapter 5

Stability Analysis

In this chapter, the stability margins of the centralized and distributed control architectures are compared with respect to changes in the stiffness parameters of the virtual couplers. The discrete-time models of multi-rate cooperative haptic controllers are developed using the techniques introduced in Chapter 4 and their stability is analyzed based on the condition of 3.19. The values of constant parameters in all scenarios are $m_1^h = m_2^h = 0.1$ kg, $m_o = m_{o1} = m_{o2} = 0.4$ kg, and $b=10$ N.s/m for all couplers in both configurations, $T_t = 1/128$ s and $T_c = 1/1024$ s. The combined communication/computation delays of $n = 0, 1, 2, 3$ network sample times have been considered in the analysis. It should be noted that given the results of Table 1, under normal LAN/MAN conditions the combined delay value is expected to be in the range of $n=1-2$ network sample times.

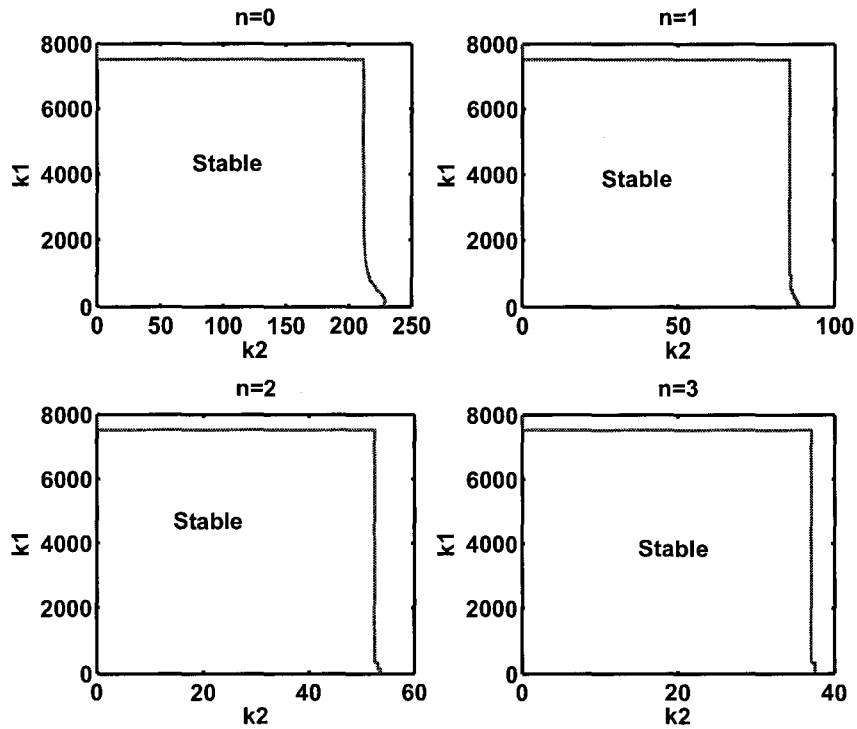


Figure 5.1: The region of stability for the centralized control architecture for different communication channel delays.

5.1 Centralized Controller

Fig. 5.1 illustrates the stability region of the centralized architecture, when all system parameter except k_1 and k_2 in Fig. 4.3 are fixed. In order to plot this figures, k_1 and k_2 are varied while the eigenvalues of matrix A_t^C in Eq. (4.8) are monitored for the stability using the condition in Eq. (3.19). The marginal values of the parameters determine the maximum stiffness that can be presented to each user. As seen in Fig. 5.1 for $n = 0$, the limited network packet rate has contributed to a significant reduction in the margin of stability with respect to k_2 , the coupling stiffness

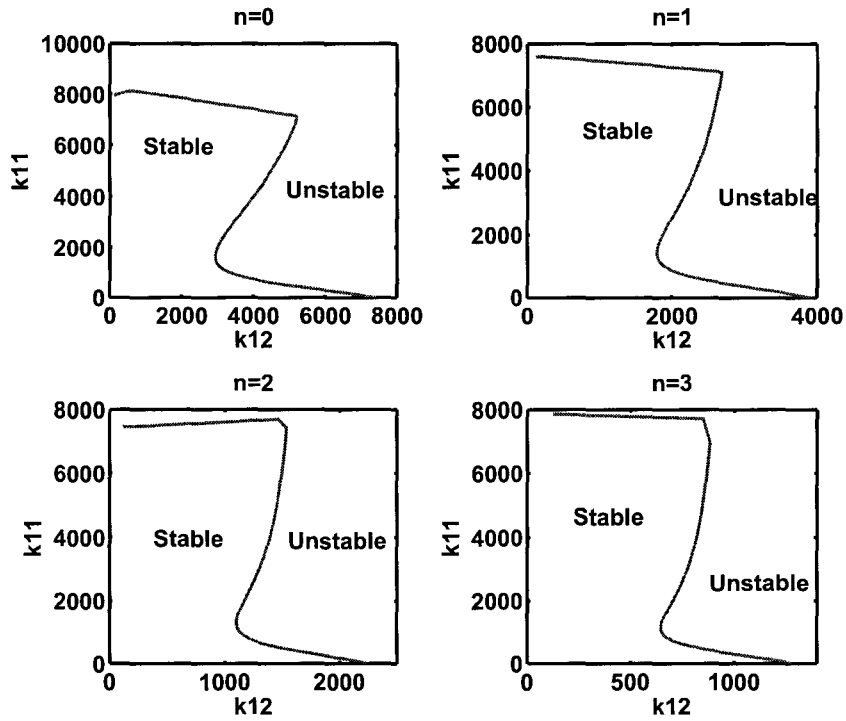


Figure 5.2: The region of stability for the distributed control architecture when $k_{o1} = k_{o2} = 1000$ N/m are fixed for different communication channel delays.

for the user across the network. As expected the marginal stiffness value for the user across the network further decreases as the n increases whereas the local user is unaffected by the communication delay.

5.2 Distributed Controller

The stability analysis was carried out for the distributed control architecture as well by checking the eigenvalues of matrix A_t^C in Eq. (4.11) for the stability using

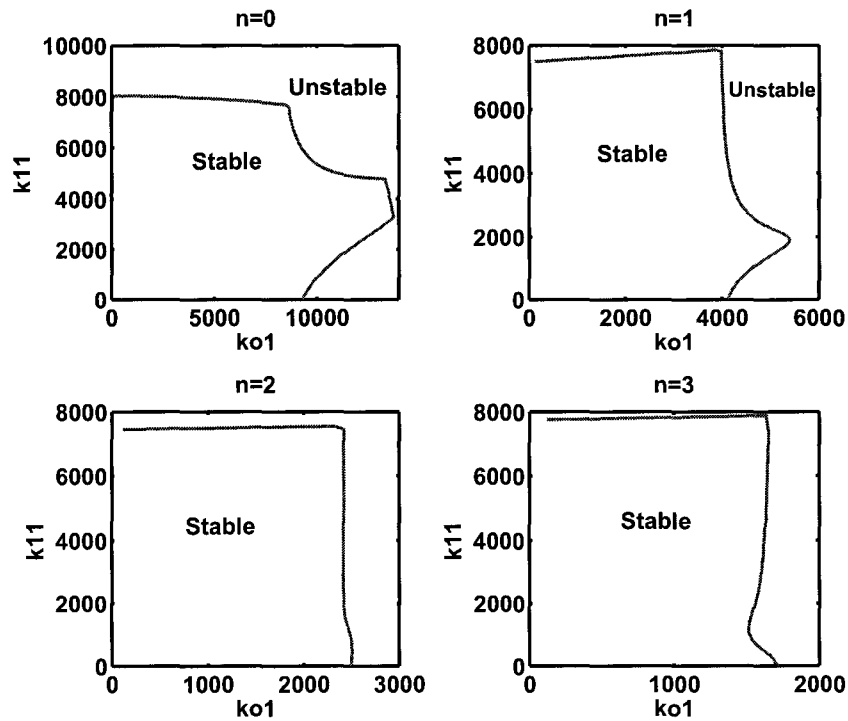


Figure 5.3: The region of stability for the distributed control architecture when $k_{12} = k_{21} = 300$ N/m are fixed for different communication channel delays.

the condition in Eq. (3.19). In Fig. 5.2, the stable region with respect to the parameters $k_{11} = k_{22}$ and $k_{12} = k_{21}$ in Fig. 4.4 is plotted while $k_{o1} = k_{o2} = 1000$ N/m are constant. Note that the stability region is noticeably enlarged when compared to that of the centralized architecture in Fig. 5.1. To study the effect of the coordinating virtual couplers between the objects, $k_{12} = k_{21} = 300$ N/m were fixed and $k_{11} = k_{22}$ and $k_{o1} = k_{o2}$ were varied with the results given in Fig. 5.3. Alternatively, Fig. 5.4 demonstrates the stable region for the case in which $k_{11} = k_{22} = 2000$ N/m are constant while $k_{o1} = k_{o2}$ and $k_{12} = k_{21}$ are changed. It should be noted that the

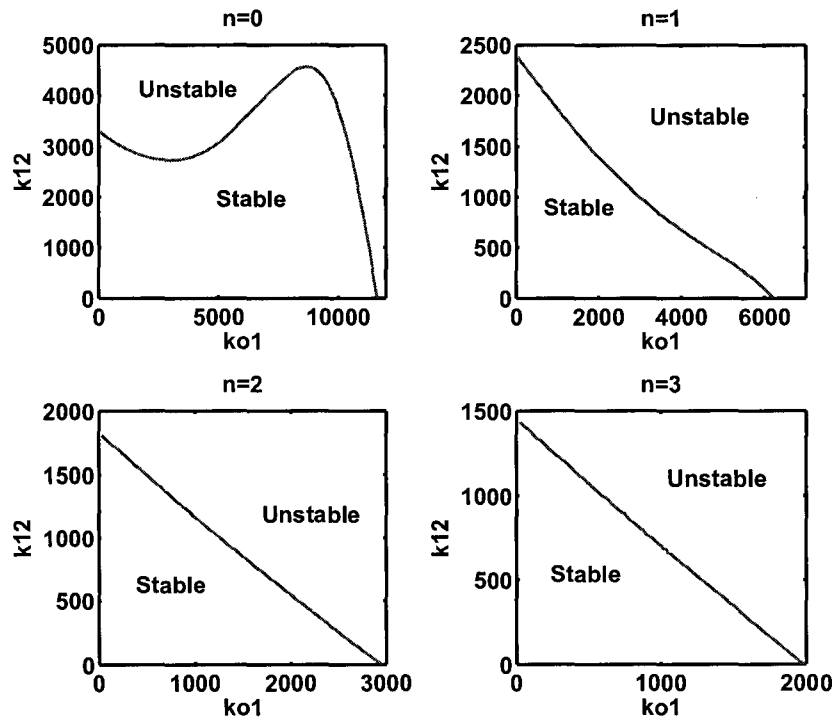


Figure 5.4: The region of stability for the distributed control architecture when $k_{11} = k_{22} = 2000$ N/m are fixed for different communication channel delays.

communication delay reduces the marginal values of k_o and k_{12} as expected.

The results of the previous analysis demonstrate that the distributed cooperative haptic architecture is capable of rendering rigid contacts under typical delays and network packet rates expected in a LAN or MAN, whereas the centralized controller can easily become unstable under such circumstances, unless the coupling stiffness of the remote user is considerably reduced.

Chapter 6

Performance Analysis

In this chapter the performance of the centralized and distributed architectures are studied. To this end, it is assumed that the users manipulate a virtual rigid object in free motion or in contact with a rigid wall. In Fig. 6.1, the mass-spring-damper models of the system under different scenarios are depicted. In hard contact, the value of k_w is relatively large, whereas in free motion k_w and b_w are set to zero. In an ideal case, the user should feel that he/she is interacting with a pure mass. To compare the controllers, the perceived admittances by the users in each architecture are compared with that of a mass in Fig. 6.1(d) for a network delay of $n = 1$ in Figs. 4.1 and 4.2.

6.1 Free Motion

In this case, it is assumed that one user is manipulating the virtual object in free motion while the second user is prohibited from interacting with the object. The perceived admittance of the object is defined as the ratio of the output velocity v_i^h

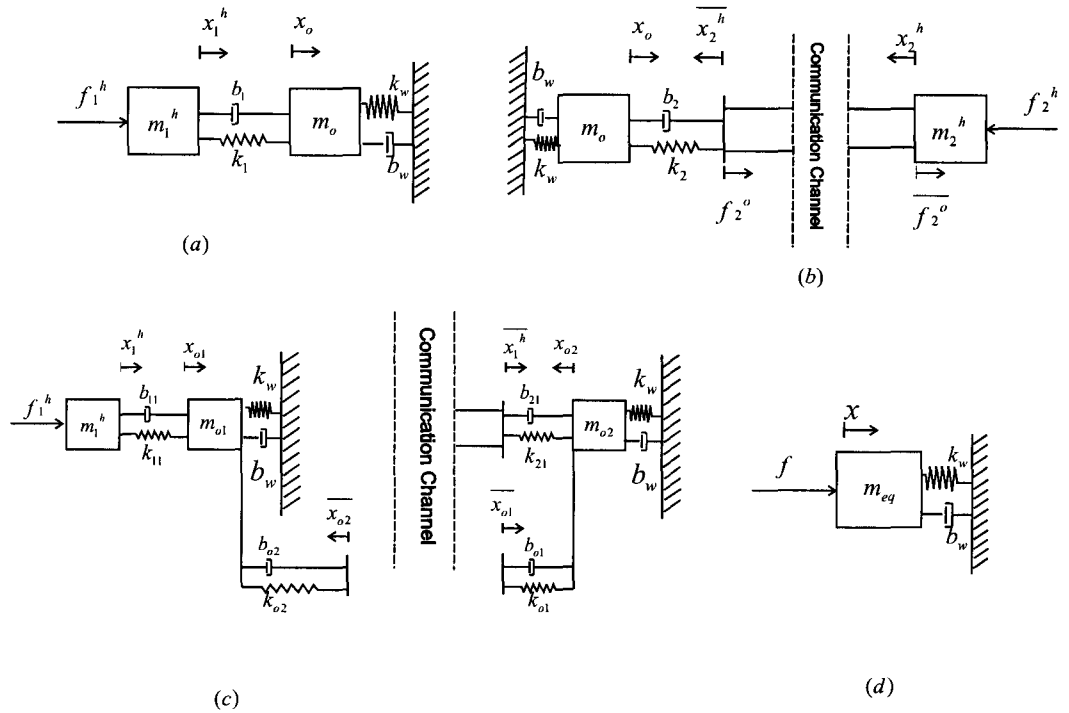
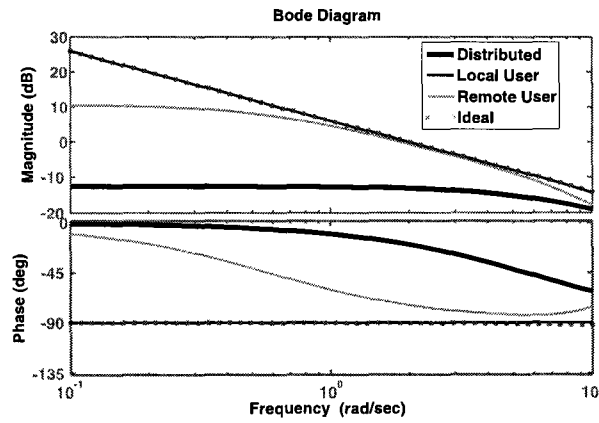


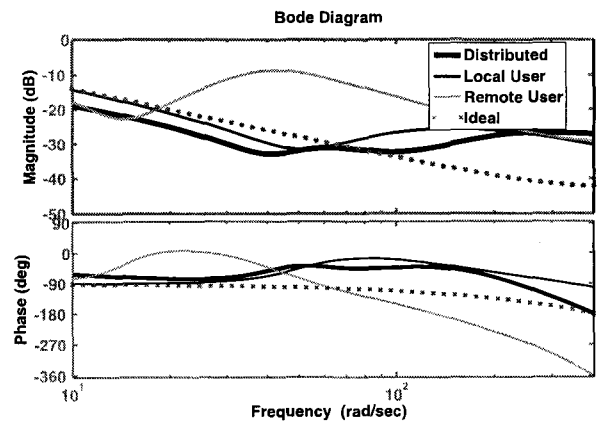
Figure 6.1: Model of haptic interaction with $k_w = 30000$ N/m and $b_w = 20$ N.s/m in hard contact and $b_w = k_w = 0$ in free motion: (a) Local user in centralized controller; (b) Remote user in centralized controller; (c) Distributed controller; (d) The ideal system.

to the input force f_i^h in Fig. 6.1. The modelling techniques introduced in Chapter 4 can be employed to formulate the discrete-time state-space representation of the systems shown in Fig. 6.1. The following equation shows the relationship between the transfer function and state-space representations of a discrete-time system.

$$G(z) = \frac{Y(z)}{U(z)} = C_D(zI - A_D)^{-1}B_D + D_D \quad (6.1)$$



(a) low frequency



(b) medium-to-high frequency

Figure 6.2: Object perceived admittance $\frac{V_i^h}{F_i^h}(j\omega)$ in free motion.

This can be proven by simply taking the z -transform of the state-space equations and solving the equations for the output signal. The relationship in Eq. (6.1) streamlines the calculation of the perceived admittance. Table 6.1 contains the parameters used in this case where the subscript o represents the moving object virtual coupler.

| | | | |
|--------------------------------------|---|---|----------------------------------|
| $k_1 = 1000 \text{ N/m}$ | $k_{o2} = k_{o1} = 300 \text{ N/m}$ | $b_{21} = 2 \text{ N}\cdot\text{s/m}$ | $m_o = 0.4 \text{ kg}$ |
| $k_2 = 85 \text{ N/m}$ | $b_1 = 15 \text{ N}\cdot\text{s/m}$ | $b_{o2} = b_{o1} = 10 \text{ N}\cdot\text{s/m}$ | $N = 8$ |
| $k_{11} = k_{22} = 3000 \text{ N/m}$ | $b_2 = 3 \text{ N}\cdot\text{s/m}$ | $T_c = 1/1024 \text{ s}$ | $m_1^h = m_2^h = 0.1 \text{ kg}$ |
| $k_{21} = 400 \text{ N/m}$ | $b_{22} = b_{11} = 20 \text{ N}\cdot\text{s/m}$ | | |

Table 6.1: The parameters values used for performance analysis of controllers in free motion.

The frequency responses are shown in Fig. 6.2, from which it is clear that the local user in the centralized architecture observes the closest admittance to that of the virtual mass, compared with the remote centralized and the distributed users. This should not be surprising since the network element is absent from the local user control loop and hence a high-rate delay free feedback loop can be implemented. It should be noted, however, that even in this case at higher frequencies the effect of spring-damper coupler becomes dominant and the response deviates from that of a mass as seen in Fig. 6.2(b). The network low packet rate and delay cause the largest deviation from the ideal response at medium to high frequencies in the perceived admittance of the remote user in the centralized controller.

At low frequencies, a dominant viscous behavior is observed in the responses of the distributed controller and the remote user in the centralized controller. The

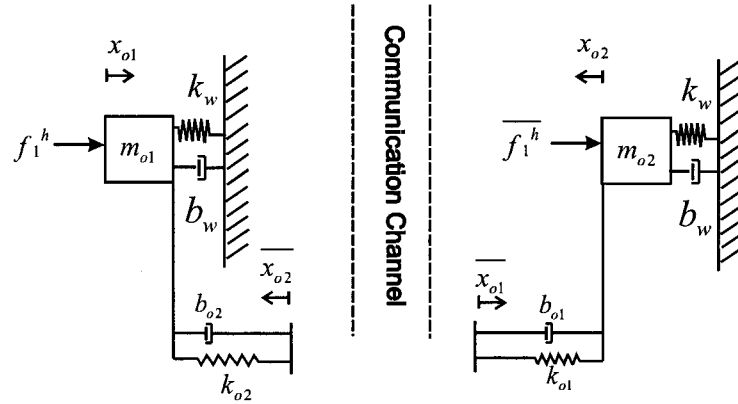


Figure 6.3: Simplified model of distributed architecture in which the user force is directly applied to the object.

amount of this damping can be analytically calculated as a function of system parameters using the discrete-time multi-rate modelling techniques introduced in Chapter 4 and the mass-spring-damper models in Fig. 6.1. This is achieved by finding the limit of the corresponding transfer functions as z approaches one.

$$b = \lim_{z \rightarrow 1} \frac{F_i^h}{V_i^h}(z) \quad (6.2)$$

However, a drawback of this approach is that for the rotational degrees of motion, the resulting damping values would depend on hand-object contact point which is unknown and can change. To overcome this problem, a simplified model is used in which the hand-object virtual couplers in Fig. 6.1 are removed and the hand forces are sampled with control and network rates and directly applied to the masses as shown in Fig. 6.3. Using this approach for the centralized controller

results in a value of zero for low-frequency damping and for the distributed controller with a control to network sampling ratio of $N = 2$, i.e. $T_t = 2T_c$, and a delay-free communication channel, e.g. $n = 0$, one may obtain

$$b_{dist} = 2 m_o k_o \left(4 \frac{m_o}{T_c} - T_c k_o - 2 b_o \right)^{-1} \quad (6.3)$$

The damping expressions obtained with the original models are rather long and are given in Appendix C. The analytical expressions of the delay in the case of $N > 2$ or when the network delay n is nonzero are complex and will not be given for brevity. In Table 6.2, the damping values obtained from the two approaches using the parameters in Table 6.1 are compared. Although the simplified method somewhat underestimates the damping, it can still provide a reasonable estimate for the rotational axes of the motion. Obviously, for the linear axes of the motion one can employ the original values.

It is evident from Eq. (6.3) that the amount of viscous damping in the distributed architecture increases as the control rate decreases. Tightening the coupling between the two copies of the object by increasing k_o would also increase the

| | Original Model | Simplified Model |
|-------------|----------------|------------------|
| Centralized | 0.04 N.s/m | 0 N.s/m |
| Distributed | 2.1 N.s/m | 1.4 N.s/m |

Table 6.2: The damping values obtained with the original and simplified models for $N = 8$.

induced viscous damping. To compensate for the undesirable effect of this viscous-type friction on user's perception of the object, it is proposed that a negative damping be added to the object dynamics. The damping computations can be performed off-line for every movable single-body object in the virtual environment using the modelling techniques presented in this thesis. If the simplified model is used, the underestimated damping value would only depend on the virtual object parameters and control coupling gains, which are precisely known. In the calculations based on the original model, the haptic device mass is also needed. This can be estimated with good accuracy in most cases. Since the communication/computation delay in the intended applications is usually around $n = 1 - 2$ network sample times, using $n = 0$ in the calculations would yield an underestimated damping value. Therefore, it is always possible to avoid a potential instability as a result of a net negative damping at the expense of an imperfect compensation. In addition, the system itself has some damping which guarantees a net positive damping value and hence a stable system.

Fig. 6.4 shows a significant improvement in the system response after introducing the compensator $b_o = -3 \text{ N.s/m}$, obtained using $n = 1$, in the distributed architecture. The perceived admittance is now close to that of the virtual mass for a wide frequency range.

6.2 Rigid Contact

As shown in Chapter 5, the distributed architecture has significantly higher marginal values for the stiffness couplers compared to those of the centralized controller. In

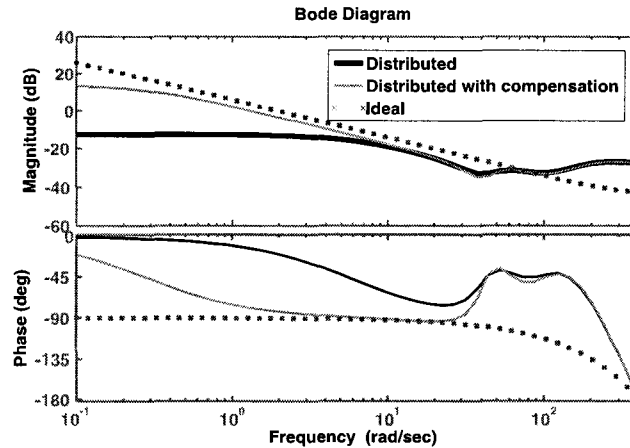


Figure 6.4: The user perceived admittance of the virtual object in the distributed controller before and after compensation compared with that of a pure mass.

contacts with rigid environments, the stiffness seen by the user can be approximated by the virtual coupler stiffness.

Fig. 6.5 displays the perceived admittance $\frac{X_i^h}{F_i^h}(j\omega)$ in the frequency domain when the user is pushing the object against a rigid wall. The controller parameters are the same as those in free motion and the wall stiffness and damping are $k_w = 30,000$ N/m, and $b_w = 20$ N.s/m, respectively. It turns out that the wall stiffness can be chosen to be substantially larger than the finger-box coupling stiffness. This can be explained by the fact that in the box-wall interaction, the wall is a static object with a fixed position in both copies of the virtual environment. However, in the finger-box interaction, the combined finger/haptic device mass introduces an extra dynamic mode in the system. As finger position is measured, transmitted over the network, and used in feedback control, this mode is subject to a low-rate network feedback loop which can limit the coupler stiffness. It is possible to calculate the equivalent stiffness seen by the user at the equilibrium using

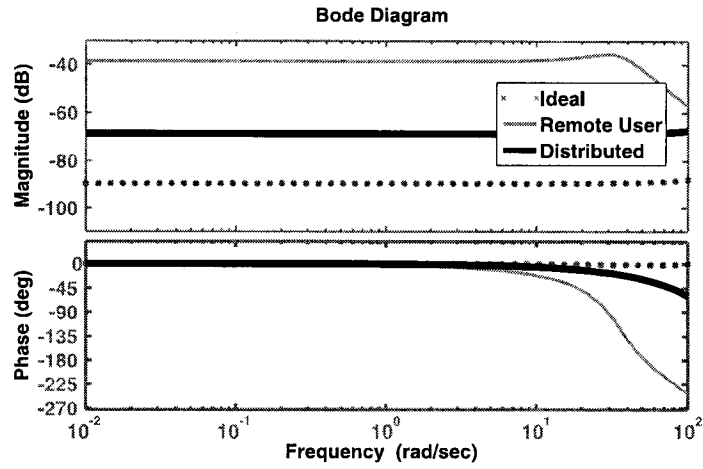


Figure 6.5: The perceived environment admittance $\frac{X_i^h}{F_i^h}(j\omega)$ when the box is in contact with a rigid wall.

the Hook's and Newton's laws which results in

$$\frac{1}{k_{eqidl}} = 1/k_w \quad (6.4)$$

$$\frac{1}{k_{eqrmt}} = \frac{k_2 + k_w}{k_w k_2}$$

$$\frac{1}{k_{eqdist}} = \frac{(k_{o2} k_{21} + k_{21} k_{11} + k_{11} k_{o2} + k_{11} k_w + 2 k_{o2}^2 + k_{21} k_w + 2 k_{o2} k_w + k_w^2)}{k_{11} (2 k_{o2}^2 + k_{21} k_w + 2 k_{o2} k_w + k_w^2)}$$

where k_{eq} is the ratio of user force to the user displacement, i.e. $\frac{F_i^h}{X_i^h}$ at the equilibrium.

Since k_w is noticeably larger than the other stiffness values, the expression for remote and local user can be approximated as follows.

$$\lim_{k_w \rightarrow \infty} \frac{1}{k_{eqrmt}} = \frac{1}{k_2} \quad (6.5)$$

$$\lim_{k_w \rightarrow \infty} \frac{1}{k_{eqdist}} = \frac{1}{k_{11}}$$

Therefore, at low frequencies the magnitude of responses in Fig. 6.5 is close to $1/k_2$, $1/k_{11}$ and $1/k_w$ for the remote, distributed and ideal system, respectively. This demonstrates a sharp contrast between the centralized and distributed controller in rendering rigid contacts where the achievable stiffness by the centralized architecture is quite limited.

In summary, it can be concluded that the distributed controller can achieve a superior performance over the centralized controller for manipulating objects in free motion and in rigid contact.

Chapter 7

A Platform for Two-finger Haptic Interaction in Cooperative Virtual Environments

An experimental platform for two-finger haptic rendering in dual-user virtual environment has been developed as shown in Fig. 7.1. The system consists of two haptic interfaces with grasping capability, centralized and decentralized control architectures for VE simulation, as well as graphical displays. Using this system, two users can cooperatively grasp and manipulate a virtual box in a plane, i.e. by moving it in x and y directions, rotating it around the z axis, and bringing it into contact with a rigid wall. The elements of the cooperative haptic platform are briefly described in the rest of this chapter.

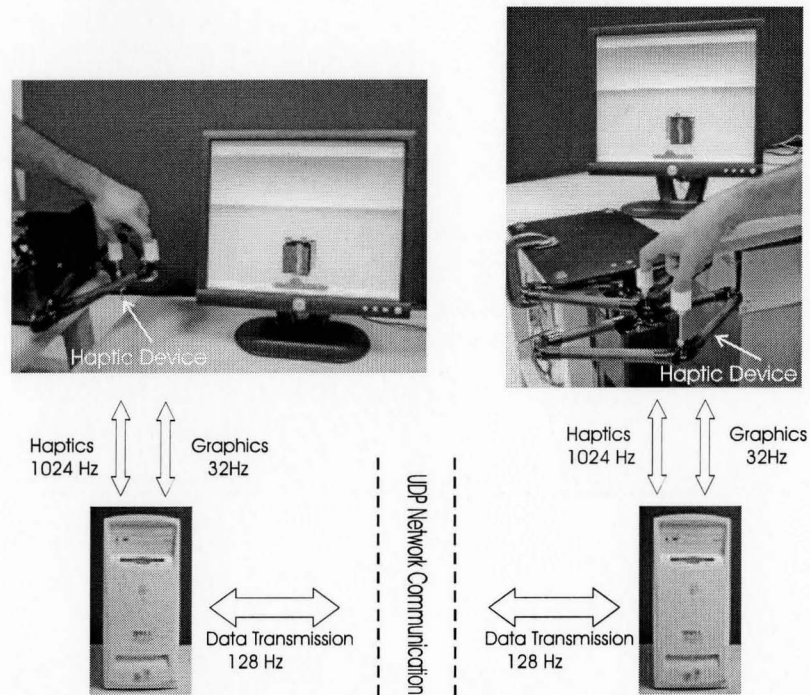


Figure 7.1: The experimental setup.

7.1 Haptic Device

Two identical planar pantograph mechanisms constitute a two-finger haptic interface. The device is a modified version of the planar Pantograph haptic interface from Quanser. These parallel manipulators are capable of measuring the endpoint position and producing force-feedback in the $x - y$ plane using rotary optical encoders and direct-drive DC motors. They are powered by two QPA linear current amplifiers from Quanser. A total of four pantograph mechanisms enable dual-user/dual-finger cooperative manipulation of virtual objects.

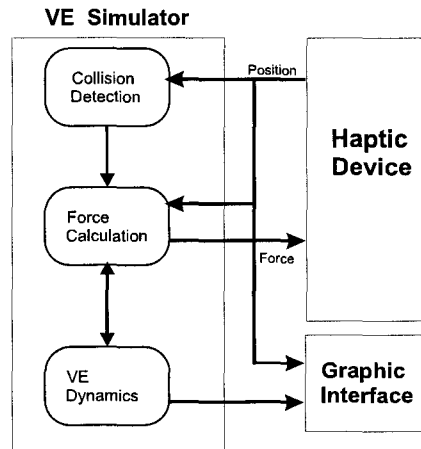


Figure 7.2: Basic elements for single-finger haptic rendering.

7.2 Virtual Environment

The VE engine is responsible for simulating the dynamics of the virtual object and generating model-based reaction forces that will be displayed to the users through the graphic consoles and haptic interfaces, as shown in Fig. 7.2. It should be pointed out that this figure represents the case in which a single user interacts with the VE by one finger. Nevertheless due to the modular structure of the experimental platform, the extension of the algorithms for the use in the dual-user/dual-finger system is trivial. The VE engine consists of three subsystems described below.

7.2.1 Collision Detection

The collision detection (CD) routine detects any potential collision between the user's finger and the surfaces of the virtual box, as well as between the virtual box and the rigid walls. The finger position in the workspace is calculated using

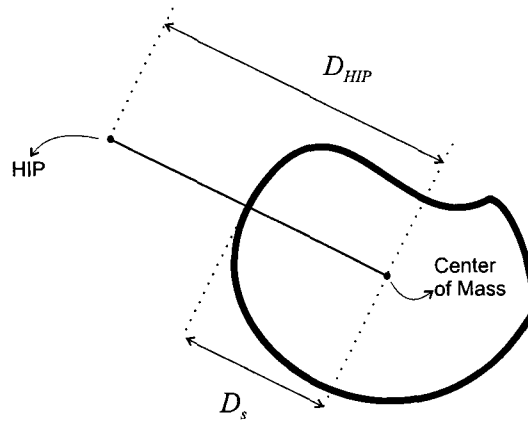


Figure 7.3: Interference test for collision detection.

encoder readings and the pantograph kinematics while the object position is generated by the VE engine. Due to the simple two-dimensional geometry of the object and environment configuration, the CD can be performed based on interference tests between the sides of the box and the user's finger as shown in Fig 7.3. In this approach the distance between HIP and object center of mass (D_{HIP}) is compared to the surface distance from the center of mass (D_s). If $D_s < D_{HIP}$ no collision is detected and hence the interaction force is zero, otherwise the collision detection routine triggers the force calculation algorithm. It assumed that the object has a convex shape. Similarly, collisions between the box and the walls can be detected by monitoring the positions of the box vertices in the x-y plane with respect to the walls.

7.2.2 Force Calculation

Fig. 7.4 displays a state machine used to generate the interaction force between each finger and the virtual box. According to this figure, there are four possible

contact states.

Free-motion

The system is in this state when there is no contact between the finger and object, and hence the interaction force is zero. Transitions from the free-motion to moment-cancellation state and from the contact-pre-slide or contact-slide to free motion state occur based on the positions of haptic interface point (HIP) and the virtual box and are triggered by the CD routine.

Moment-cancellation

The system enters into this state upon the first contact of the finger with the object surface. A sequence of impulse forces with appropriate magnitudes are generated and applied to the haptic interface in order to neutralize its velocity along the contact direction as quickly as possible. Such an approach provides the user with a more realistic sense of rigid contact [25,106]. The ideal value for the magnitude of impulses is infinity but in practice the impulse magnitude is set to maximum force that can be applied by haptic device. The conservation of momentum law implies

$$\int_0^T f(t)dt = m_h v_c \quad (7.1)$$

where m_h is the mass of user hand and haptic device lumped together and v_c is the magnitude of relative velocity between finger and object upon contact. It is assumed the the user hits a rigid wall or a relatively large mass. This equation is valid for any arbitrary signal f_t applied to device for T seconds. Replacing $f(t)$ in Eq. (7.1) by F_{Max} , the maximum haptic device force, one may calculate the number

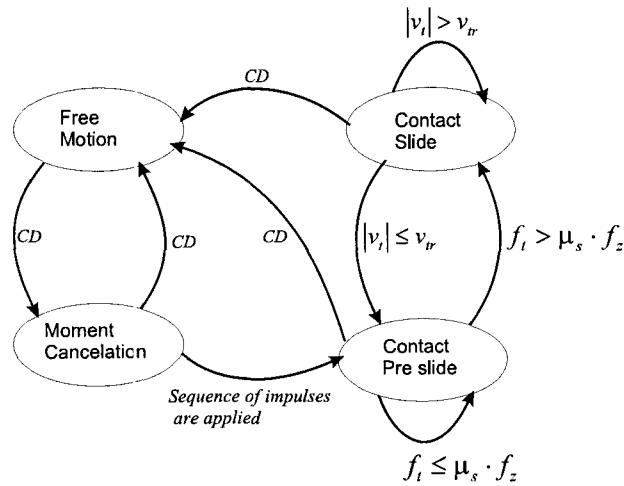


Figure 7.4: Contact states and their transitions.

of impulses needed to cancel the device/hand momentum as below.

$$F_{\text{Max}} T_c N = m_h v_c \Rightarrow N = \left[\frac{m_h v_c}{T_c F_{\text{Max}}} \right] \quad (7.2)$$

where T_c is the haptic simulation rate and the function $[x]$ returns the the largest intiger number smaller than x . In other words, the number of impulses is underestimated to avoid instability.

Contact pre-slide and contact-slide:

A penalty-based approach is adopted for calculating the interaction force between the HIP and the virtual box [107]. Linear spring-damper virtual couplers convert the penetration vector into a reaction force. The contact point frame (CPF) is defined as a frame whose z axis is normal to the object surface at the contact point. In the pre-slide state, the object/finger interaction forces expressed in the CPF are

calculated as follows.

$$\begin{aligned}
 f_x &= -k_x x_{\text{HIP}} - b_x \dot{x}_{\text{HIP}} \\
 f_y &= -k_y y_{\text{HIP}} - b_y \dot{y}_{\text{HIP}} \\
 f_z &= -k_z z_{\text{HIP}} - b_z \dot{z}_{\text{HIP}}
 \end{aligned} \tag{7.3}$$

For the contact-slide state one may write

$$\begin{aligned}
 f_z &= -k_z z_{\text{HIP}} - b_z \dot{z}_{\text{HIP}} \\
 f_x &= \frac{-\mu_k f_z \dot{x}_{\text{HIP}}}{\sqrt{\dot{x}_{\text{HIP}}^2 + \dot{y}_{\text{HIP}}^2}} \\
 f_y &= \frac{-\mu_k f_z \dot{y}_{\text{HIP}}}{\sqrt{\dot{x}_{\text{HIP}}^2 + \dot{y}_{\text{HIP}}^2}}
 \end{aligned} \tag{7.4}$$

where x_{HIP} , y_{HIP} , z_{HIP} , \dot{x}_{HIP} , \dot{y}_{HIP} , and \dot{z}_{HIP} are the positions and velocities of the HIP expressed in CPF, k and b are the virtual coupler damping and stiffness, and μ_k is the coefficient of dynamic friction. The system remains in the contact-pre-slide state as long as $f_t \leq \mu_s f_z$ with $f_t = \sqrt{f_x^2 + f_y^2}$ and μ_s being the coefficient of static friction, and switches to the contact-slide state otherwise.

The tangential velocity of HIP with respect to the CPF, v_t , triggers the transition from the contact-slide to contact-pre-slide when $|v_t| \leq v_{tr}$. Ideally, the threshold velocity $v_{tr} = 0$, however since the velocities are numerically computed from encoder readings, quantization-related errors can be noticeable at low velocities. Therefore, a small nonzero v_{tr} is empirically chosen to avoid erroneous state transitions. It should be pointed out that the proposed algorithm can be employed to generate the interaction force between any virtual contact point and a rigid surface with friction. For instance, the virtual wall can be rendered using the same

algorithm, with HIP and box surface being replaced with box corners and wall surfaces, respectively. Melder and Harwin in [108] have taken the same approach to model friction and surface forces using friction cone and god object concepts.

7.2.3 Virtual Environment Dynamics

In general, the virtual world can consist of static and dynamic objects. An Euler integration routine with fixed-step size of $1/1024s$ is invoked to calculate the motion of the dynamic objects, the virtual box in this case, based on the calculated user/object interaction forces and the object dynamics.

7.3 Graphics

Matlab's Virtual Reality toolbox has been employed for graphics rendering at the user console. The graphics update rate is set to 32 Hz which is sufficient for producing smooth motions on the displays.

7.4 Real-time Control Software

To meet the real-time requirements of haptic rendering, the real-time code is developed and executed in the Tornado/VxWorks and Matlab's Realtime Workshop environments. All local components of the haptic rendering, i.e. collision detection, force calculation and dynamic simulation run at a rate of 1024 Hz to generate a realistic force feedback.

7.5 Network Communication

Due to its relatively small overhead, the UDP protocol is used to communicate information between the two user workstations that are on a local network with a dedicated switch and small combined communication/computation latency. The network data transmission rate is set to 128 Hz to achieve a reliable communication between the haptic workstations. This packet transmission rate is dictated by the RTOS which calls the UDP communication routine on a regular basis. A handshaking protocol is implemented to ensure a synchronized data communication between the computers. Table 1.1 shows the experimental results carried out to define the communication channel characteristics. In this experiment two workstations were sending time-labelled data packet for a period of time 36 minutes with the sample rate of $1/128$ s.

Chapter 8

Experimental Results

Comparative experiments have been conducted with the dual-user/dual-finger haptic platform described in Chapter 7. A discussion of the experimental results follows.

8.1 Stability

The distributed and centralized architectures have been implemented on two machines running the VxWorks RTOS. The control parameters were set to values shown in Tables 8.1 and 8.2 in the case of centralized and distributed architectures, respectively. In these tables, N and T subscribes denote the normal and tangential directions along the contact between the finger and box, the indices x , y and R represent the x , y and rotation coordinates of the box, and the subscribe o denotes the moving object virtual coupler. The sampling times were set to $T_t = 1/128$ s, $T_c = 1/1024$ s. Coulomb friction with coefficients of 0.05 and 0.15 in the linear and rotational degrees of freedom was added between the box and ground. The

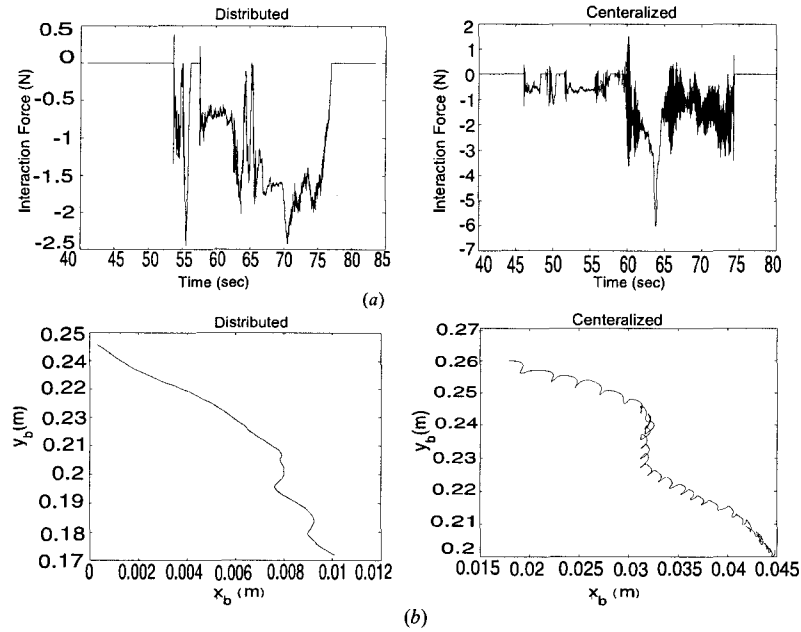


Figure 8.1: Comparison of distributed user and remote user in centralized architecture in experiment: (a) finger-box interaction force; (b) box's path on the x-y plane.

coefficient of Coulomb friction between the fingers and the box was set to 0.6.

The centralized controller with the given parameters is essentially unstable as predicted by the results of analysis in Fig. 5.1. In the experiments, while the users could cooperatively grasp and move the virtual box under the distributed controller, such operation was almost impossible with the centralized controller due to its instability. This is evident from the force and position profiles in Fig. 8.1 where the user has to exert significant damping in order to barely grasp the object and move it in the plane. The instability in the centralized framework can be avoided by reducing the stiffness of the virtual couplers for the user across the network. However, this would degrade the haptic sensation, specially in rigid contact.

| | |
|--------------------------------------|---|
| $k_{N1} = k_{N2} = 800 \text{ N/m}$ | $b_{N1} = b_{N2} = 3 \text{ N}\cdot\text{s/m}$ |
| $k_{T1} = k_{T2} = 1200 \text{ N/m}$ | $b_{T1} = b_{T2} = 10 \text{ N}\cdot\text{s/m}$ |

Table 8.1: Control parameters in centralized architecture used in experiments.

| | |
|--|---|
| $k_{N11} = k_{N22} = k_{N12} = k_{N21} = 800 \text{ N/m}$ | $b_{N11} = b_{N22} = b_{N12} = b_{N21} = 3 \text{ N}\cdot\text{s/m}$ |
| $k_{T11} = k_{T22} = k_{T12} = k_{T21} = 1200 \text{ N/m}$ | $b_{T11} = b_{T22} = b_{T12} = b_{T21} = 10 \text{ N}\cdot\text{s/m}$ |
| $k_{o1x} = k_{o1y} = k_{o2x} = k_{o2y} = 400 \text{ N/m}$ | $b_{o1x} = b_{o1y} = b_{o2x} = b_{o2y} = 8 \text{ N}\cdot\text{s/m}$ |
| $k_{o1R} = k_{o2R} = 10 \text{ Nm/rad}$ | $b_{o1R} = b_{o2R} = 0.1 \text{ N}\cdot\text{s/rad}$ |

Table 8.2: Control parameters in distributed architecture used in experiments.

8.2 Performance

Experiments have been conducted to investigate the effect of network downsampling on the users' perceived admittance. To apply consistent force in different experiments, the user forces are emulated through the control signals. One of the pantograph mechanisms is moved along a sinusoidal path with an amplitude of 0.05 m and a frequency of 2 rad/sec in the y direction using a proportional-derivative controller, while a constant force was applied to the second pantograph along the same direction. With this arrangement, the box was grasped and moved along the y direction by the two pantographs. Fig. 8.2(a) shows a snapshot from the graphical display of this experiment. The control parameters are the same as those in the previous case. Ideally, the sum of the forces applied on the virtual object must be equal to the inertial force required for moving the box along the sinusoidal path.

In Fig. 8.3, the local user force profile in the centralized architecture is compared with that of the user in the distributed architecture. When uncompensated,

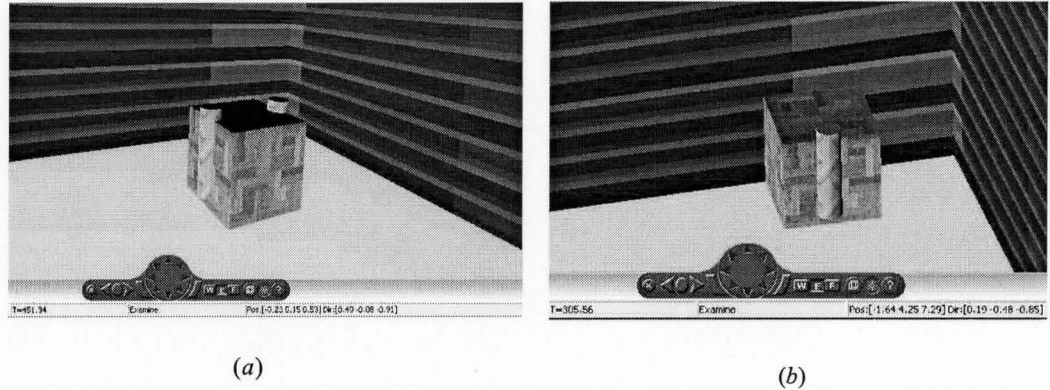


Figure 8.2: The virtual environment in performance evaluation experiments: (a) free motion; (b) rigid contact; grey cylinders represent the user fingers.

the user in the distributed system has to apply a noticeably larger force in order to generate the same motion. This is consistent with the analytical result that had predicted an extra viscous damping in the system response due to the network downsampling. As can be seen in Fig. 8.3, a negative damping compensation can significantly improve the response. The values of the damping were 0.88 N.s/m for the linear axes of motion, and 0.025 N.s.m/rad for the rotational motion, all chosen based on the results of the analysis. In the experiments, the users observed a noticeable improvement in the system response after the active damping compensation.

To compare the control architectures in rigid contact, the virtual box is pushed against a stiff wall with parameters $k_{Nw} = 4000$ N/m, $k_{Tw} = 7000$ N/m, $b_{Nw} = 30$ N.s/m, $b_{Tw} = 50$ N.s/m, using an emulated user force of $f_y = 1.5 \sin(3t)$ N, (see Fig. 8.2(b)). To avoid instability, the remote user parameters are set to $k_N = 170$ N/m, $b_N = 3$ N.s/m, $k_T = 270$ N/m, $b_T = 5$ N.s/m. The resulting haptic device

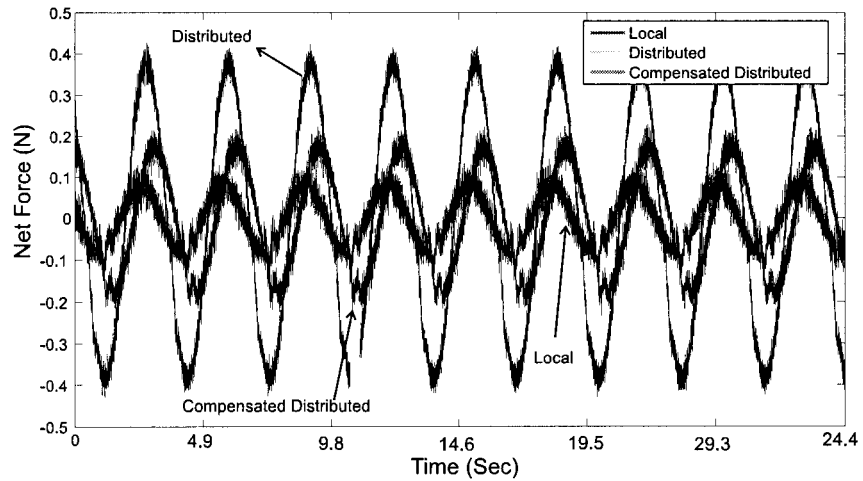


Figure 8.3: The net force exerted on the object for free motion along a sinusoidal path.

displacements are plotted in Fig. 8.4. The user in the distributed controller and the local user in the centralized controller perceive a much stiffer contact, as is evident from their smaller penetration in the virtual wall.

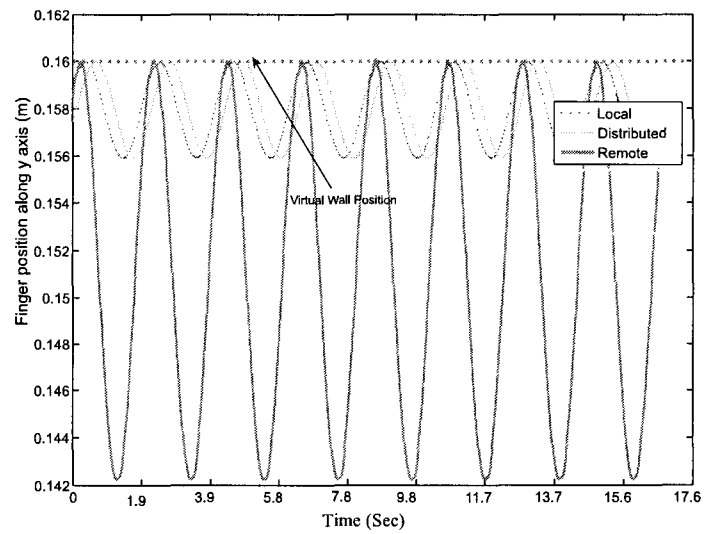


Figure 8.4: Finger position when the user pushes the object against a stiff wall.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Networked collaborative haptic environments present new challenges to the designers of haptic-enabled virtual reality systems. These are mainly due to non-deterministic network constraints such as limited packet transmission rate, latency, data loss and jitter. In this thesis the emphasis was on cooperative haptic simulation over LANs or MANs where the communication link can be characterized by its limited packet rate and a few ($n=0-2$) combined network/computation sample delays, and where jitter and packet loss may be ignored.

Using mathematical descriptions for multi-rate MIMO control systems, the stability and performance characteristics of two control architectures, namely centralized and distributed were compared. Analytical results as well as experiments conducted with a dual-user/dual-finger haptic platform demonstrated that the distributed controller possesses noticeably larger stability margins. Also, it was shown that the network downsampling and delay can cause a viscous friction

effect in the perceived impedance of the virtual object. This was eliminated by adding a negative damping to the virtual object dynamics. The negative damping value was formulated using the proposed modeling techniques. It was demonstrated that, when properly compensated, the distributed controller can provide higher haptic fidelity in free motion and in contact with rigid environments compared with that achievable using the centralized controller.

An experimental setup for multi-finger cooperative haptic was developed using VxWorks RTOS and the Pantograph mechanism. In such a setup users are able to manipulate a rigid box in a virtual world and interact with rigid walls. Experiments carried out to validate the stability and performance of centralized and distributed controllers in free motion and rigid contact.

9.1.1 Suggestions for Future Research

The following directions may be pursued for future research,

- The impact of longer time delays, delay jitter and packet loss on cooperative haptic simulation over WANs and in particular the Internet can be investigated and methods for improving stability and performance under such conditions may be sought.
- Predictive models for human hand gesture and behavior may be considered for reducing network-related performance degradation in cooperative haptics. A possible approach to the modelling of the user interaction with the virtual world is to consider a number of internal decision states for the user and proper state transition dynamics, e.g. based on a Markov chain.

- Cooperative haptics in virtual environments with deformable objects is another interesting area for future research. Techniques for performance evaluation and stability analysis of such systems can be studied. In the case of deformable haptic interaction, the haptic simulation rate can be limited due to extensive computational cost of finite element method. Also the effect of downsampling in the context of cooperative haptics for deformable bodies must be considered.

Appendix A

State Transition Matrices for Discrete-time Multi-rate Dynamics with ZOH

Reference [105] contains a detailed derivation of the state transition matrices in (4.18). The final forms of the matrices are given in here for ease of reference. A_D is composed of $N_{lcm} \times N_{lcm}$ matrix blocks of the form:

$$A_D = \begin{pmatrix} 0 & \dots & \dots & 0 & A_{D1} \\ \vdots & & & & \vdots \\ 0 & \dots & \dots & 0 & A_{DN_0} \end{pmatrix} \quad (\text{A.1})$$

where A_{Dl} is given by

$$A_{Dl} = e^{Al\tau_0}, \quad l = 1, \dots, N_{lcm} \quad (\text{A.2})$$

B_D is composed of $1 \times p$ matrix blocks of the form:

$$B_D = \begin{pmatrix} B_{D1} & B_{D2} & \dots & B_{Dp} \end{pmatrix} \quad (\text{A.3})$$

where p is the number of sample rates in the system and B_D blocks are $N_{lcm} \times N_i$ block matrices

$$B_{Di} = [b_{Di,lu}], \quad l = l, \dots, N_{lcm}; \quad \mu = 0, \dots, N_i - 1. \quad (\text{A.4})$$

The block $b_{Di,lu}$ of B_{iD} is an n_o -dimensional column vector

$$b_{Di,lu} = \begin{cases} 0 & l \leq \mu l_i \\ \int_{\mu l_i \tau_0}^{l \tau_0} e^{A(l\tau_0 - \tau)} b_i d\tau & \mu l_i < l \leq (\mu + 1)l_i \\ \int_{\mu l_i \tau_0}^{(\mu + 1)l_i \tau_0} e^{A(l\tau_0 - \tau)} b_i d\tau & (\mu + 1)l_i < l \end{cases} \quad (\text{A.5})$$

C_D is a $p \times 1$ block matrix of the form:

$$\begin{pmatrix} C_{D1} \\ \vdots \\ C_{Dp} \end{pmatrix}. \quad (\text{A.6})$$

whose blocks are $N_j \times N_o$ block matrices

$$B_{Di} = [b_{Di,lu}], \quad l = l, \dots, N_{lcm}; \quad \mu = 0, \dots, N_i - 1. \quad (\text{A.7})$$

$$C_{Dj} = [(c_{Dj,vl})^T], \quad v = 0, \dots, N_j - 1; \quad l = 1, \dots, N_0 \quad (\text{A.8})$$

The block $(c_{Dj,vl})^T$ is an n_o dimensional row vector given by:

$$(c_{Dj,vl})^T = \begin{cases} c_j^T & v = 0, l = N_0 \text{ or } vl_i = l \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.9})$$

Appendix B

Viscous Damping Calculation Using the Original Model

The damping values in the case of $N = 2$ and $n = 0$ for the single-axis model of Fig. 6.1 have been derived using Matlab's Symbolic toolbox. For the centralized controller:

$$b_{cent} = \frac{k_2}{2/T_c - b_2/m_o} \quad (\text{B.10})$$

and for the distributed controller:

$$b_{dist} = \frac{NUM}{DEN} \quad (\text{B.11})$$

$$\begin{aligned}
NUM = & T_c k_{11} (10m_1 k_o^2 T_c^2 b_o b_{11} - 24T_c k_o^2 m_1 m_o b_{11} - 40m_1 T_c m_o k_o^2 b_o \\
& - 24m_1 T_c m_o k_o^2 b_{12} + T_c^3 b_{12}^2 k_o^2 b_{11} - 24T_c b_{11} m_o^2 k_o^2 - 4T_c^3 b_{11} b_o^2 k_o^2 \\
& + 10m_1 T_c^2 b_o k_o^2 b_{12} + 8m_1 T_c^2 b_o^2 k_o^2 - 4T_c^3 b_{11} b_o k_o^2 b_{12} + 10T_c^2 b_{11} m_o k_o^2 b_{12} \\
& + 12m_1 T_c^2 k_o^2 b_{12} b_{11} + 48m_1 m_o^2 k_o^2 + 20T_c^2 b_{11} m_o k_o^2 b_o - 2T_c^2 k_{12}^2 m_o b_o b_{11} \\
& - 2T_c^2 k_{12}^2 m_1 b_o^2 - 2T_c^2 k_{12}^2 m_1 b_o b_{11} + 4T_c k_{12}^2 m_1 m_o b_o + T_c^3 k_{12}^2 b_o^2 b_{11} \\
& + 8T_c^2 m_1 b_{11} k_o b_o k_{12} - 40T_c m_1 m_o k_o b_o k_{12} - 4T_c^3 k_o b_o^2 b_{11} k_{12} \\
& + 8T_c^2 m_1 b_o^2 k_o k_{12} + 12m_1 T_c^2 b_{11} k_o b_{12} k_{12} + 12m_1 T_c^2 b_o k_o b_{12} k_{12} \\
& + 20T_c^2 M_o k_o b_o k_{12} b_{11} + 12T_c^2 m_o k_{12} k_o b_{12} b_{11} - 6T_c^3 k_{12} k_o b_{12} b_{11} b_o \\
& - 24m_1 m_o T_c k_o b_{12} k_{12} + 48m_1 m_o^2 k_{12} k_o - 24T_c m_o^2 k_{12} k_o b_{11} - 24m_1 m_o T_c b_{11} k_o k_{12})
\end{aligned} \tag{B.12}$$

$$\begin{aligned}
DEN = & (T_c b_o + T_c b_{12} - 2m_o)(k_{11} k_{12} + k_o k_{12} + k_{11} k_o) \\
& (-T_c^3 b_{11} k_o - 2T_c^2 b_{11} b_o + 4m_1 T_c b_o + 4m_1 T_c b_{11} + 4m_o T_c b_{11} - 8m_1 m_o)
\end{aligned} \tag{B.13}$$

Bibliography

- [1] R. Waters and J. W. Barrus, "The rise of shared virtual environments," in *Int. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 20–25, 1997.
- [2] Y. Yokokohji, R. Hollis, T. Kanade, K. Henmi, and T. Yoshikawa, "Toward machine mediated training of motor skills. Skill transfer from human to human via virtual environment," in *IEEE Int. Workshop on Robot and Human Communication*, pp. 32–37, 1996.
- [3] V. Hayward, P. Gregorio, O. Astley, S. Greenish, and M. Doyon, "Freedom-7: A high fidelity seven axis haptic device with application to surgical training," in *Int. Symposium on Experimental Robotics*, pp. 445–456, 1997.
- [4] S. Gibson, J. Samosky, A. M. C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima, and H. Ohkami, "Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback," in *Joint Conf. on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer Assisted Surgery*, pp. 369–378, 1997.

- [5] Y. Jung, S.-C. Yeh, and J. Stewart, "Tailoring virtual reality technology for stroke rehabilitation: a human factors design," in *Conference on Human Factors in Computing Systems archive*, pp. 929–934, 2006.
- [6] J. Mullins, C. Mawson, and S. Nahavandi, "Haptic handwriting aid for training and rehabilitation," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 2690–2694(3), 2005.
- [7] C. Basdogan, C. H. Ho, M. A. Srinivasan, and M. Slater, "An experimental study on the role of touch in shared virtual environments," *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 443–460, 2000.
- [8] R. Traylor, D. Wilhelm, B. Adelstein, and H. Z. Tan, "Design considerations for stand-alone haptic interfaces communicating via UDP protocol," in *Proc. World Haptics Conf.*, pp. 563–564, 2005.
- [9] R. Goertz, "Fundamentals of general purpose remote manipulators," *Nuclearonics*, vol. 10, pp. 36–42, 1952.
- [10] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulator, 2nd Edition*. Springer, 2000.
- [11] V. Hayward and et al., "Haptic devices and interfaces," *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004.
- [12] T. Massie and J. Salisbury, "The phantom haptic interface: A device for probing virtual objects.," in *SME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 295–301, 1994.

- [13] M. Sirouspour, S. Dimaio, S. Salcudean, P. Abolmaesumi, and C. Jones, "Haptic interface control-design issues and experiments with a planar device," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 789–794, 2000.
- [14] V. Hayward, J. Choksi, G. Lanvin, and C. Ramstein, "Design and multi-objective optimization of a linkage for a haptic interface," in *Int. Workshop Advances Robot Kinematics*, p. 352359, 1994.
- [15] B. D. Adelstein and M. J. Rosen, "Design and implementation of a force reflective manipulandum for manual control research," in *Proc. ASME Advances in Robotics*, pp. 1–12, 1992.
- [16] K. MacLean and J. Roderick, "Smart tangible displays in the everyday world: A haptic door knob," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 203–208, 1999.
- [17] M. Turner, D. Gomez, M. Tremblay, and M. Cutkovsky, "Preliminary tests of an arm-grounded haptic feedback device in telemanipulation," in *the Winter Annual Meeting of ASME*, pp. 145–149, 1998.
- [18] M. Bouzit, *Design Implementation And Testing Of A Data Glove With Force Feedback For Virtual And Real Objects Telemanipulation*. PhD. Thesis, Paris, France, 1996.
- [19] D. Gomez, G. Burdea, and N. Langrana, "Integration of the rutgers master ii in a virtual reality simulation," in *IEEE Virtual Reality Annual Int. Symposium*, pp. 198–202, 1995.

- [20] V. Popescu, G. Burdea, and M. Bouzit, "Virtual reality simulation modelling for a haptic glove," in *Proc. of Computer Animation*, pp. 195–200, 1999.
- [21] H. Iwata and H. Noma, "Volume haptization," in *Symposium on Research Frontiers in Virtual Reality*, pp. 16–23, 1993.
- [22] R. S. Avila and L. M. Sobierajski, "A haptic interaction method for volume visualization," in *Visualization Proc.*, pp. 197–204, 1996.
- [23] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, pp. 146–151(3), 1997.
- [24] D. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proc. ACM Press*, pp. 345–352, 1997.
- [25] D. Constantinescu, S. Salcudean, and E. Croft, "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 309–323, 2005.
- [26] C. Basdogan, C. Ho, and M. Srinivasan, "A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments," in *ASME Winter Annual Meeting*, pp. 77–84, 1997.
- [27] J. Chen, C. DiMattia, R. Taylor, M. Falvo, P. Thiansathon, and R. Superfine, "Sticking to the point: A friction and adhesion model for simulated surfaces," in *Proc. ASME Dynamics Systems and Control Division*, pp. 167–171, 1997.

- [28] D. Haessig and B. Friedland, "On the modelling and simulation of friction," *ASME Journal of dynamic Systems, Measurement and Control*, vol. 113, pp. 354–362, 1991.
- [29] V. Hayward and B. Armstrong, "A new computational model of friction applied to haptic rendering," *Experimental Robotics VI, Lecture Notes in Control and Information Sciences*, pp. 403–412, 2000.
- [30] P. R. Dahl, "Solid friction damping of mechanical vibrations," *AIAA Journal*, vol. 14, pp. 1675–1682(12), 1976.
- [31] C. Richard and M. Cutkosky, "Friction modelling and display in haptic applications involving user performance," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 605–611(1), 2002.
- [32] D. Karnopp, "Computer simulation of stick-slip friction in mechanical dynamic systems," *ASME Journal of dynamic Systems, Measurement and Control*, vol. 107, pp. 100–103(1), 1985.
- [33] J. Fritz and K. Barner, "Stochastic models for haptic texture," in *Proc. Conf. Telemanipulator and telepresence technologies*, pp. 34–44, 1996.
- [34] C. Ho, C. Basdogan, and M. Srinivasan, "An efficient haptic rendering technique for displaying 3D polyhedral objects and their surface details in virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 8, pp. 477–491, 1999.
- [35] J. Siira and D. Pai, "Haptic texturing - A stochastic approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 557–562, 1996.

- [36] M. A. Costa and M. R. Cutkosky, "Roughness perception of haptically displayed fractal surfaces," in *Proc. the ASME Dynamic Systems and Control Division*, pp. 1073–1079, 2000.
- [37] J. Colgate and G. Schenkel, "Passivity of a class of sampled-data systems: Application to haptic interfaces," *J. Robot. Syst.*, vol. 14, no. 1, pp. 37–47, 1997.
- [38] B. Miller, J. Colgate, and R. Freeman, "Guaranteed stability of haptic systems with nonlinear VE," *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 20, pp. 712–719, 2000.
- [39] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delay," *The Int. Journal of Robotics Research*, vol. 11, pp. 135–149(2), 1992.
- [40] G. Niemeyer and J. J. Slotine, "Stable adaptive teleoperation," *IEEE journal of oceanic engineering*, vol. 16, pp. 152–162, 1991.
- [41] M. Mahvash and V. Hayward, "Passivity-based high-fidelity haptic rendering of contact," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3722–3728, 2003.
- [42] R. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 15, pp. 465–474, 1999.
- [43] J.-H. Ryu, B. Hannaford, C. Preusche, and G. Hirzinger, "Time domain passivity control with reference energy behavior," in *IEEE Conf. Intelligent Robots and Systems*, pp. 2932–2937(3), 2003.

- [44] J.-H. Ryu, Y. Kim, and B. Hannaford, "Sampled- and continuous-time passivity and stability of virtual environments," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 772–776, 2004.
- [45] B. Hannaford and J. Ryu, "Time-Domain passivity control of haptic interfaces," *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 18, pp. 1–10, 2002.
- [46] D. Wang, Z. Y. amd W. Yuhui, Y. Lee, L. Peijun, and W. Yong, "Cutting on triangle mesh: local model-based haptic display for dental preparation surgery simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, pp. 671–683(6), 2005.
- [47] F. Barbagli, D. Prattichizzo, and K. Salisbury, "A multirate approach to haptic interaction with deformable objects single and multipoint contacts," *The Int. Journal of Robotics Research*, vol. 24, pp. 703–715(9), 2005.
- [48] C. Cho, M. Kim, C.-S. Hwang, J. Lee, and J.-B. Song, "Stable haptic display of slowly updated virtual environment with multirate wave transform," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 2465– 2470, 2005.
- [49] A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney, " Animation of fracture by physical modelling," *Visual Computation*, vol. 7, pp. 210–219(4), 1991.
- [50] S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaki, "Modelling and implementation of elastic object manipulation in virtual space," *Electron Commun Jpn Part III Fundam Electron Sci*, vol. 81, pp. 42–50(4), 1998.

- [51] K. Waters, "A physical model of facial tissue and muscle articulation derived from computer tomography data," in *Visualization in Biomedical Computing Proc.*, pp. 574–584, 1992.
- [52] S. C. H. Delingette, G. Subsol and J. Pignon, "A craniofacial surgery simulation testbed," in *Visualization in Biomedical Computing Proc.*, pp. 607–618, 1994.
- [53] U. G. Kuhnafel and B. Neisius, "Realtime graphical computer simulation for endoscopic surgery," in *Proc. Medicine Meets Virtual Reality II*, pp. 27–30, 1994.
- [54] C. Kuhn, U. Kuhnafel, H. G. Krumm, and B. Neisius, "Virtual reality based training system for minimally invasive surgery," in *Proc. Computer Assisted Radiology*, pp. 764–769, 1996.
- [55] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von, G. Fankhauser, and Y. Parish, "Simulating facial surgery using finite element models," *Computer Graphics*, vol. 30, no. Annual Conf. Series, pp. 421–428, 1996.
- [56] H. Delingette, "Towards realistic soft tissue modelling in medical simulation," Tech. Rep. RR-3506.
- [57] M. Bro-Nielsen and S. Cotin, "Soft tissue modelling in surgery simulation for prediction of results of craniofacial operations and steps toward virtual reality training systems," in *Proc. 3rd Int. Workshop Rapid Prototyping in Medicine and Computer-Assisted Surgery*, pp. 19–21, 1995.

- [58] M. Bro-Nielsen, "Mvox: Interactive 2-4D medical image and graphics visualization software," in *Proc. 3rd Int. Workshop Rapid Prototyping in Medicine and Computer-Assisted Surgery*, pp. 335–338, 1996.
- [59] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," in *Proc. Computer Graphics Forum, Eurographics*, pp. 57–66, 1996.
- [60] S. Cotin, H. Delingette, M. Bro-Nielsen, N. Ayache, J. M. Clement, V. Tassetti, and J. Marescaux, "Geometric and physical representations for a simulator of hepatic surgery," in *Proc. Medicine Meets Virtual Reality*, pp. 139–151, 1996.
- [61] S. Cotin, H. Delingette, J. Clement, V. Tassetti, J. Marescaux, and N. Ayache, "Volumetric deformable models for simulation of laparoscopic surgery," in *Proc. Computer Assisted Radiology*, pp. 793–798, 1996.
- [62] O. Zienkiewicz, *The Finite Element Method, 3rd Edition*. McGraw-Hill, London, 1997.
- [63] Y. Zhuang and J. Canny, "Haptic interaction with global deformations," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 2428–2433(3), 2000.
- [64] G. Picinbono, H. Delingette, and N. Ayache, "Nonlinear and anisotropic elastic soft tissue models for medical simulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1370–1375(2), 2001.
- [65] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, "Nonlinear finite elements for deformable body simulation using dynamic progressive meshes," *Computer Graphics Forum*, vol. 20, no. 3, pp. 349–358, 2001.

- [66] Y. Kurodaa, M. Nakaob, T. Kurodac, H. Oyamad, and M. Komorie, "Interaction model between elastic objects for haptic feedback considering collisions of soft tissue," *Computer Methods and Programs in Biomedicine*, vol. 80, pp. 216–224(3), 2005.
- [67] G. J. Song and N. P. Reddy, "Tissue cutting in virtual environment," in *Proc. Medecine Meets Virtual Reality IV*, pp. 359–364, 1995.
- [68] M. Bro-Nielsen, "Finite element modelling in surgery simulation," *Proc. IEEE Virtual and Augmented Reality in Medicine*, vol. 86, pp. 490–503(3), 1998.
- [69] A. Tanaka, K. Hirota, and T. Kaneko, "Virtual cutting with force feedback," in *Proc. Virtual Reality Annual Int. Symposium*, pp. 71–75, 1998.
- [70] S. Cotin, H. Delingette, and N. Ayache, "A hybrid elastic model for real-time cutting, deformations and force feedback for surgery training and simulation," *The Visual Computer*, vol. 16, no. 7, pp. 437–452, 2000.
- [71] L. Kim and S. H. Park, "Haptic interaction and volume modelling techniques for realistic dental simulation," *Int. Journal of Computer Graphics*, vol. 22, pp. 90–98, 2006.
- [72] G. Thomas, L. Johnson, S. Dow, and C. Stanford, "The design and testing of a force feedback dental simulator," *Computer Methods and Programs in Biomedicine*, vol. 64, pp. 53–64(1), 2001.
- [73] J. Ranta and W. Avile, "The virtual reality dental tranining system simulating dental procedures for the purpose of training dental students using haptics," in *Fourth Phantom User Groups Workshop Proc.*, pp. 73–77, 1999.

- [74] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick, "Project GROPE-Haptic displays for scientific visualization," in *Proc. SIGGRAPH*, pp. 177–186, 1990.
- [75] R. Taylor, W. Robinett, V. Chi, F. Brooks, W. Wright, R. Williams, and E. Snyder, "The nanomanipulator: A virtual-reality interface for a scanning tunneling microscope," in *Proc. SIGGRAPH*, pp. 127–134, 1993.
- [76] M. R. Falvo, S. Superfine, M. Washburn, R. Finch, V. Taylor, and F. Brooks, "The nanomanipulator: a teleoperator for manipulating materials at the nanometer scale.," in *Proc. Int. Symposium on the Science and Technology of Atomically Engineered Materials*, pp. 579–586, 1996.
- [77] A. Zorcolo, M. Tuveri, and G. Zanetti, "Catheter insertion simulation with coregistered direct volume rendering and haptic feedback," in *Medicine Meets Virtual Reality*, pp. 96–98, 2000.
- [78] P. Gorman, T. Krummel, R. Webster, M. Smith, and D. Hutchens, "A prototype haptic lumbar puncture simulator," in *Medicine Meets Virtual Reality*, pp. 106–109, 2000.
- [79] L. Hiemenz, J. McDonald, D. Stredney, and D. Sessanna, "A physiologically valid simulator for training residents to perform an epidural block," in *Proc. IEEE Biomedical Engineering Conf.*, pp. 170–173, 1996.
- [80] P. N. Brett, T. J. Parker, A. J. Harrison, T. A. Thomas, and A. Carr, "Simulation of resistance forces acting on surgical needles," *Proc. the Institution of Mechanical Engineers Part H: Journal of Engineering in Medicine*, vol. 211, pp. 335–347,

1997.

- [81] V. Vuskovic, M. Kauer, G. Szekely, and M. Reidy, "Realistic force feedback for virtual reality based diagnostic surgery simulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1592–1598(2), 2000.
- [82] U. Kuhnappel, H. Cakmak, and H. Maasz, "Endoscopic surgery training using virtual reality and deformable tissue simulation," *Computers and Graphics*, vol. 24, pp. 671–682(12), 2000.
- [83] S. Payandeh, "Force propagation models in laparoscopic tools and trainers," in *Proc. IEEE Engineering in Medicine and Biology Int. Conf.*, pp. 957–960, 1997.
- [84] C. Basdogan, C.-H. Ho, and M. Srinivasan, "Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration," *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp. 269–285, 2001.
- [85] A. Kimura, J. Camp, R. Robb, and B. Davis, "A Prostate brachytherapy training rehearsal system simulation of deformable needle insertion," *Proc. the 5th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention-Part I*, vol. 6, pp. 264–271, 2002.
- [86] X. Wang and A. Fenster, "A haptic-enhanced 3D real-time interactive needle insertion simulation for prostate brachytherapy," in *Proc. SPIE*, pp. 781–789, 2004.

- [87] P. Buttolo, D. Kung, and B. Hannaford, "Manipulation in real, virtual and remote environments," in *IEEE int. Con. on Systems, Man and Cybernetics*, pp. 4656–4661, 1995.
- [88] H. Takemura and F. Kishino, "Cooperative work environment using virtual workspace," in *Proc. ACM Conf. on Computer-supported cooperative work*, pp. 226–232, 1992.
- [89] M. O. Alhalabi and S. Horiguchi, "Haptic cooperative virtual workspace: Architecture and evaluation," *Virtual Reality*, vol. 5, pp. 160–168, 2005.
- [90] T. Hudson, A. Helser, D. Sonnenwald, and M. Whitton, "Managing collaboration in the nanoManipulator," in *Proc. IEEE Virtual Reality*, pp. 180–187, 2003.
- [91] P. Buttolo, R. Oboe, and B. Hannaford, "Architectures for shared haptic virtual environments," *Computers and Graphics*, vol. 21, pp. 421–429, 1997.
- [92] T. Yoshikawa and H. Ueda, "Construction of virtual world using dynamics modules and interaction modules," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 2358–2364, 1996.
- [93] K. Hikichi, H. Morino, I. A. K. Sezaki, and Y. Yasuda, "The evaluation of delay jitter for haptics collaboration over the Internet," in *IEEE Global Telecommunications Conf.*, pp. 17–21, 2002.
- [94] D. Wang, K. Tuer, M. Rossi, L. Ni, and J. Shu, "The effect of time delays on tele haptics," in *Proc. Haptic, Audio and Visual Environments and Their Applications*, pp. 7–12, 2003.

- [95] M. Alhalabi, S. Horiguchi, and S. Kunifuji, "An experimental study on the effects of network delay in cooperative shared haptic virtual environment," *Computers and Graphics*, vol. 27, pp. 205–213(9), 2003.
- [96] K. S. Park and R. Kenyon, "Effects of network characteristics on human performance in a collaborative virtual environment," in *Proc. Virtual Real Annu Int Symp.*, pp. 104–111, 1999.
- [97] K. Jeffay, T. Hudson, and M. Parris, "Beyond audio and video: multimedia networking support for distributed, immersive virtual environments," in *Euromicro Conf.*, pp. 300–307, 2001.
- [98] H. Arioui, A. Kheddar, and S. Mammari, "A predictive wave-based approach for time delayed virtual environments haptics systems," in *IEEE Robot and Human Interactive Communication*, pp. 134–139, 2002.
- [99] C. Carignan and P. Olsson, "Cooperative control of virtual objects over the Internet using force-reflecting master arms," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1221–1226, 2004.
- [100] J. Cheong, S.-I. Niculescu, A. Annaswamy, and M. Srinivasan, "Motion synchronization in virtual environments with shared haptics and large time delays," in *Eurohaptics Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 277 – 282, 2005.
- [101] J. Kim, H. Kim, M. Manivannan, M. A. Srinivasan, J. Jordan, J. Mortensen, M. Oliveira, and M. Slater, "Transatlantic touch: A study of haptic collaboration over long distance," *Presence: Teleoperators and Virtual Environments*,

vol. 13, pp. 328–337, 2004.

- [102] C. Gutwin, S. Benford, J. Dyck, M. Fraser, I. Vaghi, and C. Greenhalgh, “Revealing delay in collaborative environments,” in *ACM Conf. on Computer-Human Interaction*, pp. 503–510, 2004.
- [103] G. Franklin, J. Powell, and M. Workman, *Digital Control of Dynamic System, 3rd Edition*. Addison-Wesley, 1998.
- [104] X. Shen, J. Zhou, E. Saddik, and A. Georganas, “Architecture and evaluation of tele-haptic environments,” in *IEEE Int. Symposium on Distributed Simulation and Real Time Applications*, pp. 53–60, 2004.
- [105] M. Araki and K. Yamamoto, “Multivariable multirate sampled-data systems: State-space description, transfer characteristics, and Nyquist criterion,” *IEEE Transactions on Automatic Control*, vol. 31, pp. 145–154, 1986.
- [106] J. D. Hwang, M. D. Williams, and G. Niemeyer, “Toward event-based haptics: Rendering contact using open-loop force pulses,” in *IEEE Spectrum*, pp. 24–31, 2004.
- [107] S. E. Salcudean and T. D. Vlaar, “On the emulation of stiff walls and static friction with a magnetically levitated input/output device,” *Transactions ASME J. Dyn. Syst*, vol. 119, no. 1, pp. 127–132, 1997.
- [108] N. Melder and W. Harwin, “Improved Rendering for multi-finger manipulation using friction cone based god-objects,” in *Proc. Eurohaptics Conf.*, pp. 82–85, 2003.