FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa
L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Jilin Zhou
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical and Computer Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Velocity and Acceleration Estimation and Uniform Hardness Perception in 6-DOF Haptic Rendering
TITRE DE LA THÈSE / TITLE OF THESIS

The Late Nicolas Georganas
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Emil Petriu
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Dorina Petriu

Shervin Shirmohammadi

Jiying Zhao

Daniela Constantinescu
University of Victoria

Gary W. Slater
Le Doyen de la Faculté des études superieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Velocity and Acceleration Estimation and Uniform Hardness Perception in 6-DOF Haptic Rendering

by

Jilin Zhou

A dissertation submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Electrical and Computer Engineering

The Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-73920-4
Our file Notre référence
ISBN: 978-0-494-73920-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

With advances in human computer interface technologies, the sense of touch has been successfully brought into many virtual reality applications. Haptic devices are such kinds of interfaces which can generate programmable forces to its users, allowing for natural manipulations of objects inside virtual environments. In this thesis, we study two critical issues with haptic interfaces that have a tool-like end effector.

The first issue is about velocity and acceleration estimation for haptic interfaces. Since most haptic devices are equipped with neither tachometers nor accelerometers, this information has to be derived from discrete position measurements. However, due to a limited sensor resolution and a fast update rate requirement of haptic simulations, getting accurate and robust estimates is challenging. In studies of human arm/hand trajectories, researchers observed that skilled movements are usually extremely smooth that is equivalent to minimizing an objective measure with a performance index of jerk. With the assumption of a constrained minimum-jerk model for human arm/hand trajectories, an adaptive constrained 4-state Kalman filter is proposed for the velocity and acceleration estimation for haptic devices.

The second issue is about the influences of a haptic interface's effective mass and viscous damping on users' perceived virtual stiffness in 6-DOF haptic rendering. In any haptic simulations, we always hope that the desired force can be delivered to the user faithfully. But the inherent dynamics of the device may have negative effects on the final perceived force. Through experiments, we have found that the induced forces from those physical parameters affect the perceived virtual stiffness non-uniformly: a surface that should feel uniformly stiff will feel softer or harder, depending on which part of the virtual tool it contacts. The scale of the non-uniform perception is also related to the user's holding point as well as its relative position to the contact points on the tool. Nevertheless, the goal of this study is not to increase the perceived hardness but to make users perceive a uniform hardness when tapping a virtual object. For that, we propose three different methods to compensate for the non-uniform hardness perception. The subjective evaluations show the effectiveness of all three methods.

Acknowledgements

After over nine years' study at University of Ottawa, I finally realized my long-time dream, to be a Ph.D. It is my sincere pleasure to thank those who helped me to make this dream come true.

First, I must thank my supervisor, Prof. Nicolas D. Georganas for his unending support, encouragement, and guidance. Very sadly, Dr. Georganas suddenly passed away just three weeks before my defence. Dr. Georganas has been my research advisor since I was an undergraduate student in 2002. Dr. Georganas is a person who had devoted his whole life to the job. He was so passionate about the research in multimedia and was always thinking about the future. To his students, he was always there to listen and to give advice. No matter what happened, we could come to him and count on him. Certainly, Prof. Georganas will continue to live in our memories.

I am heartly thankful to Prof. Emil M. Petriu for his kindness to be my co-supervisor at a critical stage of my thesis. He showed me practical ways to formulate a research problem and the need to be persistent to accomplish any goals. Many thanks for his ideas, guidance, and support.

I would like to thank Francois Malric for many discussions in his office and being willing to listen to those obscure ideas and proofs. I appreciate his adminitrative work and other colleagues in the DISOCVER Laboratory for creating a friendly and productive research environment. I will remember every morning's coffee break with him, and David Chen. I would like to show my gratitude to Xiaojun Shen who has also helped me get this far.

Without the family's understanding and support, it is impossible for me to finish this long journey as well. I thank my wife, Jing An, for the generous love and patience she has given me in the past ten years. Half of the degree belongs to her. I thank our lovely son Zheyuan (Alex) for his smiles and occasional silly jokes. I will never forget the very first question he asked at the day of my defence when I picked him up at a summer camp. "Did you pass, Daddy?" The poor boy was worrying about it the whole day. Finally, I would like to thank my brother Binglin, and my sister Meiling for their reinforcement and consistent support. In particular, without my brother's support, I could not have the opportunity to come to Canada or the University of Ottawa in the first place.

To,
my wife Jing, and our son Zheyuan (Alex),
and
in loving memory of my parents.

Contents

List of Tables

List of Figures

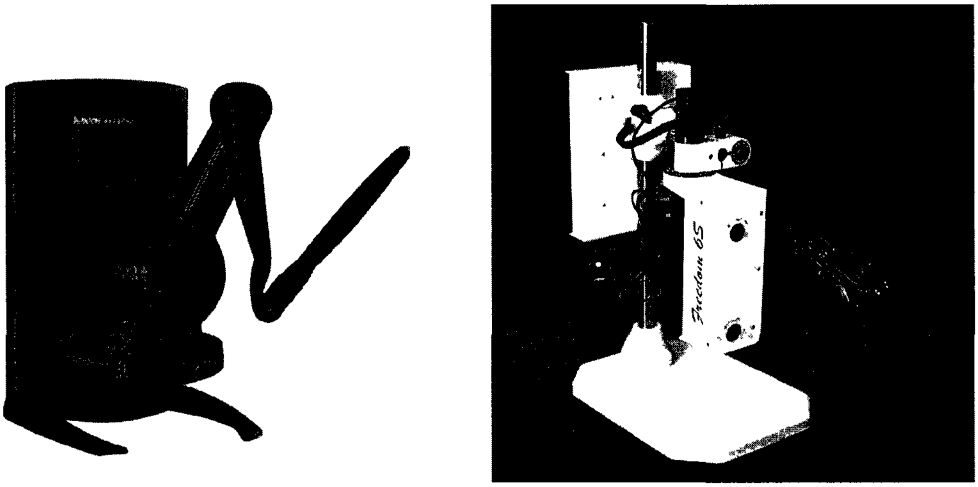
Chapter 1

Introduction

1.1 Introduction to Haptics

Haptic, originally from the Greek verb "haptesthai", pertains to the sense of touch. Haptic interfaces are such a kind of human computer interfaces which can generate programmable forces to stimulate human touch channels, so they allow a natural manipulation of virtual objects in a virtual reality (VR) application or remote physical objects with the help of a tele-robotic manipulator equipped with force/torque sensors [1] [2]. In the past two decades, the study of touch, referred to as haptics, has been a fast-growing research field for VR and tele-operation applications. This research can be broadly categorized into the following four sub-fields [3] [4]:

- Human haptics: Human haptics are mostly conducted by professionals in Psychophysics to study mechanisms of the underlying human touch modality and motor characteristics. It is well known that human haptic perception has two distinct sensing components: (i) *tactile* information from skin touch sensors which provide data about contact force, local geometric profile, texture, and temperature of the touched object-area, and (ii) *kinesthetic* information about the positions and velocities of the kinematic structure (bones and muscles) of the body [5] [6]. Understanding the related human factors is by far the most important thing for the design of those haptic interfaces and haptic-enabled applications, particularly when trade-offs have to be made due to the limitations of current hardware technologies.
- Machine haptics: As its name implies, machine haptics refers to the design and the control of haptic interfaces. Corresponding to the two sensing components, there



(a) (b)

Figure 1.1: Two commercial kinesthetic haptic interfaces: (a) PHANToM desktop from SensAble Technologies Inc.; and, (b) Freedom 6S hand controller from MPB Technologies Inc.

are mainly two kinds of haptic interfaces for each of them. In this thesis, we only focus on the kinesthetic haptic interfaces such as PHANToM desktop devices and Freedom 6S hand controllers, as shown in Fig. 1.1. A kinesthetic haptic interface usually has a pen-like stylus (end effector) connected to its active end, which can be held by a user as a tool to manipulate the interacting objects.

- Computer haptics: Computer haptics covers all research aspects in the development of haptic enabled applications to replicate real-world interacting forces for a particular task. For example, to simulate a cut in a surgery simulation, we need to know when the virtual scalpel touches the skin, how to deform the skin after the touch, when the skin starts to break, how to model the interaction force during each stage of the cut, *etc.* At a higher level, trade-offs are often needed between the stability of the simulation and the haptic perception of the user.
- Multimedia haptics: Multimedia haptics is relatively a new concept which considers haptic data such as users' movement and interaction force trajectories as a new medium type, same as its counterparts audio and video [4]. Multimedia haptics is of particular importance in networked haptic applications where the haptic data

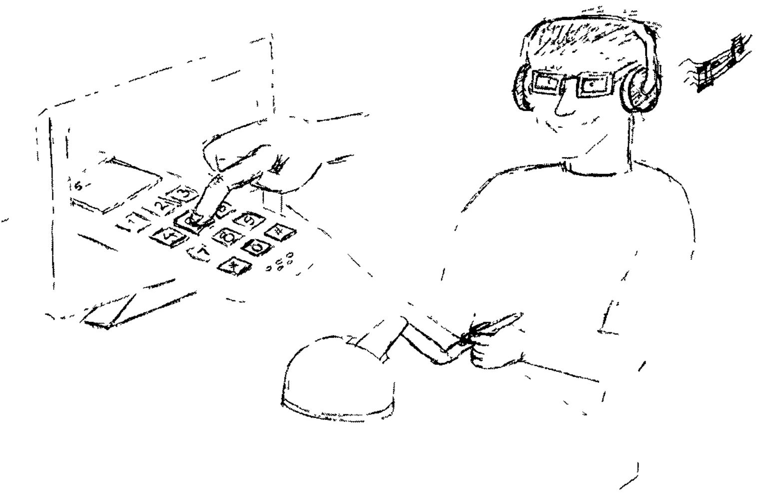


Figure 1.2: A conceptual view of a multi-modality virtual reality application.

have to be transmitted over some communication networks. At each peer site, the received haptic data along with other multimedia data from remote peers have to be reconstructed in a meaningful way. Accordingly, its format, storage, compression, transmission, and synchronization with other multimedia data need to be studied [7] [8].

1.2 Haptic Enabled Virtual Reality and Its Applications

The term "virtual reality" was initially referred to "immersive virtual reality". In an immersive VR environment, the user is fully immersed in a three-dimensional world completely generated by computers. Webster's Dictionary defines *virtual* as "being in essence or effect, but not in fact" and *reality* as "the state or quality of being real". If we combine them together, it means "something virtually existing". Nonetheless, the most accepted definition of VR so far is "virtual reality is a way for humans to visualize, manipulate and interact with computers and extremely complex data" [9]. As shown in Fig. 1.2, the visualization here not only refers to the visual output of a virtual world but also to the computer generated auditory or some other sensual outputs such as force and tactile signals. We explicitly put "haptic-enabled" in front of the "virtual reality" to emphasize its fundamental difference from the visual and audio modality. For the haptic channel, there is a bi-directional information flow between the user and the interacting

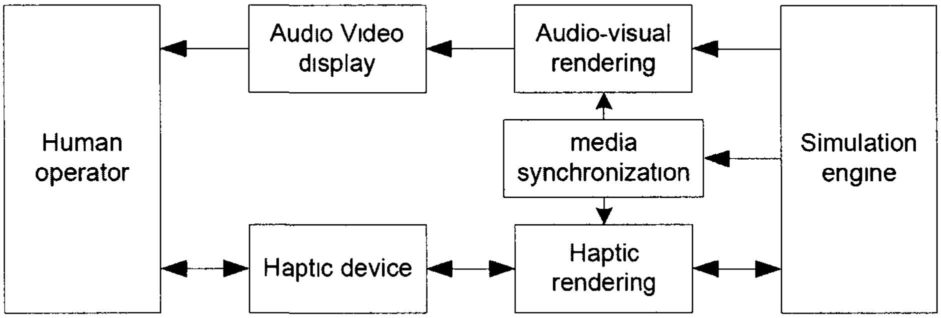


Figure 1.3: System architecture of a haptic-enabled VR application. (adapted from [10])

virtual environment. This bi-directional information flow also involves physical energy exchanges between the user and the haptic interface. But for the visual or the audio modality, the information flow is in one way where the user is mostly a passive observer.

Fig. 1.3 illustrates the system architecture of a typical haptic enabled VR application which consists mainly of four components. The first one is the simulator engine which controls the progress of the application along its life cycle and determines the behavior of the virtual environment in response to any internal or external events. The second component is responsible for the computing and the rendering of haptic and graphic responses from the virtual environment to the user. For a convincing haptic perception, haptic algorithms should be fine enough to capture our real-world touch experience in a form of force/torque feedback. The third component is the media synchronization unit which synchronizes the presentation of the haptic, audio, and video signals to the user. In contrast to the audio-video synchronization, haptic-audio, haptic-video, or haptic-audio-video synchronization are much less studied and may have been overlooked. Research has shown that the inter-frame delay between the haptic and the visual signals has considerable influences on the perceived stiffness of the simulated virtual objects [11]. Furthermore, as more networked haptic applications are being developed, the importance of the media synchronization among the haptic, the audio, and the video parts is getting more obvious. To deal with a network with a limited bandwidth and impairments such as delay and packet losses, these media need to be tagged with different priorities and handled differently at the receiver side for a meaningful reconstruction. Some initial attempts can be found at [12] and [13] where new communication protocols are designed specifically for these networked haptic applications. The final component of a haptic enabled VR application contains all physical input/output interfaces which allow us to interact with virtual environments.

In the past two decades, haptic interfaces have been applied in many industries including medicine, entertainment, education, the arts, and so on. Among all these applications, haptics in medicine has attracted the most attention. It is so natural to integrate haptic interfaces into surgical training simulators such that medical students or surgeons can get certain experiences on new skills or procedures before they practice on real patients [14] [15]. For minimally invasive surgeries, haptic feedback not only helps doctors to have a better control of the inserted surgical tools, but also allows them to palpate the internal organs which may be useful for locating abnormal regions [16] [17]. With advanced haptic interfaces and communication networks, doctors in big medical centers which are mostly located in metropolises can reach those patients living in the countryside with remote diagnosis, or even can teach the local doctors in a hand-by-hand telementoring manner [18] [19]. Haptic-enabled virtual reality also provides a new approach for post stroke rehabilitations [20] [21]. For educational purposes, haptic interfaces have been used as an experimental tool for teaching university courses such as system dynamics and control, and structural molecular biology [22]. Haptic interfaces can also find their usefulness in industrial computer-aided design (CAD) modeling and fine arts, such as the SensAble FreeForm system for product design and the ClayTools system for sculptural modeling. In the entertainment field, just look at the tremendous success of Nintendo Wii products; with some vibrators and accelerometers inside, how many sets have they been sold around the world? Think about having real key-press perception on screen-keyboards when you type a message to your friend using a stylish cell phone. Without mentioning the experimentally supported results of much less error prone typing by having force feedback, this is just plain "cool".

1.3 Haptic Rendering

Just like graphic rendering for computer graphics, which is the process of generating an image from a model through computer programming and monitors, haptic rendering is the process of generating a force from a user's interaction with a model through a haptic interface by means of programming. In [10], K. Salisbury *et al.* give a more formal definition for the rendering as "the process by which the desired sensory stimuli are imposed on the user to convey information about a virtual haptic object."

As shown in Fig. 1.4, the haptic rendering can be broken down into three major modules: collision detection (CD) algorithms, force response algorithms, and control algorithms [10]. In each rendering loop, the CD module determines the collision status

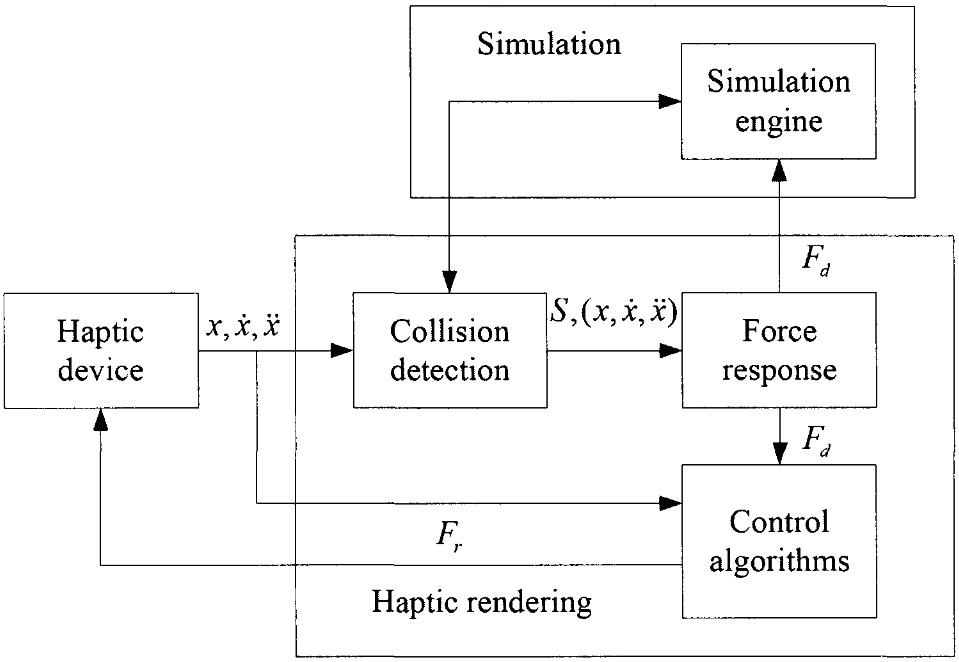


Figure 1.4: Implementation modules of haptic rendering. (adapted from [10])

between a virtual representation of the tool and its neighboring virtual objects. The movement of the tool inside the VE is often directly coupled with the end effector of the device which is held by the user. Once the collision status is determined, a set of contact points or constraints S (depend on the actual haptic rendering algorithm) along with the position of the tool and its derivatives is sent to the force response module for the computation of the interacting force F_d. With Newton's third law, there is a pair of equal and opposite forces acting on the two interacting objects. Therefore, on one side of the force response module, F_d is fed to the simulator engine which may dynamically move the interacting object according to some physical laws. On the other side, F_d is sent to the control module so it can be rendered to the user through the installed motors on the device. The control algorithms regulate F_d to an actual rendering force F_r commanded to the device taking into account of the device's capabilities. For example, if the desired interaction force F_d is larger than the maximum output force of the device, it has to be clamped or interpolated. Or we may want to compensate the device's inherent friction to improve the transparency between the user and the interacting virtual environment [23].

It is generally agreed that the haptic rendering process should be maintained at 1 KHz or even a higher update rate, in order to deliver a convincing force feedback, particularly for the rendering of stiff objects. By contrast, it only requires 30 frames per second for a smooth graphic rendering. As a result, the CD module soon becomes a bottleneck of the process as the graphic model of the scene gets more complex. One workaround is to exploit the spatial coherence of the device's movement. Gregory *et al.* develop a CD library called H-Collide specifically for the haptic rendering in which the graphical model of the scene is constructed into a two-level hybrid hierarchy in pre-processing [24]. The graphical model is first partitioned on a uniform grid. Then, an oriented-bounding-box (OBB-tree) hierarchy for the triangles in each grid cell is constructed. The CD also takes two steps. The virtual tool is first traced against the grid cells to find out current colliding cell and its contained OBB-tree. After that, the collision to the triangles stored in that OBB-tree is tested. The high spatial and temporal coherences of the device's movement make the spatial partitioning an excellent culling tool. Glencross *et al.* further exploit these coherences and propose a haptic caching technique in which only the objects in the locality of the haptic device are maintained for the CD [25]. Another workaround is to take the time consuming CD module out of the haptic control loop and run it in a slower pace [26] [27]. Once the current collision status is resolved, it is swapped into the haptic control loop. Most of time, a haptic control algorithm simply consists of a proportional-derivative (PD) controller with parameters optimized for a specific device and some safety check-up routines, so it can be well maintained at 1 KHz or a higher rate. In this case, the actual force commanded at each frame is then based on the interpolation between the current and the previous collision statuses.

In the real world, we use different kinds of tools for different tasks. These tools are very diversified in their geometric shapes and sizes. Based on the virtual representation of a tool in the CD, haptic rendering can be classified into two types: point-based, and tool-based. With the point-based haptic rendering, the tool is modeled as a single point inside the CD. This virtual point follows the movement of the end point of device, so it is often referred to as haptic interface point (HIP). As the user moves the end effector, HIP could penetrate the virtual objects that are meant to be impenetrable. Therefore, the goal of the CD algorithms is to detect such penetrations and determine the position of a constraint point on the surface of the penetrating object that is closest to the HIP. If there is no penetration, the constraint point is directly set to the location of the HIP. In two most popular haptic rendering algorithms, this constraint point is called the god-object and the proxy respectively [28] [29]. Once the location of the constraint point is figured out, the feedback force $F \in \mathbb{R}^3$ is then computed using Hooke's law

$$F = kd, \tag{1.1}$$

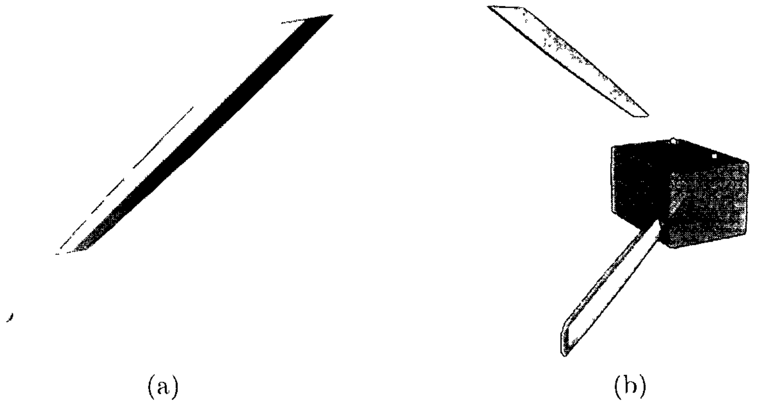
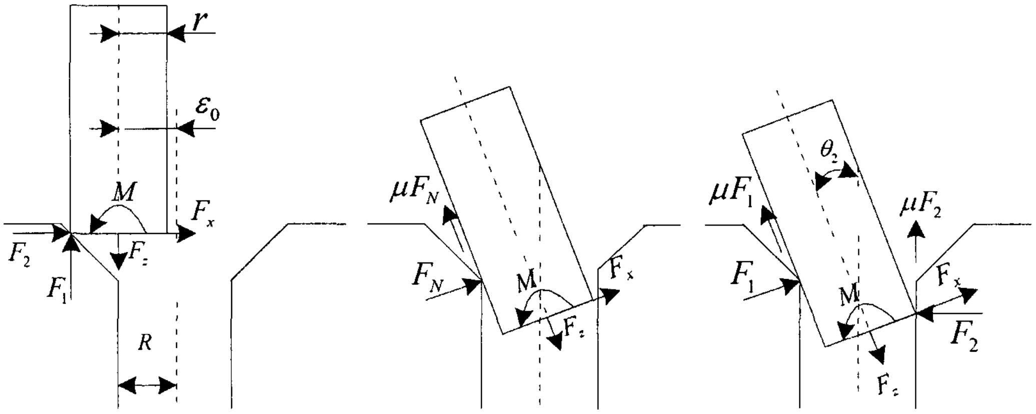


Figure 1.5: Graphical representation of a tool for point-based haptic rendering: (a) A typical graphical representation of the tool inside the virtual environment. The tip is disconnected from the handle to emphasize that the tip alone is subjected to force-feedback; and, (b) The orientation of the tool has no effects on the feedback force in point-based haptic rendering [30].

where $k \in \mathbb{R}$ denotes the stiffness of the virtual objects, and $d \in \mathbb{R}^3$ is the penetration vector from the HIP to the corresponding constraint point. It is obvious that the deeper the penetration, the larger is the feedback force. If there is no penetration ($d = \vec{0}$), the feedback force becomes zero. The point-based haptic rendering is sometimes called 3-DOF haptic rendering because it only requires a 3-DOF haptic interface (no twisting force required on the end effector). It is important to note that the orientation of the end effector has no effects on the feedback force in the point-based haptic rendering, so it makes no difference to the user how the tool is actually held. As shown in Fig. 1.5b, both representations would result in the same feedback force at the end tip no matter the tool is "inside" or above the surface.

Despite the fact that general tool object interactions can for some applications be adequately captured using a one-point contact model where the 3-DOF point-based haptic rendering can be applied [31], in many real-world tasks the former approach would prove insufficient [32]. Essentially, the following four aspects should be kept in mind for more general tool-object interactions:

- First, contact types between the tool and the interacting object are varied. For example, the peg-in-hole operation is the most frequent mechanical assembly task. As shown in Fig. 1.6, the collision status between the peg and the mating hole can



Chamfer crossing One point contact Two point contact

Figure 1.6: Three stages in a peg-in-hole operation.

be a one-point, a two-point, a surface-surface contact, or a combination of them during the operation [33] [34].

- Second, feasible trajectories of the tool are embedded in a 6-D space with both translational and rotational constraints. In contrast, it only can constrain the motion of a single point with the point-based haptic rendering. Therefore it is not suitable for these kinds of general tool-object simulations.
- Third, the user's handling postures of the tool are time varying. In the real-world, the objectives of each task usually determine how the tool is held, the number of fingers involved, and the time-varying forces exerted by the different fingers [35].
- Last, torque generated by the contact force around the user's holding point provides important information about the interaction. In fact, for the peg-in-hole operation, the torque feedback is essential for a successful completion of the task [33].

These four aspects not only increase the complexity of the haptic modeling but also require the use of haptic interfaces capable of providing both force and torque (twist force) on the stylus. Consequently, it is often referred to as 6-DOF haptic rendering.

1.4 Motivations and Problem Statements

In this thesis, we study two critical issues of those haptic interfaces having a stylus (end effector) attached to their active end to allow users to hold as a tool. The first issue is about velocity and acceleration estimation for haptic interfaces. More precisely, it is the linear velocity and acceleration of the haptic interface's end effector. Since most currently available haptic devices are not equipped with tachometers or accelerometers, this information has to be derived from discrete position measurements. The second issue we studied is about the influence of a haptic interface's effective mass and viscous damping at its active end on the perceived stiffness of the simulated virtual objects in 6-DOF haptic rendering. The motivations and problem statements are described in the following.

1.4.1 Velocity and acceleration estimation for haptic interfaces

In haptic VR applications, the velocity and acceleration information of haptic interfaces is needed in many places. In many research aiming at increasing the perceptual hardness of virtual objects, the velocity and acceleration information is needed for the generation of transient high frequency signals which are overlaid on top of the classic penalty-based force [36] [37]. The velocity and acceleration information is also important to many control algorithms for bilateral teleoperation. In his well-known four-channel architecture for the bilateral teleoperation, Lawrence proves that the velocity and acceleration information of both master and slave manipulators is needed for a full transparency between the user and the remote interacting environment [38]. For networked haptic applications, local haptic data are transmitted over some networks where the network delay, jitter, and packet loss potentially affect the overall system performance. The velocity and acceleration information can be used to predict or compensate the delayed haptic data [39] [19]. To be network friendly, the velocity and acceleration information can be exploited for the compression of the haptic data to reduce the required bandwidth as well [40].

However, since most currently available haptic devices are not equipped with tachometers or accelerometers, this information has to be derived from discrete position measurements. Due to the limited sensor resolution, the high maneuverability of human arm/hand, and the high update rate requirement of haptic simulations, getting a precise and robust velocity and acceleration estimation is very challenging. To understand the nature of the problem, let us consider a joint encoder with a resolution of $\Delta = 0.003rad$ attached to a motor shaft. N denotes the number of counter increments during a sampling interval $T = 0.001s$. At time kT , the joint velocity estimated with the first-order difference method is

$$v = \frac{\theta(kT) - \theta((k-1)T)}{T} = \frac{N\Delta}{T}, \tag{1.2}$$

which gives a joint velocity resolution of $\Delta/T = 3rad/s$. For a lever with a length $L = 100mm$ connected to the joint, the linear velocity resolution at the lever end becomes

$$\Delta/T = 300mm/s. \tag{1.3}$$

The higher the sampling rate, the worse is the velocity resolution. The acceleration estimation with the first-order difference method is even worse, which has a resolution of

$$\Delta/T = 300m/s^2. \tag{1.4}$$

Apparently, the simple first-order difference method is doomed to fail because the high frequency noise components are amplified.

Essentially, the basic criterion of the velocity and acceleration estimation from discrete position samples is to compress the high frequency noise at the low speed while capture the transient behavior of the primary signal (the measured position signal) at the high speed. Therefore, the filtering parameters should be relied on the trajectory itself. On the other hand, sophisticated filters often have so-called model-filter mismatch problems that their performance degrades significantly when the actual motion gets away from the assumed model.

In studies of human arm/hand trajectory behaviors, researchers found that skilled movements are usually extremely smooth that mathematically equal to minimizing an objective measure with a performance index of jerk. Jerk is mathematically defined as the derivative of acceleration. For constrained movements, Daohang *et al.* studied reaching movements of human arm on a constraint semi-sphere surface in [41]. The experimental results are also consistent with the theoretical analysis of the minimum jerk reaching on the surface of a sphere. With the assumption of a constrained minimum-jerk model of human arm/hand trajectory planning, we propose an adaptive constrained 4-state Kalman filter to estimation the velocity and acceleration information for haptic devices.

1.4.2 Uniform hardness perception for 6-DOF haptic rendering

In any haptic applications, we always hope that the desired force/torque from a simulation can be faithfully displayed to the user. However, different from our real world

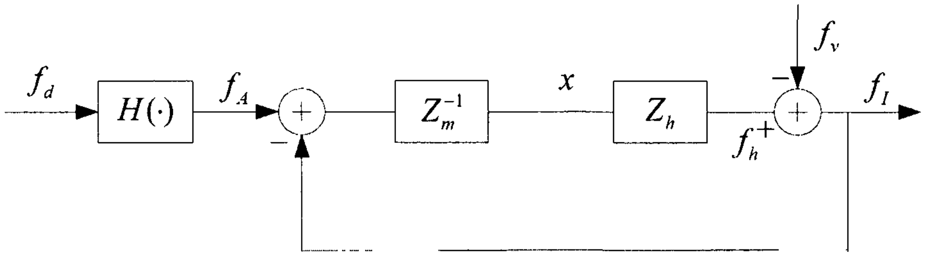


Figure 1.7: Close control loop of a 1-DOF haptic interface. (adapted from [23])

experience, the desired force/torque here is generated by actuators installed on the device where both electrical characteristics of the motors and mechanical dynamics of the device have effects on the final force/torque perceived by the user. In this thesis, we always assume the actuators are ideal and only focus on the effects of the mechanical dynamics.

Let us take a 1-DOF haptic interface with an impedance of Z_m as an example to explain the effects of the physical parameters on the user perceived force [23]. The closed control loop is illustrated in Fig. 1.7 where Z_h represents the user's impedance. The commanded force to the actuator f_A is set to be the desired force f_d from the simulation ($H(\cdot) = 1$). The actuator force f_A, combined with the interaction force f_I between the user and the device, acts on the device. The resulting device motion x then moves the user's hand that, in turn, generates a feedback force f_h to the user. This feedback force is further added to any voluntary forces f_v that the user exerts on the device to form the interaction force f_I felt by the user. The transfer function T_0 , from f_d to f_A is then given by

$$T_0 = \frac{Z_h/Z_m}{1 + Z_h/Z_m} = \frac{Z_h}{Z_h + Z_m} = Z_h Z_T^{-1}, \tag{1.5}$$

where $Z_T = Z_h + Z_m$. Apparently, if we want to have $f_A = f_I$, T_0 has to be unity, in turn, that requires Z_m being zero. But for a real haptic interface, Z_m will never be zero because of both linear terms such as mass and viscous damping and non-linear terms such as Coulomb friction. Intuitively, the physical parameters are preferred to be small so as to get a good transparency between the user and the interacting virtual environment [42]. If not, active compensation will be needed [23] [43] [44].

The second issue we studied is about the influences of a haptic interface's effective mass and viscous damping at its active end on the perceived stiffness of the simulated virtual objects in 6-DOF haptic rendering. In this thesis, the 6-DOF haptic interfaces under the study is assumed to be a serial linkage type. Unlike those studies on improving the perceived hardness of virtual objects [37] [45] [36] [46], our goal is to make users have a uniform stiffness perception when tapping a virtual object in 6-DOF tool-based haptic rendering. The effective (perceived) mass is the weight of the device perceived by the user as he/she holds the end effector. In this thesis, the Coulombic friction is not considered simply because it is very small for the haptic device we are using for the experiments. If the Coulombic friction is not negilible, it should be included in the compensation model. Indeed, to accurately measure the Coulombic friction for a 6-DOF haptic interface is not an easy task.

As discussed, in the 3-DOF haptic rendering, the tool is modeled by a single point in the haptic model and this point is also at the fixed location on the tool. Consequently, the induced forces from these physical parameters have the same effects on the final perceived force by the user. But in 6-DOF haptic rendering, the tool is modeled by its real shape. The collision points between the tool and the virtual objects can also be anywhere on the tool. Specifically, we have found that the induced forces affect the perceived harness non-uniformly as users tap a virtual object with different parts of the tool. This non-uniform perception is also related to the position of users' holding point and its relative position to the collision points. The detailed problem statement is presented in Chapter 4.

1.5 Contributions

In this thesis, we claim the following three main contributions to the field of haptics:

- A novel algorithm to estimate velocity and acceleration of haptic interfaces taking into account the behavior of the human arm/hand trajectory.
- An interesting issue in 6-DOF haptic rendering is identified for serial linkage haptic interfaces: a surface that should feel uniformly stiff will feel softer or harder, depending on which part of the virtual tool it contacts.
- Three methods are proposed to compensate the effects of a haptic interface's effective mass and viscous damping on the uniform stiffness perception of virtual objects in 6-DOF tool-based haptic rendering.

1.6 Publications Arising from This Thesis

The following papers have published for this work

- **J. Zhou**, F. Malric, E. M. Petriu, and N. D. Georganas "Uniform hardness perception for 6-DOF haptic rendering," *IEEE Trans. Instrum. Meas.*, Mar. 2010 (to appear).
- **J. Zhou**, F. Malric, S. Shirmohammadi, "A new hand Measurement method to simplify calibration in CyberGlove-based virtual rehabilitation," *IEEE Trans. Instrum. Meas.*, Nov. 2009 (to appear).
- N. Sakr, **J. Zhou**, N. D. Georganas, and J. Zhao, "Prediction-based haptic data reduction and transmission in telementoring systems," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1727-1736, Aug. 2009.
- X. Shen, **J. Zhou**, A. Hamam, S. Nourian, N. R. El-Far, F. Malric, and N. D. Georganas, "Haptic-enabled tele-mentoring surgery simulation: design, implementation and evaluation," *IEEE Multimedia*, vol. 15, no. 1, pp. 64-76, 2008.
- **J. Zhou**, F. Malric, E. M. Petriu, N. D. Georganas, "Compensating device inertia for 6-DOF haptic rendering", in *Proc. 2009 IEEE Int. Conf. Syst. Man Cybern. (SMC'09)*, San Antonio, TX, USA, Oct. 11-14, 2009, pp. 12-17.
- N. Sakr, **J. Zhou**, N. D. Georganas, J. Zhao, and E. M. Petriu, "Network traffic reduction in six degree-of-freedom haptic telementoring systems," in *Proc. 2009 IEEE Int. Conf. Syst.Man Cybern. (SMC'09)*, San Antonio, TX, USA, Oct.2009, pp. 2525-2530.
- **J. Zhou**, F. Malric, S. Shirmohammadi, "Practical calibration for upper extremity patients in haptic rehabilitation," in *Proc. IEEE Workshop Medical Meas. Appl. (MeMeA 2009)*, Cetraro, Italy, May 29-30, 2009, pp. 73-78.
- N. Sakr, **J. Zhou**, N. D. Georganas, J. Zhao, and E. M. Petriu, "Robust perception-based data reduction and transmission in tele-haptic systems," in *Proc. 3rd Joint Eurohaptics Conf. Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Salt Lake City, USA, Mar. 18-20, 2009, pp. 214-219.
- **J. Zhou**, X. Shen, E. M. Petriu, and N. D. Georganas, "Linear velocity and acceleration estimation of 3-DOF haptic interfaces," in *Proc. IEEE Int. Workshop*

Haptic, Audio and Visual Environ. their Appl. (HAVE2008), Ottawa, Canada, Oct. 2008, pp. 137-142.

- **J. Zhou**, E. M. Petriu, X. Shen, F. Malric, and N. D. Georganas, "Modeling contact forces for 3D interactive peg-in-hole virtual reality operations," in *Proc. IEEE Int. Instrum. Meas. Tech. Conf. (I_2MTC)*, Victoria, Canada, May 2008, pp. 1397-1402.

- **J. Zhou**, X. Shen, A. El Saddik, and N. D. Georganas, "Design and implementation of haptic tele-mentoring over the internet," in *Proc. 11th IEEE Int. Symp. Distributed Simulation & Real Time Appl. (DS-RT)*, Chania, Greece, Oct. 2007, pp. 201-208.

1.7 Thesis Overview

We start in Chapter 1 (current chapter) by introducing four active research fields in haptics: human haptics, machine haptic, computer haptics, and multimedia haptics, followed by a brief introduction about haptic enabled virtual reality and its applications. Haptic rendering is then introduced with an emphasis on the differences between the point-based and the too-based haptic rendering. After that, we describe the objectives and the contributions of this thesis.

In Chapter 2, we present the related human factors for the design of haptic interfaces, physical elements of a typical haptic interface, and haptic rendering. Following that, we review the current approaches to increase the perceived hardness in haptic simulations.

Chapter 3 contains our work on the velocity and acceleration estimation for haptic interfaces. We start with the explanation of optical shaft encoders used in most of haptic interfaces for the measurements of joint signals, followed by the theoretical modeling of the position resolution of a haptic interface. We then present some existing methods for the velocity and acceleration estimation from discrete position measurements. After that, human arm trajectory behaviours and their possible roles in the velocity and acceleration estimations for haptic interfaces are discussed. Following that, we present the proposed jerk constrained 4-state Kalman filter and the simulation results.

Chapter 4 presents our work on the uniform hardness perception in 6-DOF haptic rendering. We bring out different roles of user's holding point and motor's control point in 3-DOF and 6-DOF haptic rendering using a stick-on-ball example. Following that, we describe the two situations in which the user will have different hardness perception as he/she taps the virtual object. Three specialized approaches are then presented, followed by the experimental results and discussions.

We conclude the thesis in Chapter 5 with possible future directions as well as closing notes.

Chapter 2

Background and Discussion

2.1 Human Factors in Haptics

2.1.1 Kinesthetic perception

Kinesthetic perceptions include both proprioception and force perception, which originate primarily from two types of muscle afferents: Golgi tendon organs and muscle-spindle afferents. These muscle afferents provide a relatively fast way for our central nervous system (CNS) to acquire information about the static length of muscles, the rate at which muscle length changes, and the force muscles generate.

Proprioception is the perception of positions and movements of the body segments in relation to each other without the aid of the vision, or the touch [6]. Human ability to detect a change, so called Just Noticeable Difference (JND), in the movements of a limb varies on several factors including the velocity, the particular joint moving, and the contractile state of the muscles controlling the moving joint [47]. In general, the JND during faster movements is smaller than that in slower movements [47]. It also decreases when the acting muscles of the joint are actively contracting rather than relaxed. For joints along the arm, our ability to detect a change decreases from the shoulder to more distal joints. For example, the shoulder joint can detect a difference of $0.8°$ whereas the wrist and the elbow can detect a change of $2°$, and the finger joint only can detect a change of $2.5 - 6.8°$ [48] [49]. If we assume a 10 mm fingertip, the JND of the finger joint requires that haptic interfaces should have a position resolution of $10 \times \sin(2.5°) = 0.4mm$ for the finger tip, which is often exceeded by a factor of 4 in practice [50].

Force perception originates from afferent discharges from Golgi tendon organs signaling intramuscular forces that, in turn, provide the CNS with information about the forces produced by muscles [47]. The force JND that a human can reliably discriminate follows Weber's law closely, in which the smallest perceived change in the intensity of a stimulus is proportional to the physical magnitudes of the original stimulus. For forces in the range of 0.5-200 N, the average JND is around 7-10% and increases to 15-27% for forces smaller than 0.5 [47]. This is to say that if a haptic simulation delivers forces at 1 N level, the user should be able to detect a 0.1 N force change. Should the inherent mechanical friction of a haptic interface be higher than this minimum noticeable force, the user would not distinguish the desired force from the non-desired friction component.

On the other hand, a human being is capable of producing a controllable force as large as 100 N at the shoulder for a short time period [48]. Even the proximal-interphalangeal joint of the finger alone can produce around 16.5 N of force. In terms of the resolution, the force produced by muscles controlling the fingers can be reproduced reasonably accurate. Srinivasan and Chen found that the average absolute error remains approximately constant at 0.039 ± 0.006 N for force in the range of 0.25-1.5 N [51]. For mid-range force, the resolution has a mean value of 0.36 N with a standard deviation of 0.07 N [48].

2.1.2 Stiffness perception

Basically, stiffness perception originates from sensory signals conveying both force and movement information. Compared with the force perception, the JNDs for the stiffness are much coarser. Jones and Hunter found that the JND for stiffness (23%) is about three times than that for force, position, and movements (6-8%) [52] [53]. Their experiments use two electromagnetic linear motors which mechanically are equivalent to linear springs. On each experimental trial, the stiffness of the reference motor connected to the right arm was randomly set at one of eight different amplitudes in the range of 0-6260 N/m. Subjects were asked to adjust the motor connected to the left arm to match the stiffness felt by the right arm. In [54], Forrest *et al.* report that there are 2-3 perceptible levels between 0.2 and 0.5 N/mm when palpating virtual surfaces with the fingertips. In [55], Tan *et al.* experimentally determine that the minimum stiffness required to simulate a rigid object without visual feedback is 24.2 N/mm. However, it is very challenging to design a haptic interface with such a capability. In fact, most of currently available haptic interfaces can only render a virtual surface less than 2 N/mm. In [56], the differential threshold in human stiffness perception is reported to be 0.08-0.12 for unrestricted active probing onto a virtual surface using a PHANToM desktop device.

2.1.3 Kinesthetic system bandwidth

As aforementioned, the modality of touch includes both the cutaneous and kinesthetic sensory systems. The cutaneous sensory system receives stimuli from the mechanoreceptors embedded in the skin and has a perceptible frequency range of 100-1000 Hz, with peak sensitivities around 200-300 Hz [57]. But the kinesthetic sensory system only has a bandwidth up to 20-30 Hz. Furthermore, a human being has a much lower bandwidth for the kinesthetic control system. The control bandwidth here means the speed humans can respond to. In general, the maximum frequency of voluntary movements is limited by the muscle damping and the limb inertia. P. D. Nelson experimentally determines that the voluntary movements of the elbow have a maximum attainable frequency of 4-6 Hz for freewheeling tests, and a frequency range of 0.1-2 Hz with a maximum power peak concentrated about 0.5 Hz for tracking tasks without visual feedback [58]. Voluntary movements involving the hand and the arm contain frequencies below 2 Hz for tactile exploratory tasks, but can have peak frequencies in between 4 and 8 Hz for manual skills such as writing, tapping, and shading [59]. A slow performance range for the tactile exploratory tasks is due to the temporal requirements of the sequential sampling process from the mechanoreceptors. For a haptic application, the bandwidth of the haptic interface and its controller must have a larger system bandwidth to meet the human input/output bandwidth.

2.2 Haptic Interfaces

As mentioned, a haptic interface is a kind of human computer interface which allows users to "feel" simulated virtual objects in virtual environments, or experience "hands-on feelings" while manipulating physical objects with the help of a tele-robotic manipulator. The commercial haptic interfaces usually provide two or more DOFs in the position tracking of the user's motion and apply feedback force in less or the same DOFs. For instance, the PHANToM desktop device (Fig. 1.1a) has six DOFs in the position sensing and three DOFs in the output force, while the Freedom6S hand controller (Fig. 1.1a) features six DOFs in both position sensing and output force on the end effector.

One of the important performance measures for haptic interfaces is the maximum achievable stiffness, *i.e.*, how rigid a virtual object can be simulated with the device? An ideal haptic interface should render both a rigid object (stiffness $\rightarrow \infty$) and free space motions (stiffness $\rightarrow 0$) equally well. However, it is limited by many factors including the motors' capabilities, the sensors' resolution, the joints' friction, the integrity of the arm structure, the update rate of the controller, *etc.* [60] [61]. Yet, there have been many research efforts devoted to improving the perceived hardness (rigidity) in haptic simulations. Later on in this chapter, we will review these studies in much detail.

Two other important measures for haptic interfaces are the perceived (or effective) mass, and the Coulomb friction present at the active end of the device. These are preferred to be small so as to get good transparency between the user and the interacting environments [42]. If not, an active compensation is then needed. In [43], a hard-switched model-based friction compensator is presented for a tendon-driven haptic manipulator. When the relative motion between the motor and its driven joint is less than a pre-determined threshold, a pre-sliding spring model is used. Otherwise, a regular Coulomb friction model is used. With a tele-operation setup, the experimental results show that the force-displacement at the operator's side is better matched to that of the touching object at the slave side with the compensation. Inertia compensation is also performed for the slave manipulator used in the experiments to cancel the effective mass of each joint [44]. In [23], a hybrid controller is proposed to compensate for the joint frictions of haptic interfaces. The controller consists of two simultaneously operating compensators: a model-based compensator for higher velocity motions and a variable gain force feedback compensator for lower velocity motions. Their experimental results with a 1-DOF haptic interface show that the free space rendering and the linearity of the applied joint were both improved. Friction compensations for haptic interfaces are also studied in [62], [63], and [64]. On the other hand, compensation for the effective mass or physical damping is less studied for haptic interfaces.

Other important characteristics of haptic interfaces include maximum peak force that the actuators can exert, maximum continuous force, position resolution, workspace, *etc.* [65] [66]. Although haptic interfaces vary considerably in those parameters, the underlying physical design is generally the same. A haptic interface usually includes the following elements [67]:

- Current amplifier: As its name implies, the current amplifier amplifies a low power command current signal to drive a motor. Most of haptic interfaces use either Pulse Width Modulation (PWM) amplifiers to regulate motors' current such as PHANToM series of desktop devices, or analog amplifiers such as Freedom 6S hand controllers. In comparison, the PWM amplifiers consume less power and are often cheaper for an equivalent power output but have a poorer performance due to high-frequency switching noises. It is also desirable that the electrical bandwidth

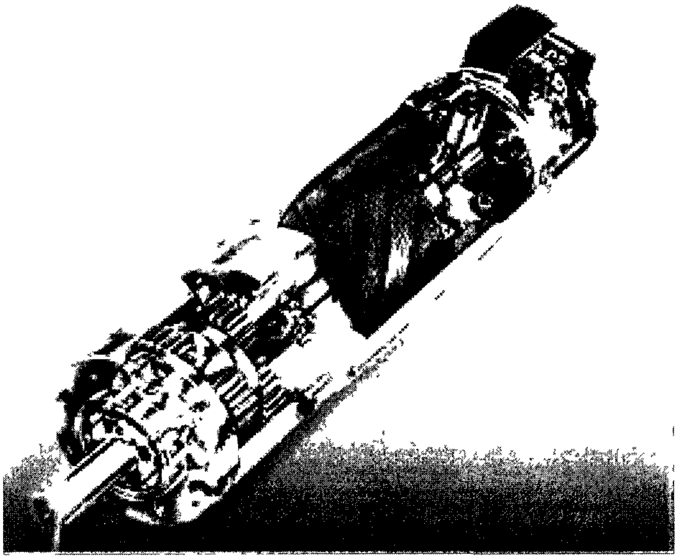


Figure 2.1: A Maxon brushed DC motor with gearhead and optical encoder (http://www.maxonmotorusa.com).

be far higher than the mechanical response of the device [68].

- Motor: Desktop haptic interfaces often use small and brushed DC motors. The current from the amplifier flowing through the motor coils generates a torque on its shaft, to which a capstan is attached. The basic law describe the relationship between the mechanical output torque τ and the flowing electric current I of a DC motor is

$$\tau = K_m \times I, \tag{2.1}$$

where K_m is the motor constant. The larger the flowing current, the larger is the output torque. However, the dissipation of heat coming from resistive losses in the motor coils is proportional to the current squared. If the motor becomes too hot, the insulation of the coils wires will melt or burn. Therefore, the motor temperature should be always monitored using a mathematical thermal model or a physical temperature sensor. For the selection of DC motors, there is also a trade-off between the heat capacity and the mass of the motor [69]. For haptic interfaces, we want the manipulator arm to be as light as possible for a good transparency, but a lighter motor means a lower heat capacity. In addition, the bandwidth of motors is crucial for delivering a fine force feedback to the user. Fig. 2.1 shows a brushed DC motor from Maxon Precision Motors Inc.

- Encoder: There are mainly two types of encoders used in haptic interfaces to measure the angle of each joint: optical shaft encoder and magneto-resistive potentiometers. The optical shaft encoder will be discussed more in Chapter 3.
- Drum: A drum is connected to the capstan of motor using different kinds of cables. The ratio between the drum's and the capstan's diameter is often called gear ratio. The drum's diameter is sometimes made to be larger than that of the capstan ($n > 1$) for two purposes: increasing the motor's torque by n times to have a stronger haptic feedback and amplifying the motor's motion also by n times to get a higher resolution in joint angle measurements. Unfortunately, the effective inertia of the motor and any friction present in the motor are increased by n^2 times at the same time. For high performance haptic interfaces, these negative effects are significant and should be minimized ($n = 1$).
- Linkage and Handle: The drum is attached to the endpoint of the device through a mechanical linkage. The torque produced by the motor on its shaft is transmitted to the user as he/she holds a handle or a thimble at the endpoint of the linkage.
- Force/torque sensors: Some haptic interfaces may have force/torque sensors installed on their end effectors for the measurements of external forces from the user.
- A/D converter and D/A converter: Due to the nature of digital control, a computer reads the sampled angular signal of each joint from an A/D converter and the command electric current for each motor is sent to the corresponding current amplifier through a D/A converter. For high performance haptic interfaces, 16-bit A/D and D/A converters are preferred.

After the description of a haptic interface's physical elements, let us discuss a bit more about the control of the device. In general, there are two paradigms in which a haptic interface can be controlled: impedance control and admittance control.

2.2.1 Impedance control

Fig. 2.2 gives a conceptual view of the impedance control of haptic interface. At the standpoint of the virtual environment, the impedance control can be described as "displacement in - force out". As the user moves the device, the motion is sensed by position sensors (joint encoders). The device reacts with forces if needed. The actual rendering

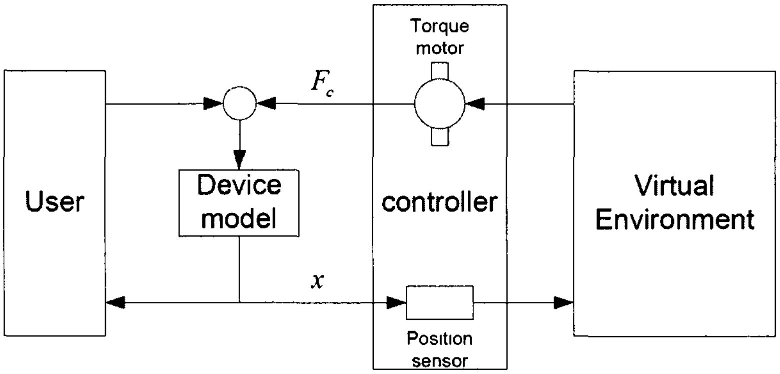


Figure 2.2: A conceptual view of impedance control of haptic interfaces [70].

process for this kind of devices was discussed earlier when we introduced the haptic rendering in Chapter 1. The impedance type haptic interfaces are commonly designed to be back-drivable, low friction and inertia, so they are natural for free space rendering, *e.g.*, the motor force is zero. However, they cannot provide very high feedback force. The PHANToM desktop and the Freedom6S hand controller mentioned above are two typical impedance type haptic interfaces. In the rest of thesis, if not explicitly mentioned, we always refer to the impedance type haptic interfaces.

2.2.2 Admittance control

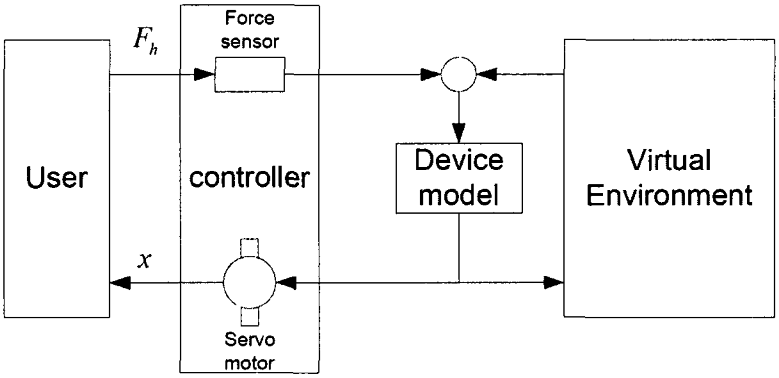


Figure 2.3: A conceptual view of admittance control of haptic interfaces [70].

In contrast to the impedance control, the admittance control is "force in - displacement out". As illustrated in Fig. 2.3, the force exerted by the user is measured by the force sensors installed on the device, and the device reacts with motion if needed. Let

Table 2.1: Comparison between impedance and admittance control haptic interfaces.

	Impedance	Admittance
Fidelity	High	Low
Mas	Low	High
Inertia	Low	High
Output force	Low	High
Cost	Low	High
Force sensing	No	Yes
Position sensing	Yes	Yes

us take a HapticMaster device from FCS Control Systems as an example to explain the concept (Fig. 2.4).

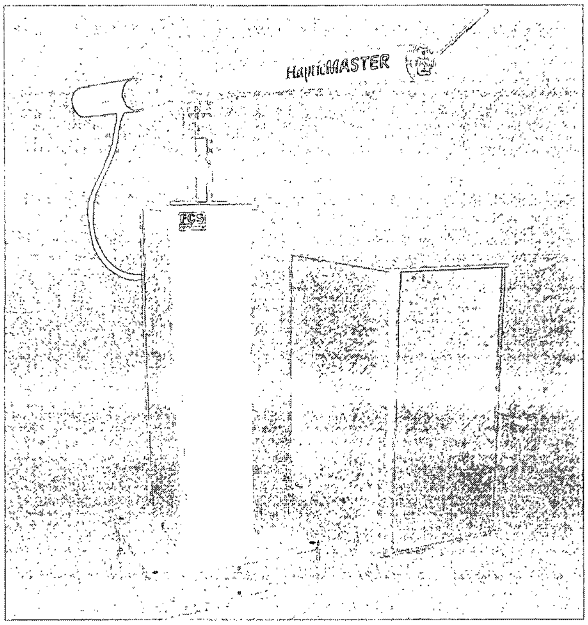


Figure 2.4: A FCS HapticMaster controller from FCS Control Systems.

Assume we want to simulate a virtual mass moving in the free space. With the force exerted by the user, the acceleration of the virtual mass can be computed with Newton's second law, which in turn is numerically integrated into the desired velocity, and then once more into the desired position, altogether forming the trajectory of the mass along the time. After that, the current desired position, velocity and acceleration are sent to a servo control loop which makes the physical device closely follow the trajectory. All of this happens 2500 times per second on a hard real-time operating system, thus the physical device behaves almost exactly as the model, *i.e.*, as a floating virtual mass or whatever inertia was set. To avoid instability caused by large instantaneous accelerations, there is a minimum allowable virtual mass. Practically, if the mass model is interacting with other virtual objects, the interaction forces have to be added to the forces from the force sensors dividing by the model mass, so they work all at the same time. In contrast to the impedance control, the admittance control is clearly more complex. The biggest advantages of the admittance type devices are much higher output force and more rigid virtual surfaces which can be simulated. The HapticMaster device can generate as high as 250 N output force and achieve a virtual stiffness of 50 N/mm while a classical PHANToM 3.0 can only provide a maximum force of 22 N (3 N continuous) and a virtual stiffness of 1-2 N/mm. A simple comparison chart between the impedance and the admittance control haptic interfaces is given in Table 2.1.

2.3 Improving Hardness Perception in Haptic Rendering

The classic haptic rendering is to apply a feedback force proportional to the user's penetration into the interacting virtual object with Hooke's law. Accordingly, one of the important performance measures for haptic interfaces is the maximum achievable virtual stiffness $K_{\max}$, *i.e.*, how rigid a virtual object can be simulated with the device? There have been many research efforts devoted to improving the perceived hardness (rigidity) in haptic simulations. In these studies, the passivity theory has been often used as an analysis tool. For an n-port network system, if it is passive, the following condition is always satisfied:

$$E_A(t) = \int_0^t f(\tau) \cdot v(\tau) d\tau + E_{store}(0) \geq 0 \; \forall t \geq 0, \tag{2.2}$$

where $E_A(t)$ denotes the available energy of the system at t, f and v are conjugate power variables flowing into the system, and $E_{store}(0)$ is the initial energy stored in the system at $t = 0$ [71]. There are two main advantages to use the passivity theory for the analysis of haptic enabled applications. First, a passive system is guaranteed to be stable. There is no need to know the internal behavior of the system, which is often very difficult to model. Consider a haptic application described using two-port network model as shown

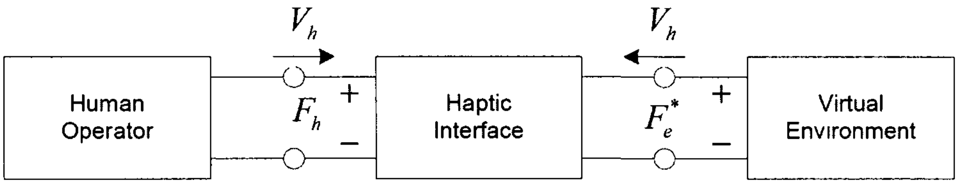


Figure 2.5: Two-port network representation for a typical haptic-enabled application. The star superscript indicates that the variable is discrete [72].

in Fig. 2.5. With classic control theories, the transfer function of the whole system has to be derived to find out the required stability criteria. However, the dynamics of each component are hard to model and usually not linear. With the digital control, the conversion between the digital and analog domains makes the analysis even harder. With the passivity theory, by only monitoring the power variables across the system's boundary, the stability of the system can be concluded. Second, interconnected passive systems are still passive. Therefore, we can determine the stability of the whole system by analyzing each subsystem independently. However, the passivity theory is a very conservative concept. A passive system is guaranteed to be stable but a stable system is not necessary to be passive.

2.3.1 Maximum achievable virtual stiffness

In [60], Colgate and Schenkel derive the maximum achievable stiffness for a 1-DOF haptic interface considering the effects of sampling and zero-order hold. The device was modeled as a rigid body m with some viscous friction b. For a unilateral constraint virtual wall modeled by a spring and a damper in parallel, the condition to achieve passivity is

$$K < \frac{2(b-B)}{T}, \tag{2.3}$$

where K and B are the stiffness of the virtual spring and the coefficient of the virtual damper respectively. Eq. (2.3) states that some physical dissipation through viscous friction is necessary to achieve passivity. It also shows that, for a particular device with fixed physical and virtual damping, the higher the sampling rate, the higher is the maximum achievable virtual stiffness. In [61], Diolaiti *et al.* generalize the condition considering the device's Coulomb friction and the quantization step in the closed control loop. Without a virtual damper, the virtual wall is guaranteed to be passive if

$$K < \min(\frac{2b}{T}, \frac{2c}{\Delta}), \tag{2.4}$$

where c and Δ denote the Coulomb friction and the quantization step respectively. For fixed Coulomb friction, Eq. (2.4) shows that the smaller the quantization step, the higher is the maximum achievable stiffness. If we increase the physical viscous damping and Coulomb friction, the maximum achievable virtual stiffness will be accordingly increased. But practically, it is not something we want to do. As the user holds the end effector of a haptic interface, a high physical damping and Coulomb friction will impede the user's motion.

2.3.2 Rate-hardness

In [45], Lawrence *et al.* divide haptic contacts with a virtual surface into two stages: a contact transition period (about first 10 ms after the initial contact) and a stable contact period (forces and penetrations are close to constant values). Fig. 2.6 plots the recorded interaction force and penetration profiles when subjects tapped on four different virtual walls. Normally, we would think that the perceptual hardness of a virtual surface is based on the quasistatic relationship between the penetration and the force. For example, Surface 4 (dotted line) is reported to be softer than Surface 1 (solid line) because the penetration of former is almost twice larger than the latter surface in the stable contact period. But how could subjects still discriminate among stiffer virtual surfaces (100-2000 N/m) where the differences in the penetration are at the very lower end of human kinesthetic position change detection of 0.4-0.5 mm [49].

The authors find that the penetration change rates are similar for all four surfaces but the rates of force change differ considerably during the contact transition period. They hypothesize that the discrimination among those stiff surfaces is based on the relationship between the rate of force change and that of the penetration change during the contact transition period instead of the quasistatic relationship mentioned above. This fast force change directly generates some fast vibratory stimulation on the skin. The hypothesis is backed up by a psychophysical fact that human sensory channels (Pacinian corpuscles) on the hand are receptive to high frequency vibratory stimulation up to 1000 Hz, with a peak sensitivity around 300 Hz [49] [73]. For that, the authors propose a new performance metric for haptic interfaces called rate-hardness H_R which is defined as

$$H_R = \frac{\text{initial force rate of change}(N/s)}{\text{initial penetration velocity}(m/s)}. \tag{2.5}$$

Their experimental results support that the perceptual hardness of a virtual surface is more closely correlated with the surface rate-hardness H_R than with the surface virtual

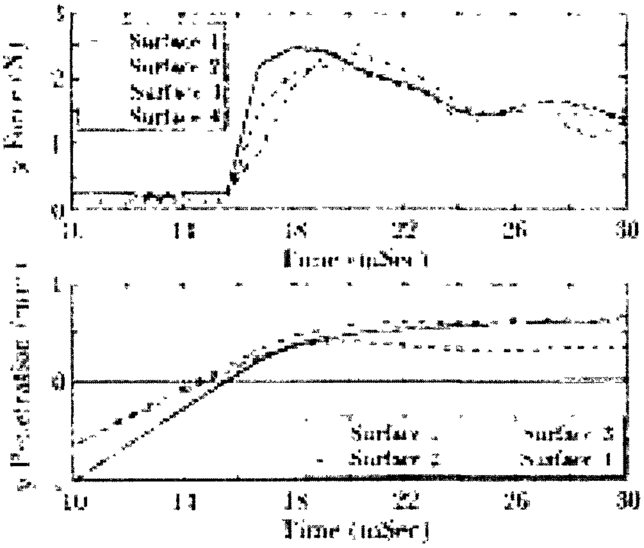


Figure 2.6: Plot of force and penetration profiles during haptically tapping on four different virtual walls [45].

stiffness K during the repeated taps on the virtual surface. This implies that the rate-hardness can possibly substitute for the large stiffness required to simulate a hard wall in virtual environments.

2.3.3 Reality-based modeling of vibrations

In [36], Okamura *et al.* present a method called reality-based modeling of vibrations to enhance the haptic tapping of virtual objects. Intuitively, if we indirectly tap a hard surface using a solid tool, we will feel some vibrations during the initial contact period. The authors model the vibration signal as a decaying sinusoidal waveform

$$Q(t) = A(v)e^{-\lambda t}\sin(\omega t), \tag{2.6}$$

where the amplitude $A(v)$ is a function of impact velocity, λ and ω denote the decay and the frequency of the sinusoid. In the paper, the vibration signal actually means the acceleration signal during the initial contact period. The parameters of the model are experimentally determined by measuring the impact velocity and acceleration signals. Specifically, the amplitude $A(v)$ is determined by the relationship between the measured acceleration and impact velocity, which is approximately linear. A least square fit is applied to the positive and negative peaks of the acceleration signal to get the decay

λ. The frequency ω is computed with fast Fourier transform (FFT) on the acceleration signal. With the proposed method, the actual feedback force for the haptic rendering of a virtual object is the classic spring force superimposed by the vibration force. To improve the model, the parameters can be further adjusted with subjective perceptual experiments. The final experimental results support that the superimposed vibration enhance the realism of the haptic tapping. Similarly, Salcudean and Vlaar overlay a short duration rectangular pulse on the classic spring force to increase the perceived hardness of a virtual wall [74].

2.3.4 Event-based haptics

In [37], Kuchenbecker *et al.* extend the idea of rate-hardness and reality-based modeling of vibration and present a paradigm called event-based haptics to improve the realism of haptic tapping by superimposing some pre-computed impact transients to the classic spring force during the initial contact period. Different from Okamura's method, the authors also consider the user's hand dynamics and the exerting grip force on the stylus. Assuming second-order hand dynamics, they demonstrate an approximate linear relationship between the hand parameters and the measured grip force. With the experimental data with real objects, they also show that the interaction force during the tapping scales linearly with the incoming velocity, the incoming acceleration, and the user's grip force on the end effector. With these relationships, it is possible to create a look-up table so an appropriated additive transient force signal can be retrieved in real-time. The experimental results with a 1-DOF haptic rendering are very encouraging. However, the process of estimating parameters for these pre-computed transient signals is very tedious. It involves the system identification of the haptic interface and the user's hand dynamics, which are not trivial tasks. Furthermore, to extend the method to a multi-DOF situation will be more challenging. To avoid such a tedious process, Han *et al.* propose a simpler way called "stiffness shifting" to improve the perceived hardness in [75]. The idea is to present a softer stiffness during the initial contact period and then switch to a higher stiffness. The switch timing is critical and is experimentally determined to be 20 ms. The authors claim that a virtual wall rendered with the stiffness shifting is perceived as hard as one rendered using the classic spring model with a 2.5 times higher stiffness.

2.3.5 Impulsive forces modeling

Unlike the above two methods where only one collision point is considered, Constantinescu *et al.* propose a hybrid method using both impulsive and penalty forces for the haptic rendering of rigid contacts [46] [76] [77]. Since the virtual tool can be modeled as a rigid object or a manipulator linkage, multiple contacts are possible. At each simulation step, the collision detection algorithm reports one of three possible states between the virtual tool and its neighboring virtual objects: free motion, colliding contact, and resting contact. The virtual tool is in the free motion if it has no contact with the neighboring objects. It is in the colliding contact if there is at least one new contact with negative contact velocity within its contact group. This state can be considered as the contact transition period aforementioned. The virtual tool is in the resting contact if it is in neither the free motion nor the colliding state. During the resting contact, the feedback force to the user is the summation of the contact force at each collision point computed with the conventional penalty force (spring force) plus a damping term. These collision forces are also applied onto the proxy of the virtual tool, whose motion is then determined using Newton's second law in its configuration space. When the virtual tool is in the colliding contact state, the feedback force to the user is computed differently. Instead of computing the penalty force, the authors first apply Newton's law of restitution at each collision. Newton's law of restitution states that if two particles collide, the relationship between the velocity before and after the collision is given by

$$v_0 = -ev_i, \tag{2.7}$$

where $e \in [0, 1]$ denotes the restitution coefficient. By adjusting the restitution coefficient, we may get a collision from perfect plastic ($e = 0$) to perfect elastic ($e = 1$, infinite stiffness). With known pre- and post-collision velocity at each collision, the contact impulses $p = [p_1 \cdots p_i \cdots p_c]$ (c is the total number of contacts) applied onto the proxy can then be solved together. The final impulsive force applied to the user's hand is calculated as

$$F = \frac{J_c^T p}{\Delta t}, \tag{2.8}$$

where J_c is the combined Jacobian matrix computed at each contact, and Δt is the time step of the simulation. The resulting impulse forces generate a large acceleration on the hand so the user gets a rigid contact with the virtual objects. The post-collision kinetic energy of the system is proved to be smaller than or equal to the pre-collision one, so the passivity of the proposed collision resolution is guaranteed. The work has been extended in [78] by explicitly setting the time duration for the colliding state to be less than 100 ms, so the user perceives a sequence of impulsive forces superimposed on the penalty force instead of only one instantaneous impulsive force upon contact with the neighboring virtual object. This is in fact equivalent to generate high frequency vibratory signals upon contact discussed above so as to increase the perceived rigidity of virtual objects.

2.3.6 Visual modality substitution

On the other hand, Srinivasan *et al.* demonstrate that the visual information of the penetration between the tool and the interacting virtual object can be controlled to influence a user's perception of stiffness [79]. The authors conducted a series of psychophysical experiments on the stiffness discrimination of two virtual springs (equivalent to haptic rendering of two virtual walls with different stiffness coefficients). During the experiments, the subjects not only feel the movements and forces through their hands, but also see the deformation of the springs graphically on a computer monitor. However, the relationship between the visually displayed deformation and the actual deformation of each spring is systematically controlled across experimental trials, from fully registered (the visual deformation equals to the actual deformation of each spring) to completely interchanged (the visual deformation for one spring equals to the actual deformation of the other spring). The subjects are blind to the relationship. The results demonstrate that the subjects' stiffness perception of a virtual spring mainly rely on their judgment on the relationship between the graphically displayed deformation and the interaction force sensed tactually, instead of the relationship between the real hand position information sensed kinesthetically and the interaction force. These results indicate such haptic illusions can be exploited to improve the limited achievable virtual stiffness of haptic interfaces. In [11], Knörlein *et al.* study the influences of inevitable visual or haptic delays in virtual environments on the perception of stiffness. They experimentally show that visual delays cause an increase in perceived stiffness while delays for haptic feedback results in a decreased perceived stiffness. They suggest that the same delay applied in both sensory channels could partially compensate the effects.

2.4 Summary

In this chapter, we first introduced haptic related human factors with an emphasis on the kinesthetic perception. Following that, we introduced haptic interfaces and their physical elements. In the second part of the chapter, we explained the concept of maximum achievable virtual stiffness for a haptic interface. With the passivity theory, the theoretical analyses have shown that the physical damping and Coulombic friction of the device have positive effects on the maximum achievable virtual stiffness. However, the passivity theory is a conservative concept itself, *i.e.*, a passive system is guaranteed to be stable but not vice versa. One less conservation bery very interesting way to disspate excessive energy is proposed by Gosline and Hayward in [80], in which an Eddy current brake is utilized as a programmable viscous damper for haptic interfaces. Practically, the easiest way to find out the maximum achievable virtual stiffness for a specific device is to test. Most of the time, the practical maximum achievable stiffness will be higher that the theoretical one because the user can absorb some excessive energy generated by the simulation.

We also reviewed current studies on the improvement of the maximum achievable virtual stiffness in a haptic simulation. The popular approach is to superimpose a short duration transient force signal on the classic penalty force during the initial contact period. However, the ways to compute the transient signal are varied.

Chapter 3

Velocity and Acceleration Estimation for Haptic Interfaces

3.1 Introduction

As most haptic interfaces are not equipped with tachometers or accelerometers, the velocity and acceleration information about the movements has to be derived from the position measurements. In this chapter, we first discuss the position resolution of haptic interfaces in Sec. 3.2. With a high sampling rate, a small measurement error in the position may result in a large error in the estimated velocity and acceleration. Accurate position measurements are also important for haptic rendering of rigid objects. The difference between being just above a virtual surface and being just below may mean presenting a zero or a high force to the user. Consider human can distinguish forces as small as 0.1-0.2 N [51]. If we use a haptic interface with a position resolution of 0.1 mm for the rendering of a virtual wall with a stiffness of 1.5 N/mm, the output force would have a component of 0.15 N corresponding to the measurement error. After that, we give a thorough literature review on the velocity and acceleration estimation from discrete position measurements in Sec. 3.3, followed by the discussion on two characteristics of human arm/hand movements in Sec. 3.4. Sec. 3.5 describes our proposed jerk constrained 4-state Kalman filter. The simulation and experimental results are presented in Sec. 3.6.

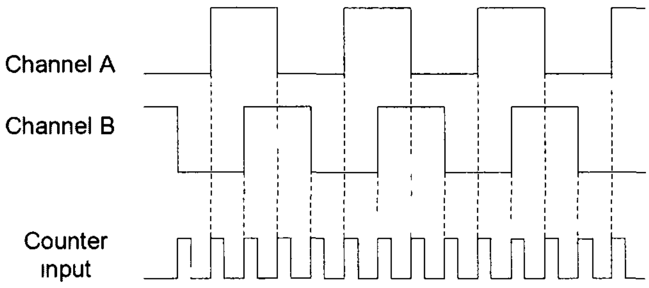


Figure 3.1: A sample output signal from an optical encoder for a clockwise rotation.

3.2 Position Resolution of Haptic Interfaces

3.2.1 Incremental optical shaft encoder

The movements of a haptic interface are often sensed by incremental optical encoders attached to its motor shafts. An incremental optical encoder usually has three basic components, including an encoder-line engraved disk, a light source, and a dual light detector unit. The encoder disk is mounted on the motor's shaft, while the light source and the detectors are mounted on a fixed frame. The two detectors are placed 1/4 of a line apart so their outputs are in quadrature phase (90 degrees out of phase). As the disk rotates with the motor's shaft, the detectors generate two quadrature signals which tell us the movement direction and the angle of the joint. Most of incremental encoders in use for haptic interfaces are digital encoders in which two detector outputs are first converted into two digital quadrature signals (channel A and channel B) as shown in Fig. 3.1 with a threshold circuit and then decoded with a decoder logic circuit. If all the transition edges of the two signals are counted, a resolution of 1/4 of line space on the disk can be achieved. For instance, if an encoder disk has 512 lines per revolution (equivalent to pulses per revolution), the resolution will be

$$\Delta = \frac{2\pi(\frac{\text{radius}}{\text{revolution}})}{512(\frac{\text{cycle}}{\text{revolution}}) \times 4(\frac{\text{transition}}{\text{cycle}})} \approx 0.003068 \quad (\frac{\text{radius}}{\text{transition}}). \tag{3.1}$$

These transition edges increment or decrement a digital counter which maintains an absolute number c_0 corresponding to a predefined zero position. Following that, if the signal is sampled every T seconds, the joint angle at time kT is

$$\theta(kT) = \Delta(c(kT) - c_0), \tag{3.2}$$

where c is the reading of the digital counter and $k \in \mathbb{N}$. It is worthwhile to note that the measurement error of an incremental shaft encoder is related to the direction of the movement. If the output of channel A is ahead of the output of channel B for a counter-clockwise movement and the counter is increasing at the same time, the measurement error has a uniform distribution from 0 to Δ. On the other hand, for a clockwise movement, if the counter is decreasing, the measurement error is also a uniform distribution but from $-\Delta$ to 0. Therefore, if the direction of the movement is not known, the measurement error should be assumed to have a uniform distribution from $-\Delta$ to Δ.

3.2.2 Theoretical modeling of position measurement resolution

To be more general, let us consider a multi-joint haptic interface. If the joint angle signal $\theta(t)$ is sampled at every T seconds, the measured joint angle at time kT is then

$$\theta_m(k) = \theta(k) + e(k). \tag{3.3}$$

where $\theta(k)$ and $e(k)$ represent the true joint angle and the measurement error at time kT respectively. As discussed, for a shaft encoder without a known movement direction, $e(k)$ can be modeled as a uniform distributed random variable, such that $-\Delta \leq e(k) \leq \Delta$ with a variance of

$$\sigma_\theta^2 = \text{var}(e(k)) = \frac{\Delta^2}{3}. \tag{3.4}$$

With the forward kinematics of a specific device, the relative translational position of a tool frame defined on the end effector to a spatial home frame defined in the workspace can be formulated as

$$\begin{aligned} x(t) &= f_1(\theta_1(t), \theta_2(t), \cdots, \theta_i(t), \cdots, \theta_n(t)), \\ y(t) &= f_2(\theta_1(t), \theta_2(t), \cdots, \theta_i(t), \cdots, \theta_n(t)), \\ z(t) &= f_3(\theta_1(t), \theta_2(t), \cdots, \theta_i(t), \cdots, \theta_n(t)), \end{aligned} \tag{3.5}$$

where θ_i denotes the angle of the i_{th} joint, and n is the number of the joints.

Let us take a SensAble PHANToM OMNI haptic device as a specific example in the following discussion. As illustrated in Fig. 3.2, the arm structure of an OMNI device is a four-bar linkage which is controlled by three motorized joints. The translational part of the forward kinematics for the OMNI device is given by

$$\begin{aligned} x_m &= \sin\theta_{m1}(L_1 \cos\theta_{m2} + L_2 \cos\theta_{m3}) \\ y_m &= L_2 - L_2 \cos\theta_{m3} + L_1 \sin\theta_{m2} \\ z_m &= -L_1 + \cos\theta_{m1}(L_1 \cos\theta_{m2} + L_2 \cos\theta_{m3}), \end{aligned} \tag{3.6}$$

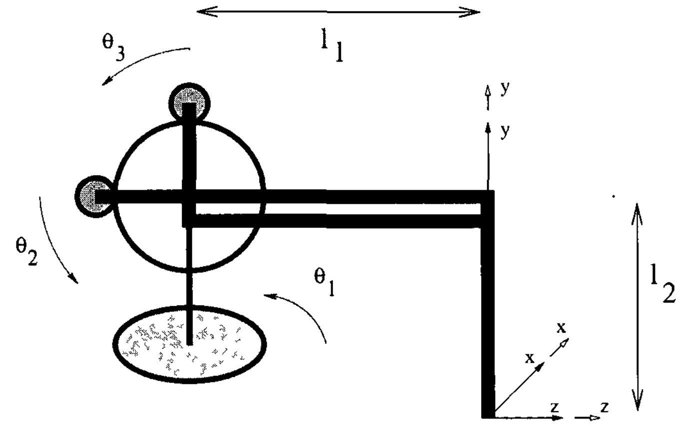


Figure 3.2: Zero configuration of a SensAble OMNI device. (adapted from [81])

where L_1 and L_2 are the length of the two rigid links [87]. The time variable t is dropped from Eq.(3.6) just for clarity. If the measurement errors of three joints are modeled by three independent random variables with a uniform distribution, $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$, it yields

$$\begin{aligned} x_m &= \sin(\theta_1 + q_1)(L_1 \cos(\theta_2 + q_2) + L_2 \cos(\theta_3 + q_3)) \\ y_m &= L_2 - L_2 \cos(\theta_3 + q_3) + L_1 \sin(\theta_2 + q_2) \\ z_m &= -L_1 + \cos(\theta_1 + q_1)(L_1 \cos(\theta_2 + q_2) + L_2 \cos(\theta_3 + q_3)). \end{aligned} \tag{3.7}$$

After considerable algebraic manipulations, the variances of $\mathbf{x}_m$, $\mathbf{y}_m$, and $\mathbf{z}_m$ are

$$\begin{aligned} \mathrm{var}(\mathbf{x}_m) &= \mathcal{E}\{\mathbf{x}_m^2\} - (\mathcal{E}\{\mathbf{x}_m\})^2 \\ \mathrm{var}(\mathbf{y}_m) &= \mathcal{E}\{\mathbf{y}_m^2\} - (\mathcal{E}\{\mathbf{y}_m\})^2 \\ \mathrm{var}(\mathbf{z}_m) &= \mathcal{E}\{\mathbf{z}_m^2\} - (\mathcal{E}\{\mathbf{z}_m\})^2 \end{aligned} \tag{3.8}$$

where

$$\begin{aligned} \mathcal{E}\{\mathbf{x}_m\} &= L_1 a_2 b_1 + L_2 b_1 b_3 \\ \mathcal{E}\{\mathbf{x}_m^2\} &= L_1^2 c_2 d_1 + L_2^2 d_1 d_3 + 2L_1 L_2 d_1 a_2 b_3 \\ \mathcal{E}\{\mathbf{y}_m\} &= L_2 - L_2 a_3 + L_1 b_2 \\ \mathcal{E}\{\mathbf{y}_m^2\} &= L_2^2 - 2L_2^2 a_3 + L_2^2 c_3 + 2L_1 L_2 b_2 - 2L_1 L_2 a_3 b_2 + L_1^2 d_2 \\ \mathcal{E}\{\mathbf{z}_m\} &= -L_1 + L_1 a_1 a_2 + L_2 a_1 b_3 \\ \mathcal{E}\{\mathbf{z}_m^2\} &= L_1^2 - 2L_1^2 a_1 a_2 + L_1^2 c_1 c_2 + 2L_1 L_2 c_1 a_2 b_3 - 2L_1 L_2 a_1 b_3 + L_2^2 c_1 d_3, \end{aligned} \tag{3.9}$$

and

$$\begin{aligned}
a_i &= \mathcal{E}\{\cos(\theta_i + q_i)\} = \frac{1}{\Delta}\cos(\theta_i)\sin(\Delta) \\
b_i &= \mathcal{E}\{\sin(\theta_i + q_i)\} = \frac{1}{\Delta}\sin(\theta_i)\sin(\Delta) \\
c_i &= \mathcal{E}\{\cos^2(\theta_i + q_i)\} = \frac{1}{2} + \frac{1}{4\Delta}\cos(2\theta_i)\sin(2\Delta) \\
d_i &= \mathcal{E}\{\sin^2(\theta_i + q_i)\} = \frac{1}{2} - \frac{1}{4\Delta}\cos(2\theta_i)\sin(2\Delta).
\end{aligned}$$

To calculate the covariances among $\mathbf{x}_m$, $\mathbf{y}_m$ and $\mathbf{z}_m$ with Eq. (3.8) directly, it will give us very lengthy expressions. Since Δ is very small for high precision shaft encoders, Eq. (3.7) can be linearized with the following approximations:

$$\begin{aligned}
\sin q_i &\simeq q_i, \\
\cos q_i &\simeq 1, \\
q_i q_j &\simeq 0.
\end{aligned}$$

for $i = 1, 2, 3 \wedge i \neq j$. We then form these three random variables $\mathbf{x}_m$, $\mathbf{y}_m$, and $\mathbf{z}_m$ as affine transformations of $\mathbf{q}_1$,$\mathbf{q}_2$, and $\mathbf{q}_3$ at different joint angles. This relationship may be expressed in matrix form as

$$\begin{bmatrix} \mathbf{x}_m \\ \mathbf{y}_m \\ \mathbf{z}_m \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \tag{3.10}$$

where

$$\mathbf{A} = \begin{bmatrix} L_1 \cos\theta_1 \cos\theta_2 + L_2 \cos\theta_1 \sin\theta_3 & -L_1 \sin\theta_1 \sin\theta_2 & L_2 \sin\theta_1 \cos\theta_3 \\ 0 & L_1 \cos\theta_2 & L_2 \sin\theta_3 \\ -L_1 \sin\theta_1 \cos\theta_2 - L_2 \sin\theta_1 \sin\theta_3 & -L_1 \cos\theta_1 \sin\theta_2 & L_2 \cos\theta_1 \cos\theta_3 \end{bmatrix} \tag{3.11}$$

and

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} \sin\theta_1 (L_1 \cos\theta_1 + L_2 \cos\theta_3) \\ L_2 - L_2 \cos\theta_3 + L_1 \sin\theta_2 \\ -L_1 + \cos\theta_1 (L_1 \cos\theta_1 + L_2 \cos\theta_3) \end{bmatrix}. \tag{3.12}$$

Using the linearity between two sets of random variables, we have

$$\mathbf{\Lambda}_{\mathbf{x}_m \mathbf{y}_m \mathbf{z}_m} = \mathbf{A} \mathbf{\Lambda}_{\mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3} \mathbf{A}^T, \tag{3.13}$$

Table 3.1: Joint measurement resolution of SensAble OMNI haptic device.

Joint	Resolution (rad./tran.)
θ_1	4.1120e-4
θ_2	4.0806e-4
θ_3	4.0749e-4

where $\mathbf{\Lambda}_{\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3}$ is the covariance matrix of $\mathbf{q}_1$,$\mathbf{q}_2$, and $\mathbf{q}_3$ given by

$$\mathbf{\Lambda}_{\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3} = \begin{bmatrix} \sigma_{q1}^2 & 0 & 0 \\ 0 & \sigma_{q2}^2 & 0 \\ 0 & 0 & \sigma_{q3}^2 \end{bmatrix} = \begin{bmatrix} \Delta_1^2/3 & 0 & 0 \\ 0 & \Delta_2^2/3 & 0 \\ 0 & 0 & \Delta_3^2/3 \end{bmatrix} \tag{3.14}$$

The workspace of an OMNI haptic device is symmetric about $\theta_1 = 0$, so are the covariances of the measurement errors. The device uses optical shaft encoders with a total of 512 pulses per revolution. As given by Eq. (3.1), the encoders have a resolution of 0.00307 radius per transition. However, to find out the resolution for each joint, we still need to know their respective gear ratios which are difficult to be measured. For that reason, we simply divide the joint angle of each joint and the corresponding encoder value retrieved from the device's API. The results are listed in Table 3.1.

To estimate the covariance among $\mathbf{x}_m$, $\mathbf{y}_m$, and $\mathbf{z}_m$, we can uniformly sample the device's workspace. Using Eq. (3.8) and (3.9), the results show that $\text{cov}(\mathbf{x}_m, \mathbf{y}_m)$, $\text{cov}(\mathbf{x}_m, \mathbf{z}_m)$, and $\text{cov}(\mathbf{y}_m, \mathbf{z}_m)$ are all close to zero, which indicate that the measurement errors along x, y, and z directions of the device are uncorrelated. Eq. (3.14) also gives the approximated $\text{var}(\mathbf{x}_m)$, $\text{var}(\mathbf{y}_m)$, and $\text{var}(\mathbf{z}_m)$ which are actually close to the precise results given by Eq. (3.8). Anyhow, we should keep in mind that $\text{var}(\mathbf{x}_m)$, $\text{var}(\mathbf{y}_m)$, and $\text{var}(\mathbf{z}_m)$ are functions of on the current joint angles, so they vary across the workspace. These statistics define the properties of the measurement errors when the device is at different configurations; namely, $e_x(kT)$, $e_y(kT)$, and $e_z(kT)$. At the center of the workspace, the standard deviations along x, y, and z directions are listed in Table 3.2. The haptic device has the least standard deviation of 0.0314 mm along y direction because it only relates to Joint 2 and Joint 3.

Table 3.2: Standard deviation of position measurement error.

Axis	Standard Deviation (mm)
x	0.0376
y	0.0314
z	0.0341

3.3 Related Work

3.3.1 Friedland's criterion

In [82], Friedland uses a Kalman filtering technique to analyze the achievable position and velocity accuracy in a system which measures position at a uniform sampling interval of T. The signal dynamics are modeled with the assumption of a zero mean, uncorrelated piecewise-constant acceleration. Let p_1, p_2, $\hat{p}_1$ and $\hat{p}_2$ denote the mean-square error of position and velocity estimate before and after a measurement, Δ be the inherent sensor resolution, and σ_a be the standard deviation of the acceleration respectively. The analytical solutions of the error covariance are given by

$$\frac{p_1}{\Delta^2} = \frac{\sqrt{1+2r}}{r^2}(\sqrt{1+2r}+1)^2 \tag{3.15}$$

$$\frac{p_2}{\sigma_a^2 T^2} = \frac{1}{2}(\sqrt{1+2r}+1) \tag{3.16}$$

$$\frac{\hat{p}_1}{\Delta^2} = \frac{\sqrt{1+2r}}{r^2}(\sqrt{1+2r}-1)^2 \tag{3.17}$$

$$\frac{\hat{p}_2}{\sigma_a^2 T^2} = \frac{1}{2}(\sqrt{1+2r}-1), \tag{3.18}$$

where

$$r = \frac{4\Delta}{\sigma_a T^2}. \tag{3.19}$$

The ratio r can be taken as a measure of noise-to-signal ratio. To get an insight of the relationship between the sampling rate and the achievable position error, the ratios p_1/Δ^2 and $\hat{p}_1/\Delta^2$ are plotted in Fig. 3.3. Eq. (3.19) shows that it is possible to estimate the position with a greater accuracy than the inherent sensor resolution if the ratio $r \geq 16.66$. Friedland formulates this design guideline as

$$T < 0.49\sqrt{\frac{\Delta}{\sigma_a}}, \tag{3.20}$$

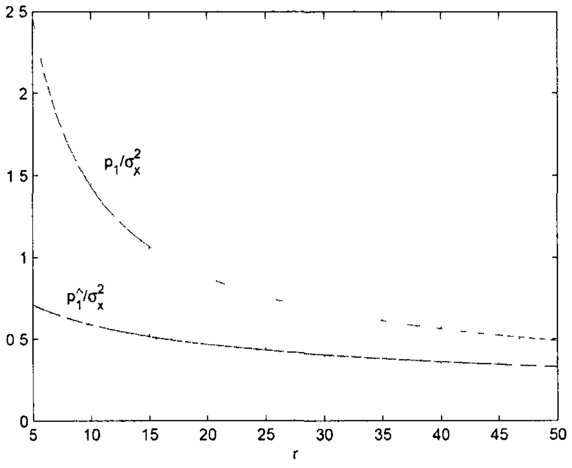


Figure 3.3: Position error ratio before and after measurements.

which states that the maximum position error is bounded by the inherent sensor error by use of a sufficiently high sampling rate. But for the velocity estimation, it is always greater than the unity. The higher the product of $\sigma_a T$, the worse is the estimated velocity

3.3.2 Singer's optimal Kalman filter model

Back in 1970, Singer derived a well-known optimal Kalman filter to estimate states of maneuvering targets, such as aircrafts, ships, and submarines, where data samples are received from radars [83]. The dynamic equation of target motion is constructed with the assumption that the target normally moves at a constant velocity and all the maneuvering actions are viewed as perturbations. The maneuvering is quantified by two parameters: the variance of the acceleration σ_m^2 and the duration of the maneuvering T_m. Singer exploits an important fact that if a target is accelerating at time t, it is likely to be accelerating also at time $t+\tau$ for a sufficiently small τ. With the standard discretization procedure, the discrete time target equation of motion has a form

$$x(k+1) = \Phi(T,\alpha)x(k) + u(k), \tag{3.21}$$

where T is the sampling interval, α is the reciprocal of T_m, and $\Phi(T,\alpha)$ is the state transition matrix

$$\Phi(T,\alpha) = \begin{bmatrix} 1 & T & \frac{1}{\alpha^2}(-1+\alpha T+e^{-\alpha T}) \\ 0 & 1 & \frac{1}{\alpha}(1-e^{-\alpha T}) \\ 0 & 0 & e^{-\alpha T} \end{bmatrix}. \tag{3.22}$$

When αT is small (≈ 0), $\Phi(T,\alpha)$ and the process covariance matrix $Q(k) = E\{u(k)u(k)^T\}$ reduce to

$$\Phi = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \text{and } Q(k) = q \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix}, \tag{3.23}$$

where $q = 2\alpha\sigma_m^2$. Back to our haptic interfaces, they are actually a kind of man maneuvering targets. For haptic simulations where T is usually around 1 ms, these reduced matrices are computationally attractive but should be used with caution, *i.e.*, a small T does not imply small αT. For skilful human hand movements, α is usually large.

3.3.3 Bélanger multiple integrator model

In [84] and [85], Bélanger *et al.* study the asymptotic behavior of the Kalman filtering for the robotic joint velocity and acceleration estimation as T tends to zero. The joint angle signals are assumed to be generated by passing a white noise through a linear and all-pole filter. As T goes to zero, this all-pole filter can be simplified to a multi-integrator system. The resulting system of equations in state space is described as

$$\begin{aligned} \dot{x}(t) &= Fx(t) + G\omega(t) \\ y(k) &= Hx(k) + e(k) \end{aligned} \tag{3.24}$$

where

$$x(t) = \begin{bmatrix} \text{position} & \text{velocity} & \text{acceleration} & \cdots \end{bmatrix}^T$$

$$F = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
G &= \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^T \\
H &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \\
w(t) &= \text{white Gaussian process noise with covariance } q \\
e(k) &= \text{measurement noise with covariance } \sigma_R^2 \\
Q(k) &= E[\omega(k)\omega^T(k)] \\
Q_{ij}(k) &= \frac{q}{(n-i)!(n-j)!(2n-i-j+1)} T^{2n-i-j+1}
\end{aligned}$$

If both velocity and acceleration estimates are needed, at least a triple-integrator model is needed. Considering a triple-integrator model and assuming zero-order hold for the input, we will get the following discret model [86]:

$$\Phi = \begin{bmatrix} 0 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } Q(k) = q \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}. \tag{3.25}$$

Surprisingly, they share the same mathematical form as the simplified Singer's model discussed above. However, the conditions to get these reduced forms are different: $T \to 0$ for the Bélanger triple integrator, and $\alpha T \to 0$ for the Singer's model respectively. The process noise variance q for the Singer's model also has an explicit physical meaning which can be determined intelligently instead of the trial-and-error method for the Bélanger's model. Nevertheless, we may conclude that the Bélanger triple-integrator model takes the special form of the Singer's model.

To compare the performance of these methods, we simulate a single robot joint under PD control modeled by

$$\frac{y}{y_d} = \frac{6s + 100}{s^2 + 6s + 100} \tag{3.26}$$

The same example is used by Bélanger in [84]. The desired acceleration trajectory $a_d(t)$ and the theoretical output signals are reproduced in Fig. 3.4. It is assumed that the shaft encoder has a $\theta_m = 0.003°$ resolution and the sampling frequency is 100 Hz. Two acceleration amplitudes, $A = 10$ and 1 $°/s^2$, are tested, corresponding to a slow and fast trajectory respectively.

The simulation results are consistent with the criterion derived by Friedland in [82]. However, for a shaft encoder with a high inherent resolution, once the sampling frequency satisfies the Friedland's criterion, the further increase of the sampling frequency

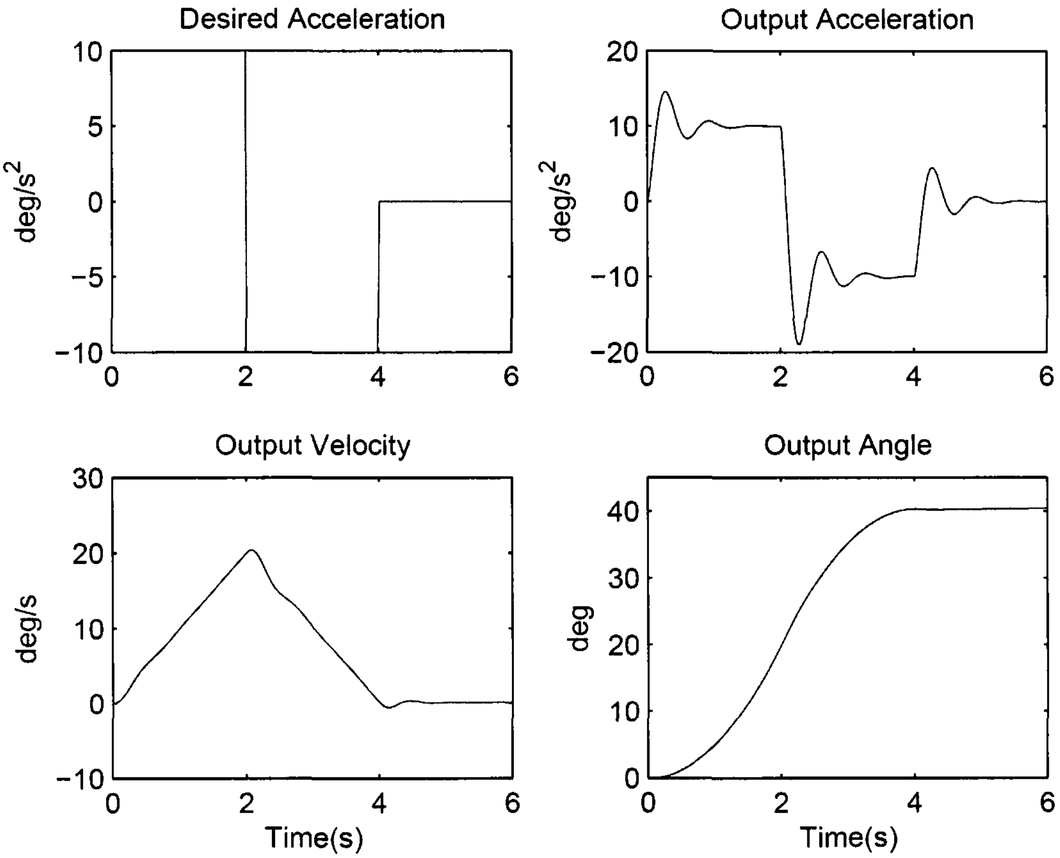


Figure 3.4: Angle, velocity and acceleration signals, $A = 10$

will not further improve the accuracy. Compared with the first-order finite difference method, the triple-integrator model is 2-4 times better in terms of the standard deviation of the velocity estimation, and there is an order of magnitude improvement in the acceleration estimation. For a 10 *ms* sampling interval, the first-order finite difference method does give an acceptable velocity estimation, but the acceleration estimation has an unacceptable standard deviation of 21.6 $^\circ/s^2$. Compared with the double-integrator, the triple-integrator model gives slightly better results. However, this does not imply that the higher the filter's order, the better the estimation results. It really depends on the dynamics of the motion itself. Table 3.3 summarizes the statistics of the estimation errors with these different methods.

Table 3.3: Statistics of estimation errors with different estimation methods.

method	q	angle (°)		vel. (°/s)		accel. (°/s^2)	
		mean	std	mean	std	mean	std
Finite Diff.	-	1.47e-3	8.76e-4	3.50e-4	1.24e-1	1.68e-2	2.16e+1
Triple Integ.	1000	-1.46e-3	6.89e-4	-2.40e-5	5.94e-2	2.20e-3	1.35
	200	-1.46e-3	6.95e-4	1.71e-4	5.60e-2	7.06e-3	1.19
	20	-1.44e-3	1.12e-2	9.15e-4	7.38e-2	1.82e-2	1.75
Double Integ.	1000	-1.46e-3	8.72e-4	-3.43e-4	1.78e-1	-	-
	200	-1.46e-3	8.60e-4	-3.50e-4	1.56e-1	-	-
	20	-1.46e-3	8.10e-4	-3.33e-4	9.33e-2	-	-
	2	-1.46e-3	9.07e-4	-3.42e-4	7.21e-2	-	-
Singer Model	110	-1.46e-3	7.37e-4	1.52e-4	5.41e-2	3.80e-3	1.29

To understand the effects of the process noise covariance q in the triple-integrator model, we sweep q from 20 to 1000 and the results are presented in Fig. 3.5. Clearly, there is an optimal selection range for q. For example, the standard deviation of the angle error is 0.122° at $q = 20$, which is almost 4 times larger than the inherent resolution of 0.003°. There is also a trade-off between the mean error and the standard deviation. A smaller q will give smoother estimation results but a larger mean error. It implies that q should be adaptive. The same system is also simulated with a double integrator model. The results show that the double integrator model is less sensitive to the choice of q.

To get accurate estimation at both low and high velocity, Bélanger *et al.* also proposed a method which switches between a time-varying Kalman filter for faster motions and a constant-time Kalman filtering at slower motions [91]. For fast motions, the pulse from the encoder triggers the filtering process using the time-varying Kalman filter so as to have more updates. On the other hand, if the time since the last pulse is greater than a preset time and still no new pulse comes in, the estimator will actively read the encoder value and update with the constant-time Kalman filter. Their experimental setup is a DC motor under the velocity control through a PD controller and the results show the improvements on the acceleration estimation at the high velocity. But for a multi-joint

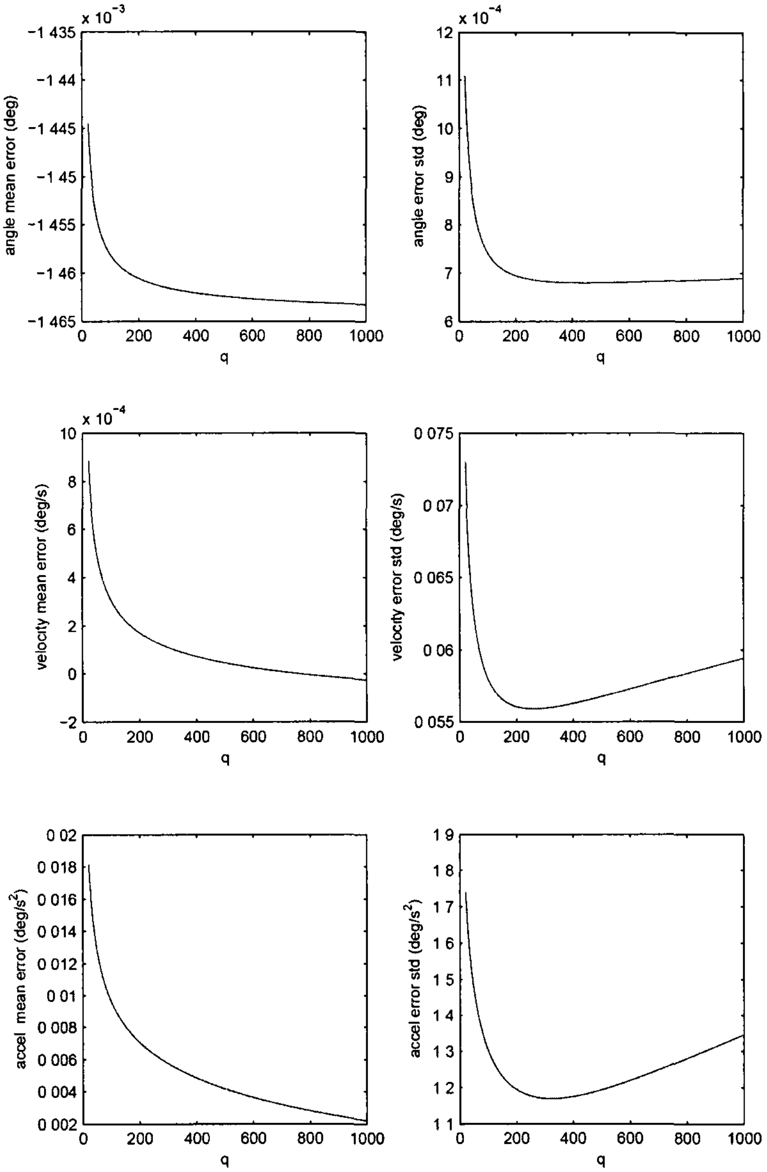
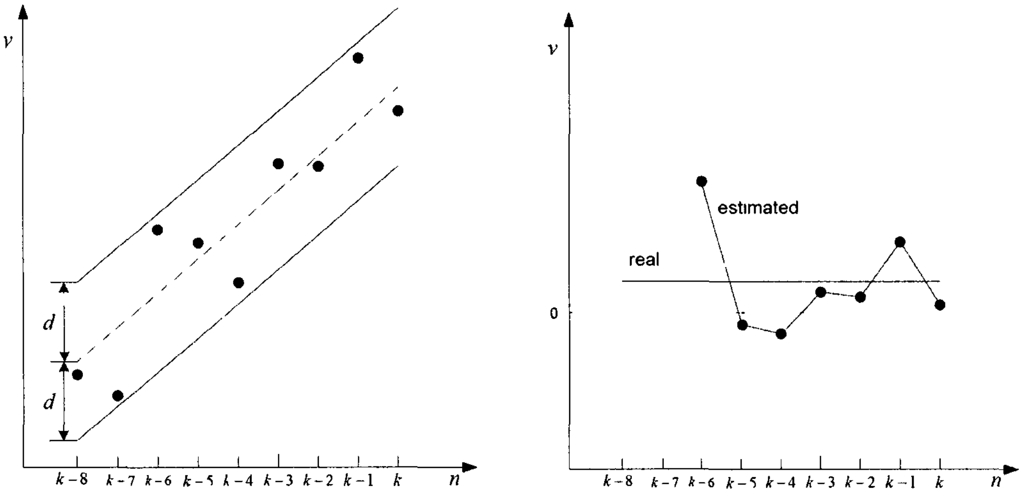


Figure 3.5: Angle, velocity and acceleration estimation mean errors and their standard deviations with a triple integrator Kalman filter as the variance of the process noise q changes from 20 to 1000.

haptic interface, the implementation of the approach is more demanding. First, the low level access of the encoder signal is needed. Second, in the case that one joint has a low velocity while another joint may have a high velocity, synchronization of the estimates from different joints becomes an issue.



(a) position (b) velocity

Figure 3.6: Adaptive window method for velocity estimation. The dots represent the data samples at different sampling instant

3.3.4 Janabi-Sharifi's adaptive windowing method

In [87], Janabi-Sharifi *et al.* proposes a first-order adaptive windowing method which minimizes the error variance of velocity estimates for haptic interfaces. They assume that the position trajectory of the device's end effector has a piecewise continuous and bounded derivative and a uniformly distributed ($\| e_k \|_\infty = d$) measurement noise. The method is to find a maximum window size $M \in \{2, 3, \cdots\}$ such that

$$|y_{k-i} - \hat{y}_{k-i}| \le d, \qquad \forall i \in \{0, 1, 2, \cdots, M-1\}, \tag{3.27}$$

where y_{k-i} is the position measurement at $(k-i)T$, $\hat{y}_{k-i} = y_k - iT\hat{v}_k$, and $\hat{v}_k = \frac{y_k - y_{k-M+1}}{(M-1)T}$. Since only the start and the end measurements in the window are used, the velocity profile will have overshoots if M is too small. To provide additional smoothing, all the samples in the window are used in the velocity estimation with the least-square method. In this case, the estimated velocity at kT can be written as

$$\hat{v}_k = \frac{12\sum_{i=0}^{M-1}(\frac{M-1}{2} - i)y_{k-i}}{TM(M-1)(M+1)}. \tag{3.28}$$

The detailed derivation of Eq. (3.28) is given in Appendix A.

In Fig. 3.6a, a position trajectory which has a constant velocity during time interval $((k-8)T, kT)$ is plotted. With the adaptive window method, the window size M for each sampling point starting from $k-6$ to k will be $\{2,2,3,3,4,2,4\}$. As shown in Fig. 3.6b, the resulting velocity is quite spiky since the least-square fitting has no much use for too small window sizes. To get an insight of how often the window size being two happens, we simulate a constant position trajectory with 10000 sample points corrupted with a uniform distributed noise using Matlab. With a total of 100 runs, the average occurrences of window size being two are 2498. It means that about of the velocity estimation uses first-order finite difference method which is doomed to fail for high sampling rate. For non-constant position trajectory such as human hand movements, will surely happen quite often. Therefore, a minimum window size has to be determined to compress the noise. With this minimum window size, the least-square fitting does smooth the resulted velocity estimation. On the other hand, if the window size becomes too large, it induces too much time delay and affects the stability margin of the system. Nevertheless, the idea behind the method is very important, which is to compress the measurement noise while capture the transient behavior. In [88], Liu proposes a similar method to find the window size based on the required relative velocity accuracy.

3.3.5 Newton predictor enhanced Kalman filtering

In [89], Han *et al.* estimate the angular acceleration of a manipulator joint by firstly passing the signal through a triple-integrator Kalman filter to get raw acceleration estimates and then feed these raw estimates to a Newton predictor to get the final smoothed estimates. The assumption to use a Newton predictor is that the acceleration signal can be expressed as a polynomial [90]. An acceleration feedback control for one joint of a 2-DOF mechatronic system is presented to compare the proposed Newton predictor enhanced Kalman filtering (NPEKF) method against the separated Kalman filtering (KF) method and Newton predictor (NP) method. The results show that the phase lag of the angular acceleration estimated by NPEKF is 1/8 that estimated by NP and 1/3 by KF for a 10 Hz sine torque input. Moreover, the NPEKF is able to suppress the noise within the frequency range of 2-10 Hz for the acceleration signal. But the Newton predictor has very large gains at the higher frequencies and consequently the noise gets amplified. Since the acceleration of a human arm/hand trajectory is in the range of 20-30 Hz, NP does not serve our objective.

In summary, the basic criterion of the velocity and acceleration estimation from discrete position samples is to compress the high frequency noise at the low speed while keep the transient behavior of the primary signal at the high speed. Therefore, the filtering parameters should be relied on the trajectory itself. On the other hand, sophisticated filters often have the model-filter mismatch problems that their performance degrades significantly when the actual motion gets away from the assumed model. Theoretical analysis shows that a second-order model with parameters varying with muscle activation and elbow angle was unable to reproduce the experimental observations of the human arm trajectory. The least complex competent characterization of human arm movement requires a fourth-order model [91]. To get a suitable model to describe the motion of the human manipulated haptic interfaces, some known human arm trajectory characteristics may be exploited.

3.4 Human Arm/Hand Trajectory Behavior

3.4.1 Constrained minimum-jerk movement

The human arm trajectory formation refers to the planning and control of the kinematic aspects of arm movements [92]. The trajectory here includes both the configuration of the arm in space and the higher derivatives of the movements. Skilled human arm/hand movements are usually smooth and developed through training and practice. Nelson suggests that in addition to meet the task oriented objectives, many of these skilled movements appear to satisfy more general physical principles under different constraints [93]. The objective measures which have been considered include time, force, impulse, energy, and jerk. Among them, the jerk, mathematically defined as the time derivative of acceleration, gets the most attention. Hogan studies the smooth trajectories under the constraints of minimum-jerk for the single-joint forearm movements in [94]. Flash and Hogan then generalize this principle to the multi-joint motions in [95]. It states that in moving from an initial to a final position in a given time duration from t_0 to t_f, the cost function to be minimized is

$$C_j = \frac{1}{2}\int_{t_0}^{t_f} \{(\frac{d^3x}{dt^3})^2 + (\frac{d^3y}{dt^3})^2 + (\frac{d^3z}{dt^3})^2\}dt, \tag{3.29}$$

where $(x(t),y(t),z(t))$ is the coordinate of the hand. The authors solve the optimized trajectory with calculus of variations. The solutions have a zero sixth derivative:

$$x^{(6)} = 0, \quad y^{(6)} = 0, \quad z^{(6)} = 0. \tag{3.30}$$

Their general expression is apparently a 5th order polynomial. The polynomial coefficients can be solved with the two boundary conditions and the time duration.

The minimum-jerk model can be considered as a special form of a more generalized hypothesis which is the minimization of the time integral of the squared nth-derivative of the coordinates of the hand. For each order $n \in (1, \infty)$, it corresponds to a different member of the family. Flash and Hogan report that the mean value of the peak-to-average velocity rate is close to 1.8 with a standard deviation of around 10 % based on 30 movement measurements which is close to the minimum-jerk cost function. The peak-to-average ratio increases as n increases and finally goes to infinity [95]. Therefore, the high-order velocity profiles are incompatible with the experiments. By calculating the predicted velocity profiles for different order of n, it is shown that only the minimum-jerk is compatible with the experimental data. Daohang *et al.* study reaching movements of human arm on a constraint semi-sphere surface in [41]. The subjects are asked to move from one point to another point on a virtual semi-sphere surface by holding a SensAble Phantom 3.0 haptic device. The experimental results are also consistent with the theoretical analysis of minimum-jerk reaching on the surface of a sphere.

3.4.2 The two-third power law

Since 1980s, there has been much research on the study of handwriting and drawing motions. In [96], Viviani and Terzuolo first found the relationship between the angular velocity ω and the instantaneous curvature k of the trajectory during the movement, widely referred to as the two-thirds power law. The relationship is formalized as:

$$\omega = ck^{1-\beta}, \tag{3.31}$$

where β is approximately 1/3 and the gain factor c is set by the tempo of the movement. If it is expressed in terms of tangential velocity v, it is

$$v = ck^{-\beta}. \tag{3.32}$$

Many experimental results from tracing and drawing different figures support the formulation [97] [98]. The law is then generalized to include the segmentation characteristics of the trajectory for extended shapes and free scribbling:

$$v = c_j \left(\frac{R}{1+\alpha R}\right)^{\beta} \quad \text{for each segment } j, \tag{3.33}$$

where $\alpha = 0.05$ and $R = 1/k$. In [99], subjects are asked to trace increasingly large ellipses with their fingers and the result shows that the larger the ellipse, the clearer the segmentation in the velocity profile. The authors note that the motions of the arm joints which have clear segmentations of endpoints in Cartesian space are purely oscillatory. This is due to the nonlinearities in the forward kinematics of the human arm. For the velocity or acceleration estimation, this implies that the dynamic model which fits well in the joint space maybe not fit in the Cartesian space.

3.5 Proposed Approaches

3.5.1 Estimation in Cartesian space

In haptic simulations, when the user holds the device's end effector and manipulates the virtual objects, the endpoint trajectory can be described in either joint space or Cartesian space. There are two ways to get the linear velocity in the Cartesian space. The first one is to estimate the joint velocity first and then convert it into the linear velocity with the device's Jacobian matrix (indirect approach). The second one is to compute the relative position of the end effector in Cartesian space first and estimate the linear velocity from these position measurements directly (direct approach). Theoretically, two methods should give the same results. But it is not the actual case because the estimation needs a model to describe the signal itself. For a low speed trajectory, they usually give very close results since Jacobian transformation can be well approximated by a linearized model. But for a high speed signal, the differences cannot be ignored. We recorded a motion trajectory using the OMNI haptic device and estimated the velocity with a triple-integrator Kalman filter. The resulting velocity profiles with these two approaches are plotted in Fig. 3.7. For this specific example, there are two main reasons for these large differences: motion in the joint space cannot be modeled by the triple-integrator model well, and the kinematic transformation between the joint space and the Cartesian space is nonlinear.

From the perspective of human arm trajectory formation, some researchers have argued that human arm/hand trajectories are planned in the joint variables of the arm while others state they are planned in Cartesian space; joint rotations are then tailored to produce these desired hand movements [100]. The second view has gained more support from the studies of planar, unconstrained human movements [92]. The simple experiment of asking subjects to move between two targets shows that the subjects generally tended to generate roughly straight trajectories with a single-peak, bell-shaped speed profile no matter where these two targets are located in the reachable space of the human arm.

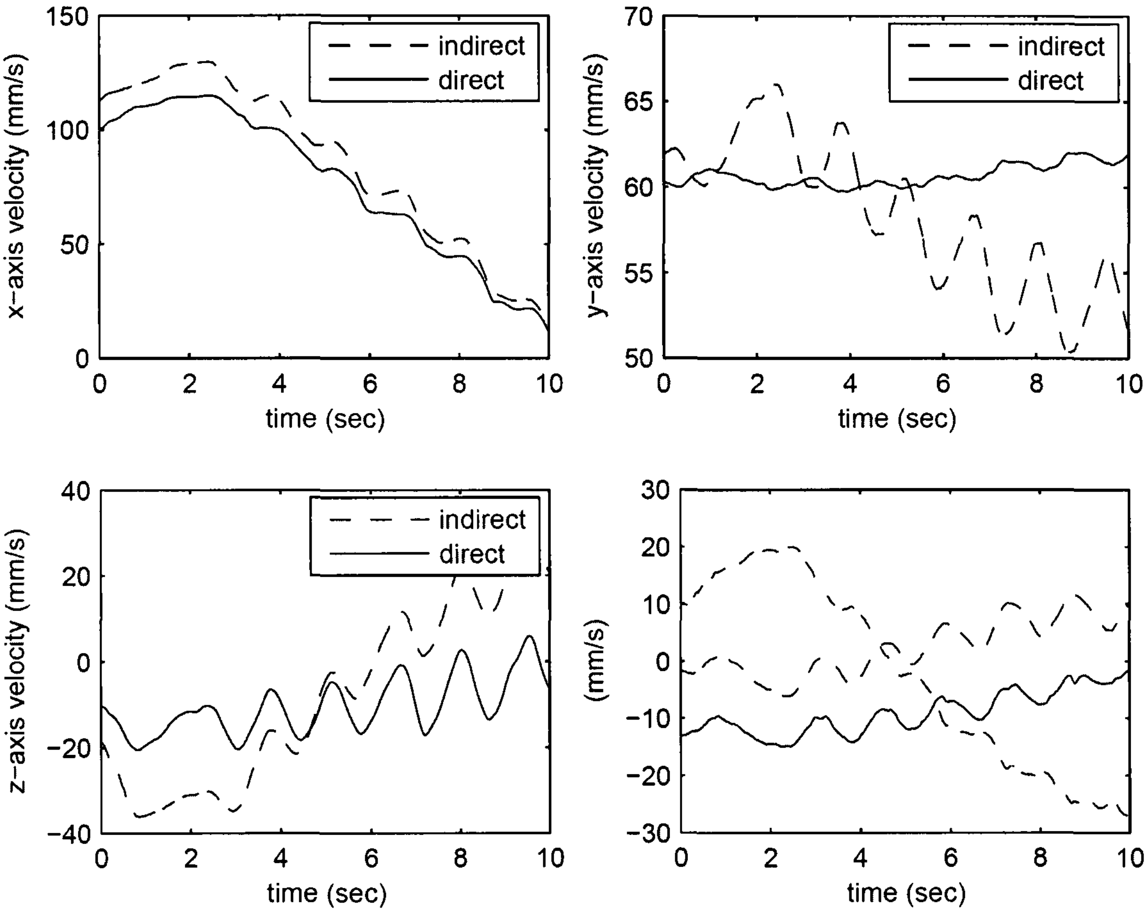


Figure 3.7: Comparison of the linear velocity estimation between the direct and indirect approach

These invariant features of movements are a strong indication that planning takes place in Cartesian space rather than joint rotations. Furthermore, as aforementioned, it is the user that moves the device in the Cartesian space inducing the rotations of the device's joints in the joint space. Therefore, a direct estimation in the Cartesian space is preferred.

3.5.2 Jerk constrained 4-State Kalman filter

As aforementioned, human arm/hand movement for a smooth trajectory has at least 5 non-zero derivatives. An accurate model of the motion should include all these derivatives. But a large number of derivatives make a model difficult to implement in real-time.

For example, a typical haptic servo loop runs around 1 KHz for a continuous force feedback. Therefore, commonly used models have less derivatives. The model considering the second derivative of the position is called acceleration model, while the model considering the rate of the acceleration is referred as to jerk model. In [101], Mehrotra *et al.* derive a full 4-state Kalman filter equation for highly maneuvering target tracking in which the third derivative of the target position, the jerk, is included. Since human arm/hand movement has at least 5 non-zero derivatives, it will be fair to expect the jerk filter to provide a higher accuracy than the acceleration filter.

Following Singer's model, the correlation function $r(\tau)$ associated with the jerk is

$$r(\tau) = E[j(t)j(t+\tau)] = \sigma_j^2 e^{-\alpha|\tau|}, \qquad \alpha \geq 0, \tag{3.34}$$

where σ_j^2 is the variance of the target jerk and α is the reciprocal of the maneuver time constant. Through Wiener-Kolmogorov whitening procedure, the jerk $j(t)$ is represented as a function driven by a white noise $w(t)$. The differential equation results from the whitening procedure is

$$\dot{j}(t) = -\alpha j(t) + w(t). \tag{3.35}$$

Then a full 4-state jerk model Kalman filter tracking equation for human arm movement in continuous time can be expressed as

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\alpha \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w(t), \tag{3.36}$$

where x, $\dot{x}$, $\ddot{x}$, $\dddot{x}$ denote the position, velocity, acceleration, and jerk of the target respectively. The derivation of the discrete model and the initialization parameters for Kalman filtering are detailed in [101]. For the special case that αT is small, the state transition matrix $\Phi(T, \alpha)$, the covariance matrix $Q(k)$ for the process noise, and the initial covariance matrix of error P reduces to

$$\lim_{\alpha T \to 0} \Phi(T, \alpha) = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.37}$$

$$\lim_{\alpha T \to 0} Q(k) = 2\alpha\sigma_j^2 \begin{bmatrix} \frac{T^7}{252} & \frac{T^6}{72} & \frac{T^5}{30} & \frac{T^4}{24} \\ \frac{T^6}{72} & \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^5}{30} & \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^4}{24} & \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix}, \tag{3.38}$$

$$\lim_{\alpha T \to 0} P = \begin{bmatrix} \sigma_m^2 & \frac{\sigma_m^2}{T} & \frac{\sigma_m^2}{T^2} & 0 \\ \frac{\sigma_m^2}{T} & \frac{2\sigma_m^2}{T} & \frac{3\sigma_m^2}{T^3} & \frac{5\sigma_j^2 T^2}{6} \\ \frac{\sigma_m^2}{T^2} & \frac{3\sigma_m^2}{T^3} & \frac{6\sigma_m^2}{T^4} & \sigma_j^2 T \\ 0 & \frac{5\sigma_j^2 T^2}{6} & \sigma_j^2 T & \sigma_j^2 \end{bmatrix}, \tag{3.39}$$

where σ_m^2 is the measurement noise variance. The filter assumes the jerk to be constant during the sampling interval T as αT is sufficiently small.

Considering the constrained minimum-jerk characteristic of the human arm/hand trajectory, we make σ_j adaptive within $[\sigma_j^{\min}, \sigma_j^{\max}]$. The estimation starts with the minimum value. After each update, the estimated position $\hat{x}(k)$ is compared with the measured position $x(k)$. If their absolute difference is larger than a predefined threshold Δ_t, it means that the current jerk variance is too small to capture the transient behavior of the trajectory. σ_j is then increased by Δ_j. On the other hand, if the absolute difference is smaller than the threshold, it is decreased by Δ_j to honor the minimum-jerk movement. Fig. 3.8 illustrates the architecture and the updating procedure of σ_j. Currently, the threshold Δ_t is set to the position resolution of the device. If the human arm movement is consistent with the constrained minimum-jerk, the model will give a decent velocity and acceleration estimation at each sampling instant. At the same time, the position estimation is within the bound of the resolution Δ_t. The 4-state Kalman filter in the Fig. 3.8 can be replaced with a 3-state Kalman filter which represents an adaptive constrained minimum-acceleration model.

3.5.3 Implementation of a Kalman filter

Essentially, Kalman filtering estimates a process by using a form of feedback control: the filter estimates the state of a dynamic system and then obtains feedback from measurements. As such, mathematical equations for the Kalman filtering fall in two groups: temporal update equations and measurement observational update equations.

The temporal propagation equations project the current state and error covariance estimates to obtain priori estimates for the next time step. Mathematically, the temporal propagation can be described by

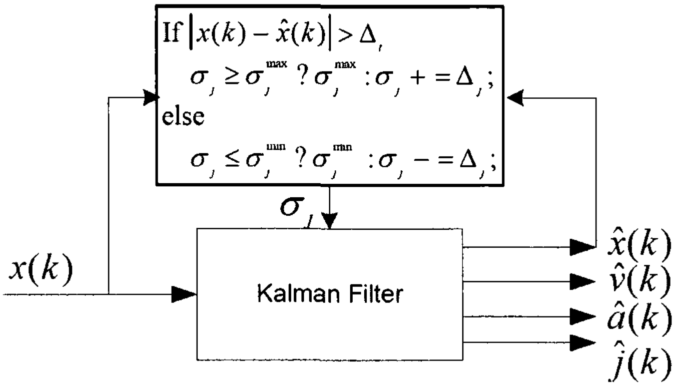


Figure 3.8: Jerk constrained 4-state Kalman filtering.

- One-step ahead state estimate prediction:

$$\hat{x}(k+1|k) = \Phi\hat{x}(k|k) + Gu(k), \tag{3.40}$$

where Φ is the state transition matrix, $\hat{x}(k|k)$ is the a posterior estimate of the state at kT, and $\hat{x}(k+1|k)$ is the a priori estimation of the state at $(k+1)T$.

- One-step ahead state estimate covariance prediction:

$$P(k+1|k) = \Phi P(k|k)\Phi^T + Q(k), \tag{3.41}$$

where $Q(k)$ is the covariance matrix of the process noise, $P(k|k)$ is the a posterior estimate of the error covariance matrix at kT, and $P(k+1|k)$ is the a priori estimation of the error covariance matrix at $(k+1)T$.

The observational update equations are responsible for the feedback, *i.e.*, incorporating a new measurement into a priori estimate to obtain an improved a posteriori estimate. The mathematical equations for the observation update include

- Kalman gain $K(k)$:

$$K(k) = P(k|k-1)H^T[HP(k|k-1)H^T + R(k)]^{-1}, \tag{3.42}$$

where H is the measurement matrix, and $R(k)$ is the covariance matrix of the measurement noise.

- A posteriori estimate of the state:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)[y(k) - Hx(k|k-1)], \tag{3.43}$$

where $y(k)$ is the measurement at kT.

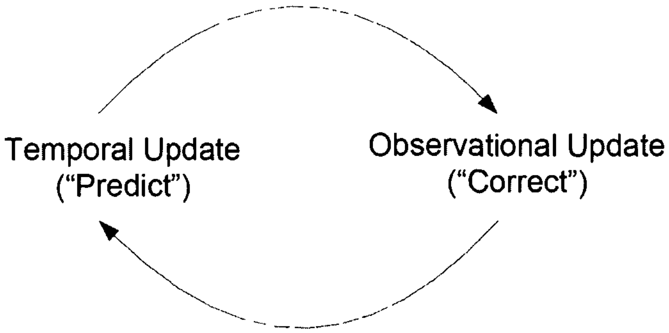


Figure 3.9: An ongoing discrete Kalman filter cycle. The temporal update projects the current state estimate ahead in time. The observational update adjusts the projected estimate by an actual measurement at that time.

- Estimation error covariance matrix $P(k|k)$:

$$P(k|k) = P(k|k-1) - K(k)HP(k|k-1). \tag{3.44}$$

The whole Kalman filtering resembles that of a predictor-corrector cycle as shown in Fig. 3.9.

Theoretically, the performance of Kalman filtering is characterized by the covariance matrix of estimation uncertainty. Soon after this conventional Kalman filter was implemented on digital computers, it was found that the observed mean-square estimation errors were often much larger than the value predicted by the covariance matrix. The variances of the filter estimation errors diverge from their theoretical values, *i.e.*, the solutions obtained for Eq. (3.44) have negative variances which are theoretically impossible. The problem was eventually determined to be caused by computer round-offs. Since that, many alternative implementation methods have been proposed to tackle the problem. The most successful approaches often represent the error covariance matrix P with symmetric products of triangular factors. One of them is a pair of algorithms, including the Bierman algorithm for the observational update of the modified Cholesky factors U and D of the error covariance matrix

$$P = UDU^T, \tag{3.45}$$

and the corresponding Thornton algorithm for the temporal update U of D and for the covariance matrix [102] [103]. Bierman's algorithm is one of the most stable implementations of the Kalman filter observational update . The UD factorization of the covariance matrix for the temporal update in the discrete Riccati equation is proposed by Catherine

Thornton. We adopt these two algorithms for the implementation of our proposed adaptive Kalman filter and the experimental results have shown that they are very stable. For interesting readers, the C source code is provided in Appendix B. A comprehensive but very practical coverage on Kalman filter can be found in [104].

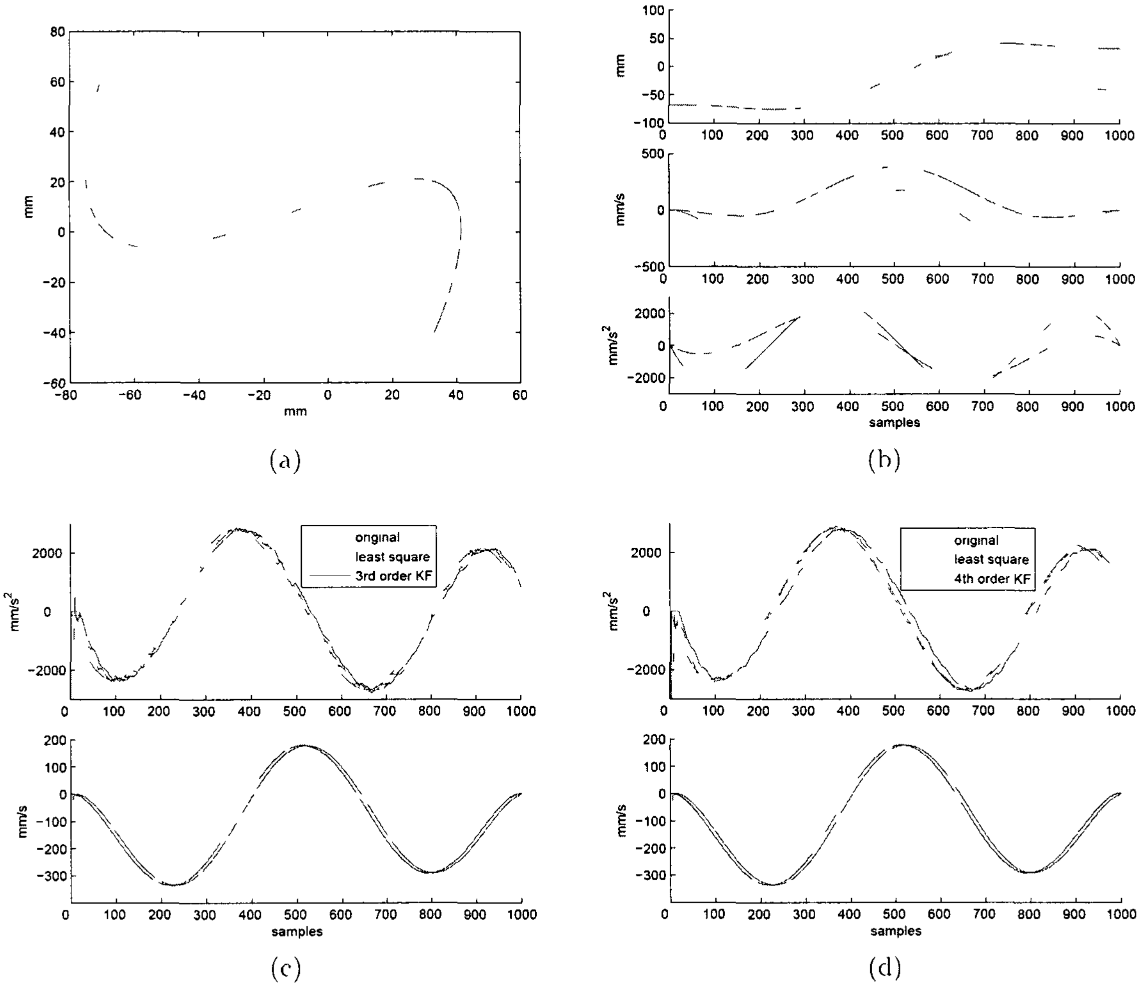
3.6 Simulation and Experimental Results

3.6.1 Simulation results

Fig. 3.10a plots a 2-D minimum-jerk trajectory with two via-points. The trajectory is sampled at 1 KHz and lasts one second. A uniform distributed noise $[-0.03, 0.03]$ mm is added to the original trajectory. The position, velocity and acceleration signals of the trajectory are shown in Fig. 3.10b. Two estimation methods are compared: adaptive 4-state Kalman filter and adaptive 3-state Kalman filter. Least-square method with a window size of 30 is taken as a reference since it gives a smoothed and delayed estimation. As shown in Fig. 3.10c and 3.10d, the proposed constrained minimum-jerk method outperforms both least-square and constrained minimum-acceleration method. For the velocity estimation, the proposed adaptive 4-state Kalman filter has a 0.047 mm/s mean error while the adaptive 3-state Kalman filter and least square method has a mean error of 0.194 mm/s and 1.535 mm/s respectively. For the acceleration estimation, the adaptive 4-state Kalman filtering gives a 3.648 mm/s^2 mean error which is a order better than the other two methods, a mean error of 32.811 mm/s^2 for the adaptive 3-state Kalman filtering and 58.503 mm/s^2 for least square method. In terms of the standard deviation, the proposed method also outperforms the other two methods. Table 3.4 lists the respective mean errors and the standard deviations. Interestingly, if the maneuvering acceleration variance tends to zero, the adaptive minimum-acceleration Kalman filtering will converge to the least square method.

3.6.2 Time delay compensation for haptic-enabled telementoring

In the section, we apply the proposed 4-state Kalman filter as a predictor to compensate network latencies in haptic-enabled telementoring. Telehaptics is the concept of networked haptics and occurs when a communications network links haptic devices at remote locations [105]. Haptic-enabled telementoring is a special form of bilateral tele-



(a) (b)

(c) (d)

Figure 3 10. Comparison of different velocity and acceleration estimation method for an exemplary minimum jerk trajectory: (a) 2-D minimum jerk trajectory; (b) the original trajectory; velocity and acceleration; (c) adaptive constrained minimum-jerk method, and, (d) adaptive constrained minimum-acceleration method

haptics in which one person takes on the role of a mentor while the other acts as a trainee. As Fig. 3 11 shows, a mentor aids a trainee in accomplishing the task and acquiring the essential skills

In the real-world, there are two correlated trajectories in a force sensorimotor skill. the movement and the force trajectory. These two information flows form a closed-loop system with the interacting environment and the user. The movement is caused by

Table 3.4: Estimation errors for an exemplary minimum-jerk trajectory.

method	adaptive range	velocity (mm/s)		acceleration (mm/s^2)	
		mean	std	mean	std
4-state KF	0.5~2.0 m/s^2	0.047	1.153	3.648	1.315e2
3-state KF	40~50 m/s^3	0.194	1.580	32.811	2.510e2
least square	M=30	1.535	17.086	58.503	3.984e2

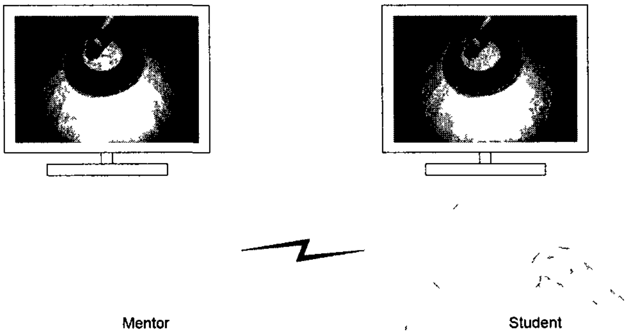


Figure 3.11: A conceptual view of haptic-enabled telementoring.

the muscle forces while the sensory inputs in turn modify the muscle forces with the user's intentions. The sensory inputs can be the visual cues, haptic cues, or combined hapto-visual cues from the interacting environment. Some experimental studies have shown that the spatio-temporal and force/torque profiles of expert users are significantly different from those of naive users [15]. Then what are the natural ways to teach these force sensorimotor skills in the real-world? They are usually taught in the following three steps:

1. The mentor performs the task and the trainee learns by visual observations. Let us take the English alphabets writing as an example. The teacher writes the characters on the blackboard so that the student can learn the sequence of each stroke. Verbal explanations usually accompany the teaching.

2. The mentor holds the trainee's hand which in turn holds the tool and performs the task. After the first step, the teacher lets the student try it himself/herself and verifies the result. If the teacher sees the outcome, he/she will hold the trainee's

hand and write the character again in a hand-holding manner. We call this hand-holding teaching in the thesis "haptic-enabled mentoring". Verbal explanations also usually accompany the teaching.

3. The trainee practices the skill by himself/herself. At the end of the class, the teacher asks the student to write the characters ten times as homework and hand it the next morning for verification.

If we take a look of the current haptic-enabled simulations, most of them just provide environments for the users so that they can practice the skills on their own which corresponds to Step 3. Step 1 is usually presented through video clips. But the second step is a very effective and natural paradigm for the teaching of some force sensorimotor skills, in which there is a direct contact between the mentor and the trainee. With this paradigm, the mentor may actively perform the task while the trainee is just passively guided. In this case, the trainee feels directly from the interacting objects and gets to know what should feel under the correct movements. On the other hand, the trainee can be active while the mentor is just passively holding the trainee's hand. In this case, the mentor feels a filtered force feedback from the interacting objects through the trainee's hand. Once the mentor notices the incorrect movements or feels the incorrectly exerted force pattern, he/she intervenes and brings the trainee back to the correct path. There have been some studies on the application of haptic feedback in the learning of spatio-temporal trajectories [106] [107] [108]. For interested readers, a more detailed discussion about these studies and the system architecture for the proposed haptic-enabled telementoring can be found in Appendix C.

For the purpose of providing this natural and effective teaching paradigm over the Internet for the training of sensorimotor skills, we combine the concepts of telementoring and haptic simulations together and term it "haptic-enabled telementoring". The advantages of the haptic-enabled telementoring are as follows:

1. It improves on the current haptic training system and provides a natural way to teach the sensormotor skills.

2. With haptic-enabled telementoring, both the mentor and the trainee hold the tool in a natural way instead of one of them feeling filtered dynamics of the interacting environment. The indirect perception of the environment might affect the procedures.

3. It extends the hand-holding teaching paradigm over the fast developing Internet.

Table 3.5: Legend for a haptic-enabled telementoring structure.

Master site S_m		Slave site S_s	
x_m	master state	x_s	slave state
x_{ds}	delayed x_s	x_{dm}	delayed x_m
x_{cs}	compensated x_{ds}	x_{cm}	compensated x_{dm}
F_{me}	simulation force	F_{se}	simulation force
F_{mh}	mentor exerting force	F_{sh}	mentee exerting force
F_{mc}	controller force	F_{sc}	controller force
T_h	sampling interval	T_l	transmission interval
MC	master controller	SC	slave controller
MP	master predictor	SP	slave predictor

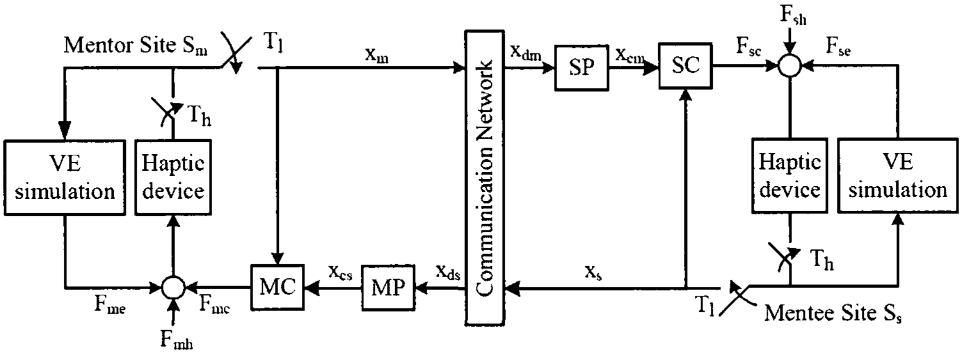


Figure 3.12: Haptic-enabled telementoring implementation diagram.

The detailed structure for the haptic-enabled telementoring is presented in Fig. 3.12 and its legends are listed in Table 3.5. During the telementoring, the mentor indirectly grabs the trainee's hand through the coupling of two devices, dragging the hand through a series of motions. This design reinforces the apprenticeship model, while allowing the mentor to have more active role in guiding the trainee. Ideally, the forces sent to the haptic device at mentor and trainee sites, respectively, are as follows:

$$\begin{aligned} F_m &= F_{mc} + F_{me} \\ F_s &= F_{sc} + F_{se} - F_{sh} \end{aligned} \tag{3.46}$$

F_{mc} and F_{sc} are the guiding forces for the tracking of two haptic devices held by the mentor and the trainee respectively. It is often implemented with a PD controller as

$$\begin{aligned} F_{mc} &= K_{mp}(x_{cs} - x_m) + K_{md}(\dot{x}_{cs} - \dot{x}_m) \\ F_{sc} &= K_{sp}(x_{cm} - x_s) + K_{sd}(\dot{x}_{cm} - \dot{x}_s), \end{aligned} \tag{3.47}$$

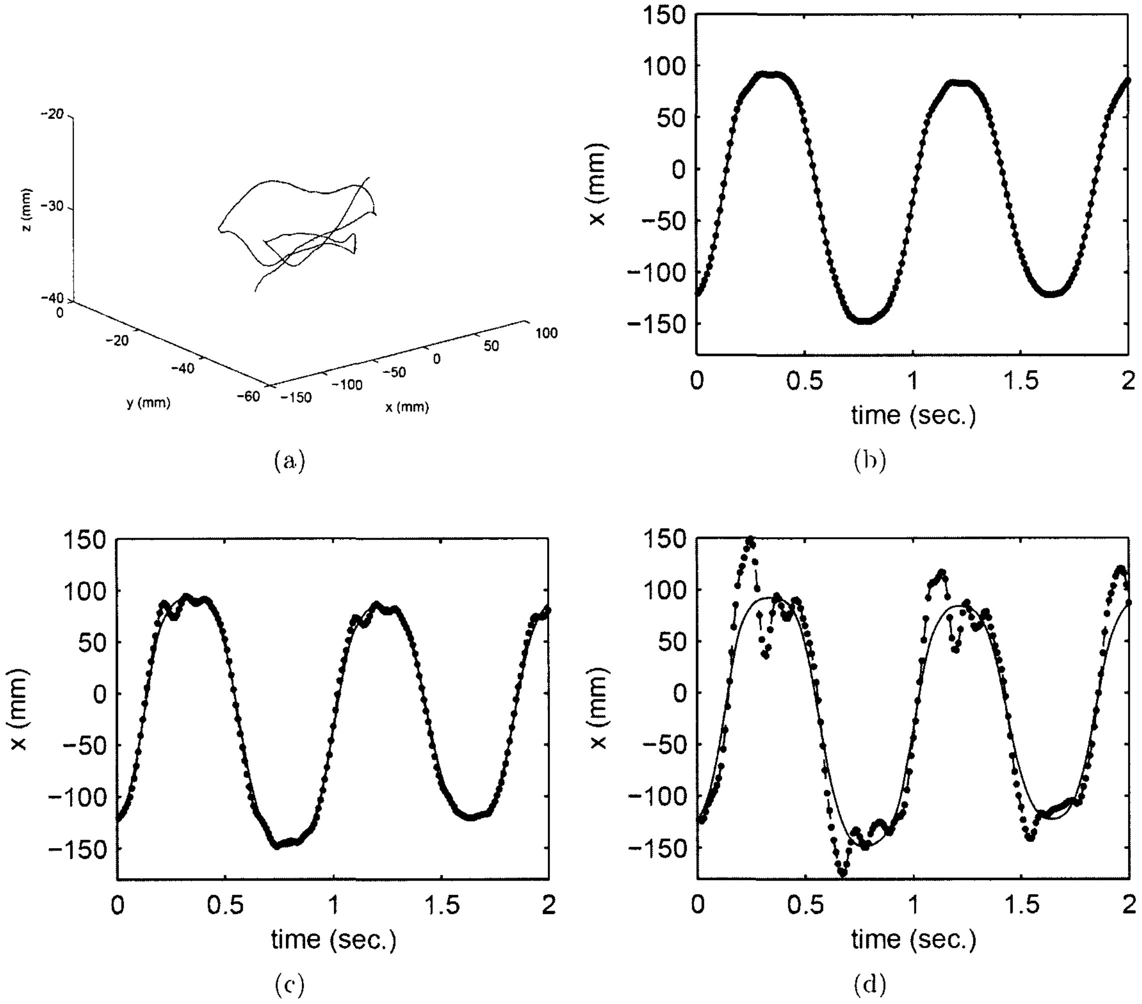
where K_{mp} and K_{sp} denote the adjustable proportional gains, and K_{md} and K_{sd} denote the derivative gains at two sites. One reason for an adjustable gain is to keep the stability under the effects of time delay. F_{sh} is the active force the trainee exerting on the device. If F_{sh} is measurable, it can be balanced out to make the trainee totally passive to the system. Since there is a closed-loop between two sites, any changes in F_{sc} will affect F_{mc} that, in turn, affect the mentor's feeling. For example, if the trainee does not follow the movement well, the guiding force will be large at both sites and the mentor may feel difficult to proceed to the task.

Transparency and stability are two critical issues in designing telehaptics or teleoperations. We propose an adaptive prediction solution to compensate for network latencies. Nonetheless, we take neither the stability nor the transparency into account from the perspective of control theories so far. Our idea is to predict the delayed haptic data to moderate the negative effects of the network latencies present to the upper layer applications. A more detailed discussion about the time delay effects and the design of the system can be found in Appendix C.3. Simply put, the solution consists of two stages:

- At the sender side, besides propagating current states to the receiver, the sender estimates velocity, acceleration, and jerk on the basis of its motion profile. All this information is sent to the receiver so it can predict the sender's future motion.

- At the receiver side, the receiver predicts the sender's state using the received information that is based on 4-state Kalman filter. The receiver then uses the predicted information in the simulation rather than waiting for the update, which could be delayed due to end-to-end latencies.

To predict the spatial update of a remote haptic device, we need an accurate dynamic motion model to describe the haptic state. As discussed previously, human arm/hand movement is better described as a constrained minimum-jerk model, so the proposed 4-state Kalman filter used for velocity and acceleration estimation is used for prediction as well. However, in cases where haptic devices move at a high speed, the prediction results give considerably high overshooting rates, especially at the turning points. The



(a) (b)

(c) (d)

Figure 3.13: Prediction results under different time delays using different methods. The solid line is the original signal and the dotted line represents the predicted signal: (a) the experimental trajectory in 3-D space; (b) 20 ms delay; (c) 50 ms delay; and, (d) 100 ms delay.

predicted signal with the Kalman filter is comparable with the original and the delayed trajectories. Even the standard deviation of the predicted signal is acceptable, but the maximum deviation will cause a stability problem. Generally, when the trajectory is slowly varying over time, the prediction of over 100 ms will give acceptable results by using the Kalman solution. As the speed goes up, it becomes more challenging, especially at the trajectory's turning point as shown in Fig. 3.13d. To alleviate the overshooting effects at the turning point of a high-speed trajectory (a sharp turn), let us consider human arm/hand movement characteristics again.

Objects usually slow before turning around. This phenomenon occurs when we drive, run, or write. If a human arm trajectory starts with a high speed and slows down quickly, the likelihood of it changing direction is high. This is consistent with the two-third law discussed previously. During the turning, the curvature gets larger and the velocity becomes lower. Based on this observation, we divide the prediction into two states: normal state and adaptive state. In the normal state, the Kalman filter is used for the prediction. Therefore, we have

$$\hat{\mathbf{x}}(n+N) = \mathbf{A}(NT_s)\hat{\mathbf{x}}(n) \tag{3.48}$$

where $\hat{\mathbf{x}}(n) = [\hat{p}(n)\,\hat{v}(n)\,\hat{a}(n)\,\hat{j}(n)]'$ describes the estimated position, velocity, acceleration and jerk at time nT_s, and $\mathbf{A}(NT_s$ denotes the state transition matrix of the Kalman filter over $N = T_d^{\max}/T_s$ steps. The transition of the state happens when the velocity is high and starts to slow. We can simply formulate this condition as $\hat{a}(n)\hat{v}(n) < 0$. Let s denote the state, and we can describe the state transition as

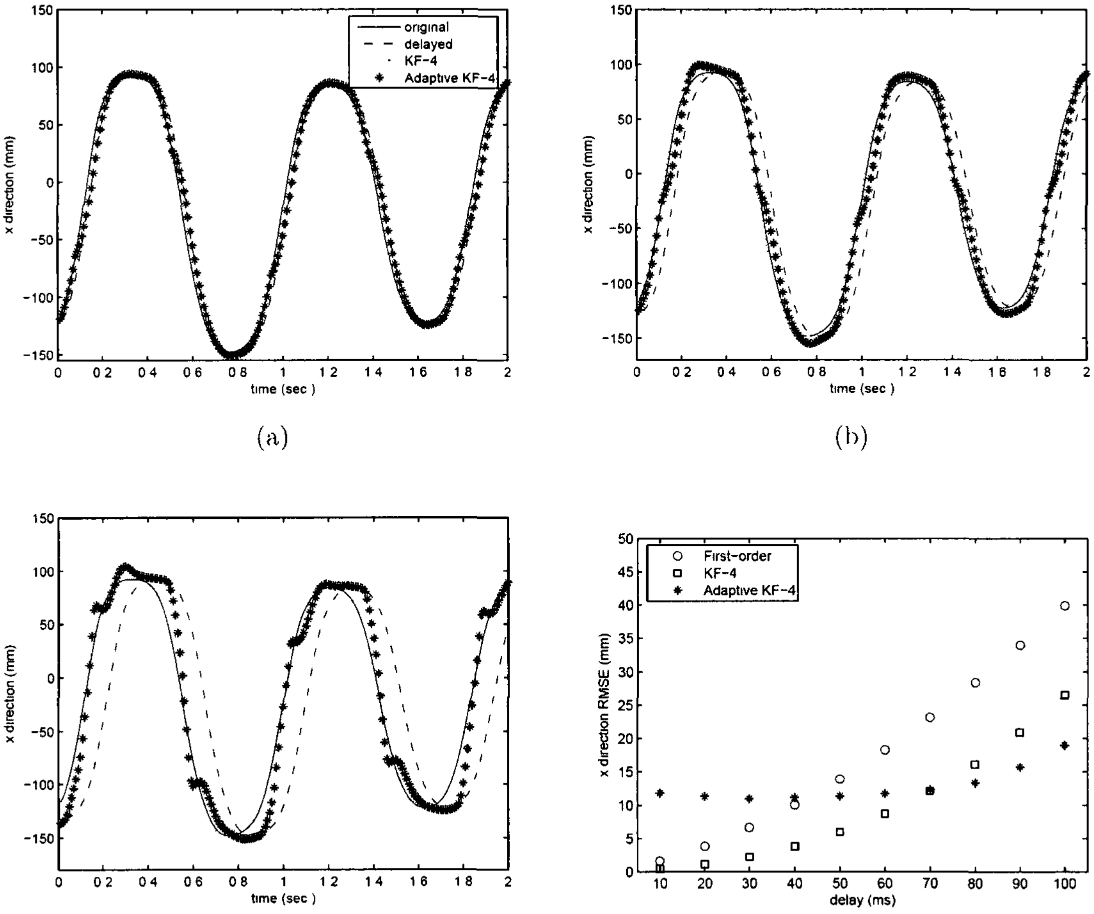
$$s = \begin{cases} \text{adaptive,} & \text{if } 500mm/s < |v| \ \wedge\ \hat{a}(n)\hat{v}(n) < 0; \\ \text{normal,} & \text{otherwise.} \end{cases} \tag{3.49}$$

During the adaptive state, we predict the position using

$$\hat{p}(n+1) = p(n) + \alpha T_s \hat{v}(n), \tag{3.50}$$

where $0.5 < \alpha < 1.0$ is a gain factor to smooth the transition and $\hat{v}(n)$ is computed using the least square method. Similarly, when the predictor switches back to the normal state, there is a gain factor to smooth the transition. The threshold value of 500 mm/s was chosed based on the experimental results.

Fig. 3.14 plots the prediction results for the same experimental trajectory using three different methods: the first-order velocity prediction, the 4-state Kalman filter prediction,



(a) (b)

(c) (d)

Figure 3.14: Adaptive prediction results under different time delays: (a) 20 *mm* delay; (b) 50 *mm* delay; (c) 100 *mm* delay; and, (d) root mean square error for time delays between 10 *mm* and 100 *mm*.

and the adaptive 4-state Kalman filter with the overshoot compensation. We present the results as a root mean square error (RMSE) in each movement direction (x, y, and z) to measure the accuracy in predicting the haptic trajectory under different delays. The position RMSE in each direction is calculated with the following formula:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(p(i) - \hat{p}(i))^2} \tag{3.51}$$

We notice that if we consider a combined RMSE in three directions together, the results may be misleading because the amplitude and velocity of the trajectory varies significantly in different directions. Fig. 3.14d shows the RMSEs for time delays in between 10 and 100 *ms*. In terms of the RMSE, the 4-state Kalman Filter performs the best if the time delay is under 70 *ms* while the adaptive 4-state Kalman Filter with the overshoot compensation performs the best if the delay is higher that 70 *ms*. However, the first-order velocity prediction performs better than the adaptive 4-State Kalman filter when the delay is under 40 *ms*. This is due to the transition period between the normal state and the adaptive state. These results suggest a real-time switching of the prediction method based on the current network latency.

3.7 Summary

In this chapter, we first explained the working principle of incremental rotary encoders used in most haptic interfaces, followed by the discussion about their resolutions. Then, we described a way to model the position resolution of a haptic interface using the device's forward kinematics. After that, we presented our studies on the current velocity and acceleration estimation methods. These studies suggest that good estimation filters should be adaptive to the characteristics of the trajectory. Since it is the user that manipulates the device's end effector, we then discussed two primary characteristics of human arm/hand trajectory: constrained minimum-jerk movement and two-third power law. The constrained minimum-jerk behaviour reflects the fact that skilful human arm/hand trajectories are often smooth while the two-third power law simply states that the velocity of human arm/hand trajectory usually slows down at turning points.

Based on the constrained minimum-jerk behaviour, we proposed a jerk constrained 4-state Kalman filter to estimate the velocity and acceleration for haptic interfaces. The simulation results showed that the proposed Kalman filter model outperforms the 3-state Kalman filter, particularly for the acceleration signal. We also modified the jerk constrained 4-state Kalman filter into a predictor to compensate for network latencies in a haptic-enabled telementoring scenario. In case of experiencing a high network time delay and a high speed trajectory, the predictor is switched in between the 4-state Kalman filter and a regular velocity prediction filter to avoid a large prediction overshoot. The experimental results showed that the online switching prediction performs much better than using the velocity information only.

Chapter 4

Uniform Hardness Perception in 6-DOF Haptic Rendering

4.1 Introduction

In Chapter 2, we reviewed the research works aiming at improving the perceived hardness in haptic rendering. In this chapter, we study something related but not the same. Our goal is to make users have a uniform hardness perception in 6-DOF haptic rendering. Through experiments, we have found that the induced forces from a haptic interface's effective mass and viscous damping at its active end affect the perceived hardness non-uniformly as users tap the virtual objects with different parts of the tool. As discussed in Chapter 1, collision points between a virtual tool and its interacting virtual objects can be time-varying in 6-DOF haptic rendering. This non-uniform hardness perception is also related to the positions of the user's holding point and the motor control point of the device. Based on the experimental results and theoretical analysis, we propose three compensation methods in this work and evaluate their performances with a stick-on-ball simulation. A total of six stiffness intensities in the range of 0.1-0.6 N/mm at a step of 0.1 N/mm were randomly presented to a total of 10 test subjects. Experimental results show both the need for compensation and the effectiveness of the proposed methods.

In the rest of chapter, we first discuss the role of the user's holding point and the motor's control point in 6-DOF haptic rendering in Sec. 4.2. The maximum linear stiffness at the collision point on the stylus is also described. Sec. 4.3 details the effects of the induced forces from the effective mass and viscous damping on the perceived hardness in two steps: theoretical analysis, and identification of these physical parameters for a 6-

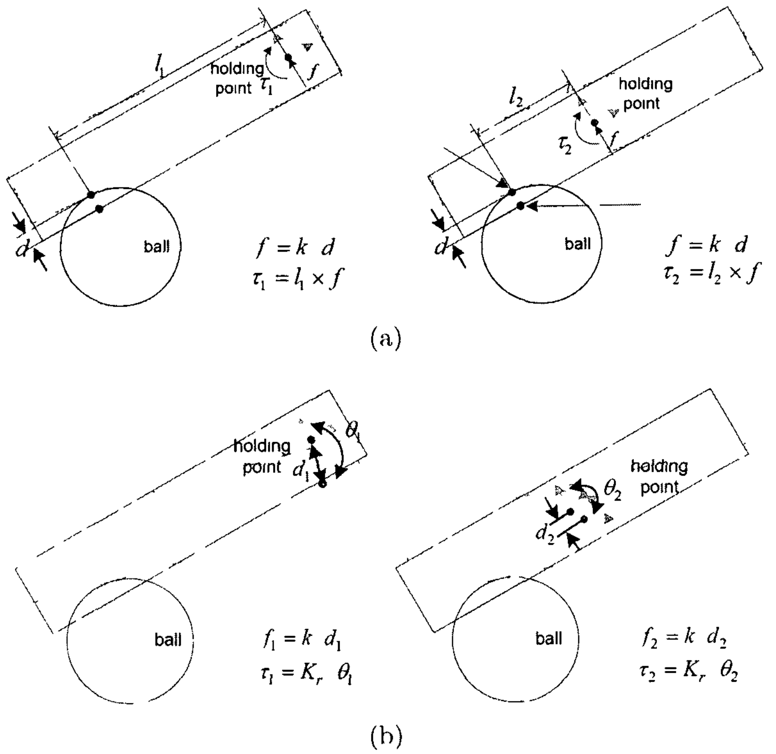


Figure 4.1: Primary holding point affects force/torque feedback in 6-DOF haptic rendering. The user taps a stationary rigid ball with a stick. Solid line represents the end effector (device tool) and stippled line represents the simulated stick (virtual tool): (a) direct rendering; and, (b) simulation-based rendering.

DOF haptic interface. Following that, Sec. 4.4 presents the three proposed compensation methods. The subjective evaluation experiments and results are discussed in Sec. 4.5.

4.2 Three Different Points in 6-DOF Haptic Rendering

4.2.1 Primary holding point

Currently, there are two main methods to compute interaction force/torque for 6-DOF haptic rendering: direct rendering and simulation-based rendering [109]. Let us take a stick-on-ball example to explain the role of a user's primary holding point in these two methods. The example is described in Fig. 4.1, where the user taps a stationary rigid ball with a stick.

With the direct rendering as illustrated in Fig. 4.1a, the virtual stick (virtual tool) follows the motion of the stylus (device tool) in the free space, but is constrained on the surface of the virtual object during the collision. When the stick touches the ball, the collision point on the surface of the stick with the deepest penetration and its corresponding contact point on the surface of the ball are first determined. As aforementioned, a feedback force $f \in \mathbb{R}^3$ is calculated using the linear Hooke's law

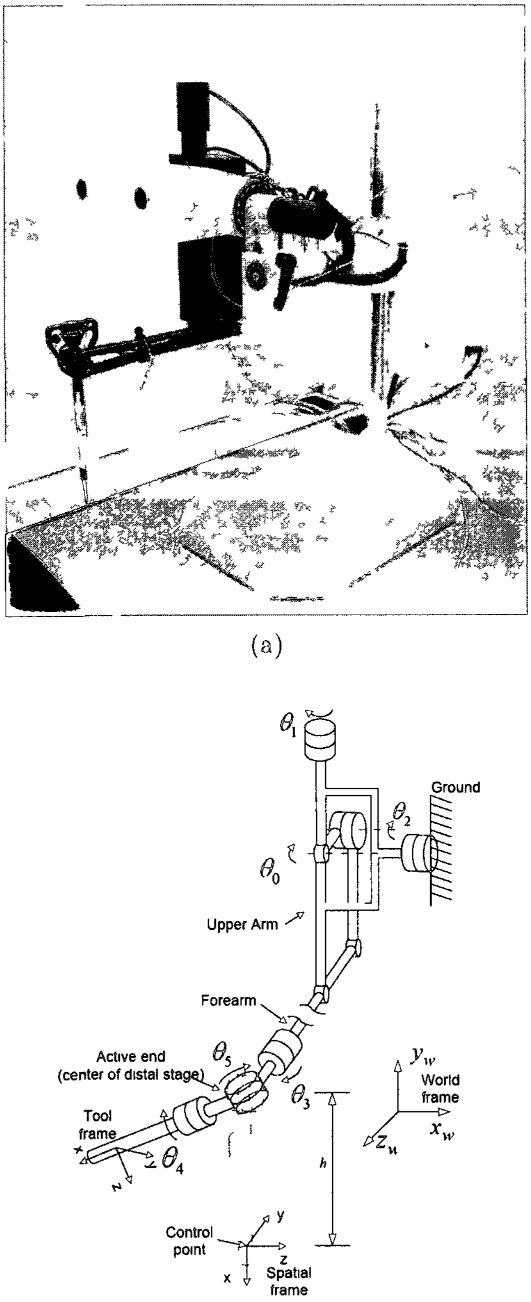
$$f = kd, \tag{4.1}$$

where $d \in \mathbb{R}^3$ is the penetration vector between the two points, and k is the stiffness constant. The larger the stiffness constant, the harder the user feels the ball. Meanwhile, the force will generate a torque τ around a pre-defined holding point on the virtual tool. This force/torque pair $F = [f \quad \tau]^T \in \mathbb{R}^6$ tries to push the stick out of the ball so the user perceives the collision and the hardness of the ball. Here, different holding points will result in different feedback torque components.

With the simulation-based rendering as illustrated in Fig. 4.1b, the feedback force/-torque is computed by applying a rigid-body simulation on the virtual tool. At the beginning of each rendering cycle, the current motion of the stylus (device tool) is converted to a force/torque pair acting on the virtual stick (computed in the previous cycle) by setting a viscoelastic coupling in between. This coupling force/torque is added to the collision force/torque between the virtual stick (computed in the previous cycle) and the virtual ball. The resultant force/toque is then applied on the virtual stick and Newton's second law determines the resulting linear and rotational motions of the virtual stick. The final feedback force/torque to the user is re-calculated as the viscoelastic coupling between the updated virtual stick and the current end effector (device tool). Basically, the final force/torque pair with the simulation-based haptic rendering is to align two coordinate systems together: one attached to the device tool, and the other attached to the virtual tool. Here, the location of the primary holding point affects both feedback force and torque.

4.2.2 Motor control point for force/torque rendering

With the feedback force/torque ready, it is time to render it onto the end effector of the device. In this chapter, a Freedom6S hand controller from MPB Technologies Inc. is used for all the discussions and experiments. At its active end, the device can generate both



(a)

(b)

Figure 4.2 (a) A MPB Freedom6S Hand Controller from MPB Technologies Inc. and (b) A conceptual view of the serial links and six joints.

translational force and rotational torque (twist force). As depicted in Fig. 4.2, the device has a pen-like stylus with one end connected to its active end. During the initialization stage, programmers can specify a tool frame on the stylus which is co-located with a fixed spatial frame (home frame) in the workspace. The forward kinematics retrieved from the device driver describes the relative configuration of this moving tool frame to the fixed spatial frame. Let $F_t = [f_t \quad \tau_t]^T \in \mathbb{R}^6$ denotes the feedback force/torque written in the tool frame. The required joint torques to render F_t onto the end effector are given by

$$\tau_J = (J^b(\theta))^T F_t, \tag{4.2}$$

where $\theta \in \mathbb{R}^n$ denotes the current joint angles, and $J^b(\theta) \in \mathbb{R}^{6\times n}$ is the current body Jacobian matrix ($n = 6$ for the Freedom6S hand controller). The origin of the tool frame is referred to as the motor control point, since the Jacobian matrix is written with respect to it. Expanding Eq. (4.2) for the Freedom6S controller, it gives

$$J^b(\theta) = \begin{bmatrix} \xi_0^\dagger & \xi_1^\dagger & \xi_2^\dagger & \begin{matrix} 0 & 0 & 0 \\ -h\cos(\theta_5)\sin(\theta_4) & 0 & h\cos(\theta_4) \\ -h\cos(\theta_5)\cos(\theta_4) & 0 & -h\sin(\theta_4) \\ \sin(\theta_5) & 1 & 0 \\ \cos(\theta_5)\cos(\theta_4) & 0 & \sin(\theta_4) \\ \cos(\theta_5)\sin(\theta_4) & 0 & \cos(\theta_4) \end{matrix} \end{bmatrix}, \tag{4.3}$$

where h is the distance from the active end to the control point (Fig. 4.2b). Each column of $J^b(\theta)$ corresponds to a joint twist written with respect to the tool frame in the current configuration [110]. $\xi_0^\dagger$, $\xi_1^\dagger$, $\xi_2^\dagger$ are not shown here because of their lengthy expressions. From Eq. (4.2), we may see that the correct joint torques can be derived only if the holding point conforms to the control point. Should an unmatched $J^{b'}(\theta)$ be applied, the user would perceive a distorted $F_t' = ((J^b)^T)^{-1}(J^{b'})^T F_t$ at the holding point rather than the desired F_t. We can roughly estimate the largest tolerable discrepancy between these two points based on the human just noticeable difference (JND) in torque, which is around 12.7% when the reference signal is in the range of 60-90 mNm for a pinch grasp [111]. With the same stick-on-ball application, if the holding point is 70 mm away from the collision point, a 1 N contact force will generate a 70 mNm torque around the holding point. If it is 60 mm away, a 60 mNm torque will be generated. In this case, the allowed mismatch between the holding point and the control point is approximately 10 mm.

However, to derive a single holding point from a user's handling postures on a tool is difficult, particularly for those involving multi-fingers and time-varying finger forces.

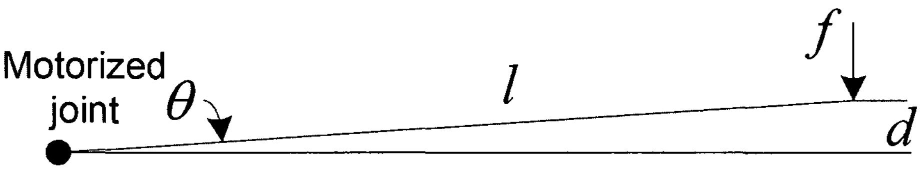


Figure 4.3: Relationship between angular stiffness and linear stiffness.

In fact, we can avoid this problem by fixing a control point on the stylus and always write F_t with respect to it. In turn, no matter how the tool is handled, the actual force/torque perceived by the user will be naturally induced from F_t. The only tradeoff is that application developers have to compute F_t for the fixed control point instead of the actual holding point, which may not be intuitive. In the rest of the chapter, the control point is fixed at the center of the distal stage ($h = 0$). Now the device Jacobian matrix becomes

$$J^b(\theta) = \begin{bmatrix} & & & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ \xi_0^\dagger & \xi_1^\dagger & \xi_2^\dagger & 0 & 0 & 0 \\ & & & \sin(\theta_5) & 1 & 0 \\ & & & \cos(\theta_5)\cos(\theta_4) & 0 & \sin(\theta_4) \\ & & & \cos(\theta_5)\sin(\theta_4) & 0 & \cos(\theta_4) \end{bmatrix}. \tag{4.4}$$

4.2.3 Maximum linear stiffness at collision point

Through experiments, we have noticed that users can feel vibrations and hear uncomfortable squeaking sounds as the collision point (the point of force application) moves away from the center of the distal stage. Here, we explain it with the relationship between the achievable linear stiffness at the collision point and the corresponding angular stiffness.

Let us look at a 1-DOF example as shown in Fig. 4.3, where a collision happens at the end of a lever with a penetration depth of d. Since the motorized joint (the rotation axis) is a distance l away, the angular motion θ is approximately d/l. If we simulate a linear stiffness of k at the collision point, a feedback force $f = kd$ is then required. This implies that a joint torque of $\tau = lf$ is needed to generate f at the collision point. In turn, the required angular stiffness K_θ at the motorized joint is given by

$$K_\theta = \tau/\theta = l^2 \cdot K. \tag{4.5}$$

Eq. (4.5) states that as the collision point moves away from the rotational axis, the scaling factor l^2 makes the achievable linear stiffness decrease sharply. As in 6-DOF haptic rendering the collision point can be anywhere on the tool, it is important to know the maximum achievable linear stiffness $k_{\max}$ at the collision point to guarantee the stability of the simulation. The Freedom6S controller has a linear stiffness around 1000 N/m at the center of distal stage while the maximum angular stiffness on the stylus are 4.0, 0.2, and 2.5 Nm/rad for pitch, roll, and yaw respectively. Referring to Fig. 4.3, the maximum achievable linear stiffness $k_{\max}$ at the center of the distal stage is determined by the angular stiffness of the first three joints (Joint 0, 1, and 2) where the motors are more powerful and directly drive the respective joint, *i.e.*, the gear ratio is one. But for the pitch, roll, and yaw of the end effector, they are driven by the respective motor indirectly through some tendons. For instance, if the collision point is 0.1 m away from the center of the distal stage on the stylus, then

$$k_{\max} = \min(1000, \frac{4.0}{0.1^2}, \frac{0.2}{0^2}, \frac{2.5}{0^2}) = 400\ N/m. \tag{4.6}$$

For another popular haptic interface, PHANToM 3.0, the angular stiffness for pitch, roll, and yaw are not given in the specification sheet. Nonetheless, this is a generic problem for such kinds of 6-DOF haptic interfaces.

4.3 Problem Statements

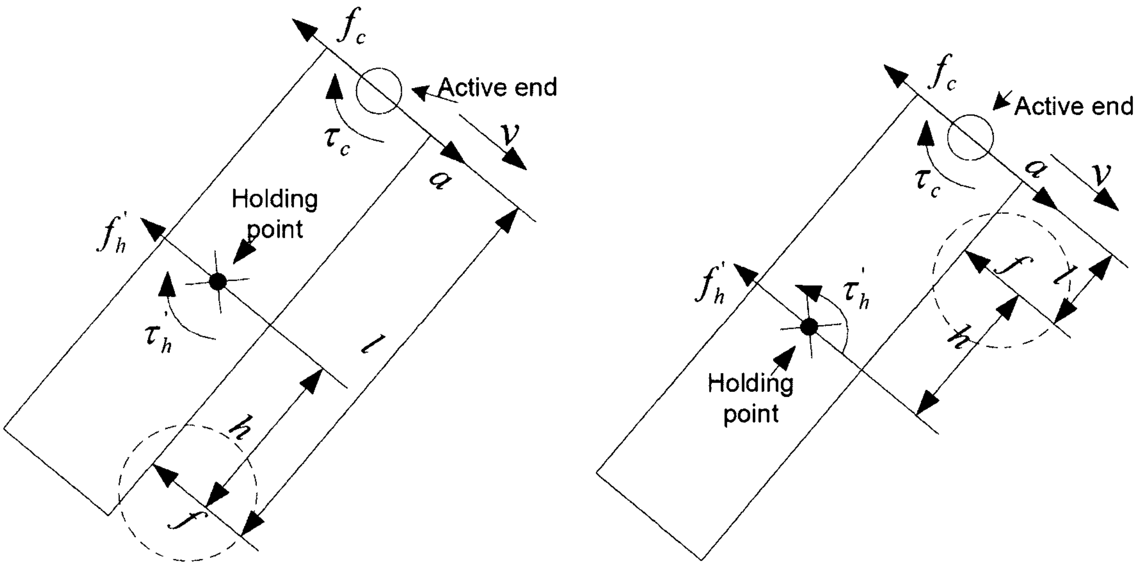
4.3.1 Theoretical analysis

As shown in Fig. 4.2, one end of the stylus is connected to the device at the center of distal stage while the other end has nothing connected to it. As the user manipulates the stylus, he/she will feel being dragged by the end which connects to the device. This kind of impedance can be described by the equivalent physical parameters at the active end, such as the effective mass, viscous damping, and Coulomb friction. In this section, we discuss the effects of these physical parameters on the hardness perception of the simulated virtual objects with two exemplary cases. For simplicity, we assume a pinch grasp hand posture on the stylus, in which a single holding point can be easily identified.

Let us again take the stick-on-ball as an example. As shown in Fig. 4.4, a contact force f acts on the stick when the stick touches the ball. The equivalent wrenches of f at the active end and the holding point are

$$F_c = [f_c \quad \tau_c]^T = [f \quad l \times f]^T \tag{4.7}$$

$$F_h = [f_h \quad \tau_h]^T = [f \quad h \times f]^T. \tag{4.8}$$



(a) Case I

(b) Case II

Figure 4.4: Two cases where users feel different hardness: (a) the collision point is between the holding point and the free end of the stylus; and, (b) the collision point is between the active end of the device and the holding point.

Let us assume that F_c can be faithfully rendered through the motors. If the stick is at rest, the user will perceive F_h around the holding point. But during the contact transition period, things are different.

Case I: The collision point is between the holding point and the free end of the stylus as shown in Fig. 4.4a. Let us assume that the stick moves toward the ball with an incoming velocity v and acceleration a. During the contact transition period, the actual force/torque perceived at the user holding point is

$$\begin{aligned} F_h' &= \begin{bmatrix} f_h' & \tau_h' \end{bmatrix}^T \\ &= \begin{bmatrix} f - (ma + bv) & \tau_c - (l - h) \times (f - (ma + bv)) \end{bmatrix}^T \\ &= \begin{bmatrix} f - (ma + bv) & h \times f + (l - h) \times (ma + bv) \end{bmatrix}^T, \end{aligned} \tag{4.9}$$

where m and b denotes the effective mass and the viscous damping of the device at the center of the distal stage. Comparing Eq. (4.8) with Eq. (4.9), we can see that the translational force at the holding point f_h' is different from f_h by the induced force

$-(ma+bv)$, while the torque around the holding point τ_h' is different from τ_h by an induced torque $(l-h) \times (ma+bv)$. However, f_h' and τ_h' always vary in the opposite direction, *i.e.*, one is increasing while the other is decreasing. For example, if $f \cdot (ma+bv) < 0$ (dot product), the user will feel a smaller force but a larger torque. The unmatched force/torque pair generates a fast clockwise rotation around the holding point which makes the user feel a stiffer virtual object than it should be.

Case II: The collision point is now between the holding point and the active end of the device as shown in Fig. 4.4b. Now, the equivalent wrenches for at the center of the distal stage and the holding point become

$$F_c = [f_c \quad \tau_c]^T = [f \quad l \times f]^T \tag{4.10}$$

$$F_h = [f_h \quad \tau_h]^T = [f \quad -h \times f]^T. \tag{4.11}$$

During the contact transition period, the actual force/torque felt at the holding point is

$$\begin{aligned} F_h' &= \left[f_h' \quad \tau_h'\right]^T \\ &= \left[f-(ma+bv) \quad \tau_c-(l+h)\times(f-(ma+bv))\right]^T \\ &= \left[f-(ma+bv) \quad -h\times f+(l+h)\times(ma+bv)\right]^T. \end{aligned} \tag{4.12}$$

Comparing Eq. (4.11) and Eq. (4.12), f_h' is also different from f_h by $-(ma+bv)$ and τ_h' is different from τ_h by $(l+h) \times (ma+bv)$. Different from Case I, they always change in the same direction, *i.e.*, one is increasing and the other is also increasing. Now, if $f \cdot (ma+bv) < 0$, the user still feels a smaller translational force but also a smaller torque. Under the same situation, we would expect a counterclockwise rotation around the holding point in the real-world. But due to the counterbalance from the induced torque term, the user will not perceive such rotational movements. For that, the unmatched force/torque pair makes the user feel a softer virtual object than it should be.

In summary, the distinct hardness perception of the simulated object in these two cases is due to the discrepancy of the holding point and the active end of the device. No matter where the collision point (the point of force application) is on the virtual tool, the force/toque perceived at the holding point is delivered by the equivalent force/torque generated at the active end of the device. However, the effective mass and viscous damping induced force at the active end generates an additional torque term around the user's holding point, which causes the distinct hardness perception as the collision point is at different sides of the holding point. In comparison, this problem does not exist in 3-DOF haptic rendering because there is only one single interaction point on the virtual tool and the user's holding point has no effects on the feedback force. In that case, if the induced forces from those physical parameters at the active end are not too large, the user will adapt to it in a short time.

4.3.2 Measurements of effective mass and viscous damping

As illustrated in Fig. 4.2b,the movements of the upper arm and forearm of the Freedom6S are controlled by Joint 0, 1 and 2. As users manipulate the stylus, the angular positions of these three joints determine the translational position of the center of the distal stage in the workspace. We first experimentally evaluate the dynamic parameters for Joint 0, 1 and 2, and then convert these joint space parameters to the equivalent Cartesian space ones at the center of the distal stage. However, to fully characterize these dynamic parameters across the whole workspace is not an easy task for haptic interfaces [112]. Here, we limit our estimation when the center of the distal stage is co-located with the center of the workspace.

In robotics, a motor-driven mechanical joint is often modeled as

$$J\ddot{\theta}(t) + B_r\dot{\theta}(t) + \sigma_r \operatorname{sgn}(\dot{\theta}(t)) = T(t), \tag{4.13}$$

where J is the joint inertia, B_r is the rotational damping coefficient, σ_r is the rotational Coulomb friction, T is the driving torque, and t is the time variable. But this model often needs to be modified for a specific device. Through experiments, we found that the dynamics of Joint 0 for the Freedom6S is dominated by the gravity of the upper arm, the forearm, and the stylus altogether. We change the dynamic model for Joint 0 as

$$T_0(t) = G_0 \sin(\theta_0(t)). \tag{4.14}$$

To quantify the gravity effect for later compensation, we adopted an approach proposed by Ma and Hollerbach in [113] by rotating the joint in step for its full range and fitting the result data of joint angle and the corresponding driving torque to Eq. (4.14). The parameter G_0 was then extracted using the regular least square method to be 0.3855 Nm. The comparison between the experimental and the simulated results is shown in Fig. 4.5.

For Joint 1 and 2, their dynamics are much more complex. We have noticed that if there is no external force acting on the two joints, they will return to their equilibrium

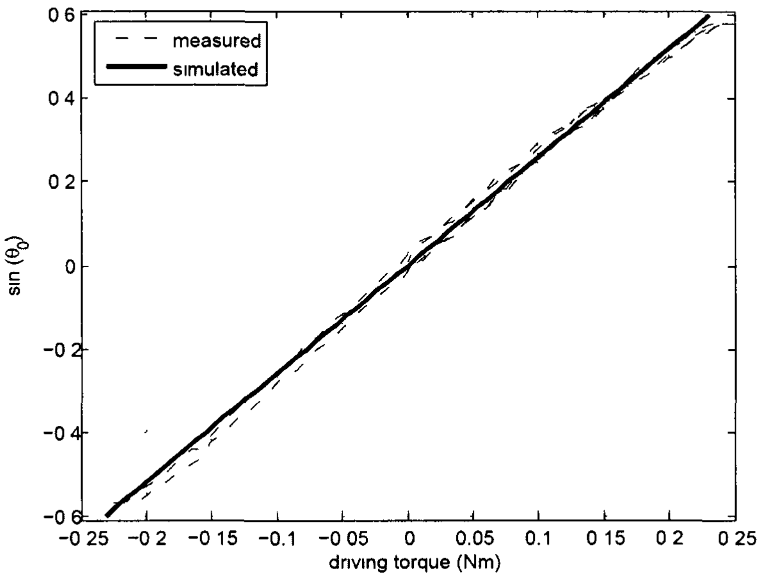


Figure 4.5: The dynamics of Joint 0 is dominated by the gravity of the upper arm, the forearm, and the stylus. The figure plots the relationship between the gravity and the driving torque.

position. This is mainly due to the elasticity of the tendons passed through the pulleys and the self balance design of the whole arm structure around the center of the distal stage. Therefore, we include an additional spring term in Eq. (4.13), which gives

$$J\ddot{\theta}(t) + B_r\dot{\theta}(t) + K_r\theta(t) + \sigma_r \operatorname{sgn}(\dot{\theta}(t)) = T(t), \tag{4.15}$$

where K_r is the rotational stiffness of the joint. To make the estimation of the parameters easier, we decide to linearize Eq. (4.15) first. Let us assume an input $\theta(t) = A\sin(\omega t)$. Substituting it into Eq. (4.15), we get

$$\begin{aligned} T(t) = &-JA\omega^2 \sin(\omega t) + B_rA\omega\cos(\omega t) + \\ &K_rA\sin(\omega t) + \sigma_r \operatorname{sgn}(A\omega\cos(\omega t)). \end{aligned} \tag{4.16}$$

Following the same simplification method used in [61], we approximate the nonlinear Coulomb term by neglecting its higher order harmonics

$$\operatorname{sgn}(A\omega\cos(\omega t)) \approx \frac{4}{\pi}\cos(\omega t). \tag{4.17}$$

Now the output $T(t)$ becomes

$$\begin{aligned} T(t) &= A[(-J\omega^2 + K_r)\sin(\omega t) + (B_r\omega + \frac{4\sigma_r}{\pi A})\cos(\omega t)] \\ &= M\sin(\omega t + \phi). \end{aligned} \tag{4.18}$$

We can further write this input-output relation using the concept of describing function which is defined as the complex ratio of the fundamental component of the nonlinear element by the input sinusoid [71]. It yields

$$\begin{aligned} N(A,\omega) &= \frac{Me^{j(\omega t+\phi)}}{Ae^{j\omega t}} = \frac{M}{A}e^{j\phi} \\ &= (-J\omega^2 + K_r) + j(B_r\omega + \frac{4\sigma_r}{\pi A}). \end{aligned} \tag{4.19}$$

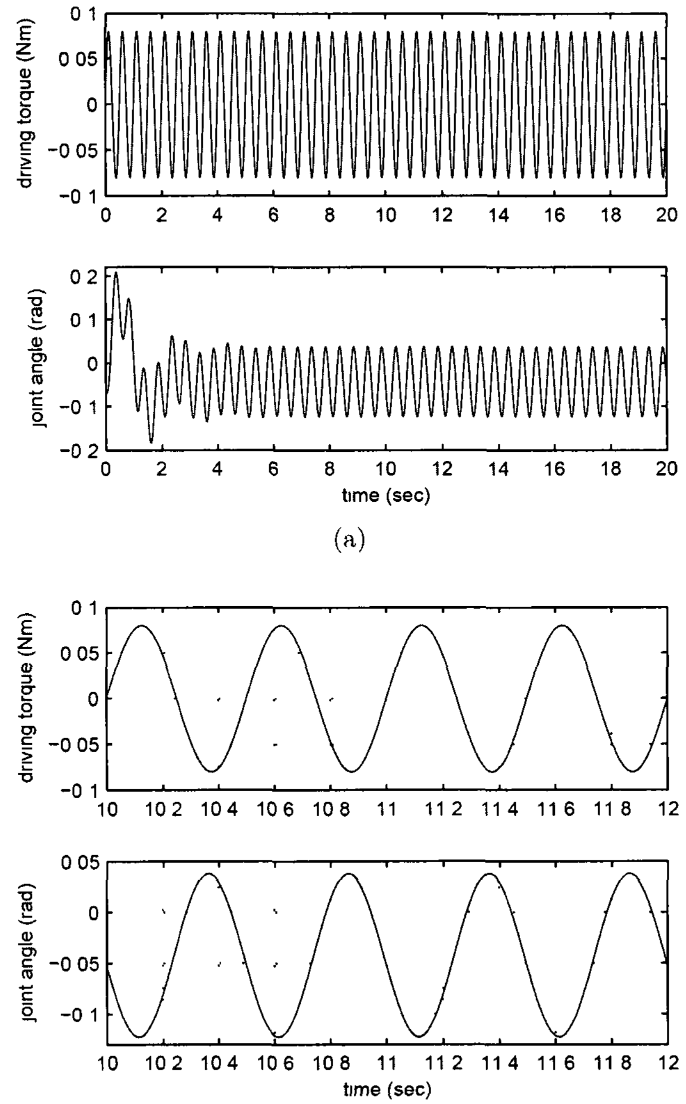
We may rewrite (4.18) in a matrix form as

$$\begin{bmatrix} \frac{M}{A}\cos(\phi) \\ \frac{M}{A}\sin(\phi) \end{bmatrix} = \begin{bmatrix} -\omega^2 & 1 & 0 & 0 \\ 0 & 0 & \omega & \frac{4}{\pi A} \end{bmatrix} \begin{bmatrix} J \\ K_r \\ B_r \\ \sigma_r \end{bmatrix}. \tag{4.20}$$

Here the whole model is treated as a lumped nonlinear element. As explained in [71], with a describing function, the nonlinear element can be considered as if it were a linear element with a frequency response function $N(A, \omega)$ in the presence of sinusoidal input. Unlike the frequency response of a linear system, describing function of a nonlinear element depends on the input magnitude A. With the describing function in the hand, the parameters for each joint can be experimentally determined. First, a set of excitation sinusoidal signals

$$T(t) = M\sin(\omega t) \tag{4.21}$$

with different M and ω are sent to each joint and the resulting joint angular motions are recorded. From each recorded angular signal, its amplitude A and the phase lag ϕ to its excitation signal can be determined. Finally, the least square method is applied to estimate J, K_r, B_r, and σ_r using Eq. (4.20). An excellent discussion about this technique and the experimental procedures is presented in [114].



(a)

(b)

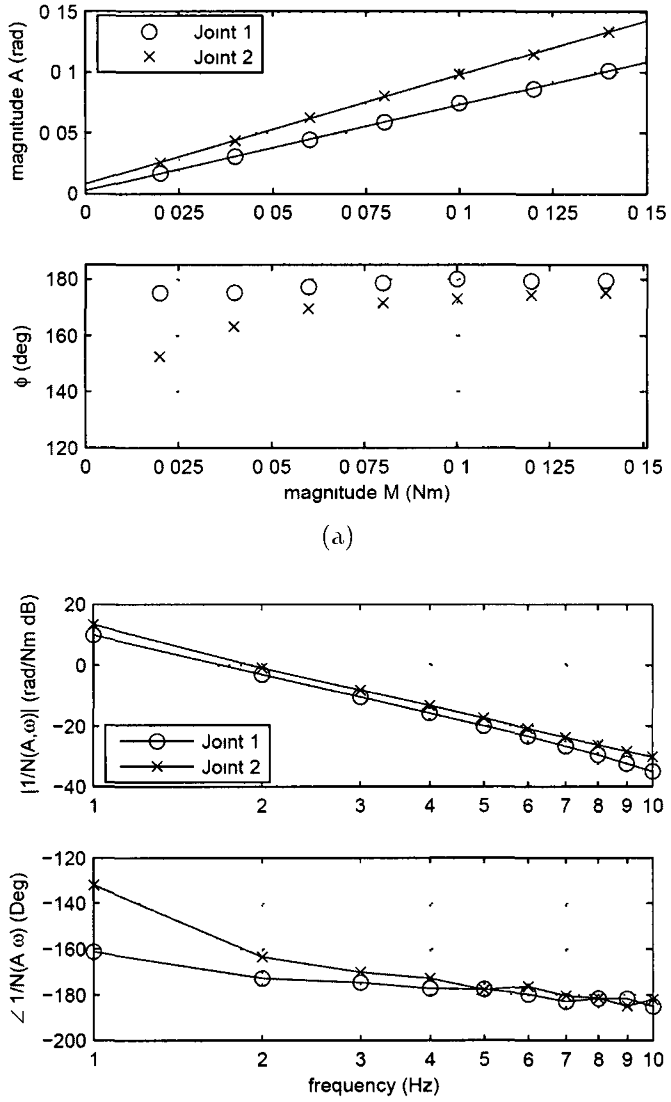
Figure 4.6: A pair of excitation and response signals for Joint 2 of Freedom 6S hand controller at $M = 0.08Nm$ and $f = 2Hz$: (a) a full view of 20 seconds experimental data, and (b) a zoomed view of 2 seconds experimental data.

In our identification for the dynamics of Joint 1 and Joint 2 of the Freedom6S hand controller, we only focused on the low frequency below 10 Hz. Considering that human arm/hand trajectories are mainly within this frequency range, such an approximation will provide useful information about the device's dynamics as the user manipulates the

Table 4.1: Combinations of testing amplitudes and frequencies for Freedom6S hand controller's Joint 1 and Joint 2.

Amplitude M (mNm)	Frequency f (Hz)									
	1	2	3	4	5	6	7	8	9	10
10	×									
20	×	×	×	×	×	×	×	×	×	×
30	×									
40	×	×	×		×	×	×	×	×	×
50	×									
60	×	×	×	×	×	×	×	×	×	×
80		×	×		×					
100		×	×	×	×	×	×	×	×	×
120		×	×		×					
140		×	×	×	×	×	×	×	×	×
160			×		×					×
180			×	×	×	×	×	×	×	×
200					×					×
220				×	×	×	×	×	×	×
240					×					×
260				×	×	×	×	×	×	×
280					×					×
300				×	×	×	×	×	×	×

stylus. To examine the linearity of the joints, we send the sinusoidal excitation signals with different amplitudes at each testing frequency. A typical input driving torque signal and the resulting joint angular signal are plotted in Fig. 4.6. Table 4.1 lists all the testing combinations of the amplitude and the frequency for Joint 1 and Joint 2. Every single experiment lasts 20 seconds so there is enough time to let the transient response die out. At the beginning of every single experiment, the device is always at its equilibrium position. When one joint is under the testing, all other joints of the device are physically locked. We conducted the same experiments on two identical Freedom6S hand controllers. The experimental results generally agreed in terms of the input/output gain, the phase lag, and the frequency of the zero crossing point.



(a)

(b)

Figure 4.7: (a) Amplitude dependence of $N(A, \omega)$ at 2 Hz: point samples are the experimental data and the solid line on the top sub-figure is the linear fit between the input and output amplitude, and (b) Frequency dependence of $1/N(A, \omega)$

Fig. 4.7a shows the system as a function of the input amplitude at 2 Hz. The gain of $N(A, \omega)$ for both Joint 1 and 2 can be well fitted with a straight line but there is a very small shift from the origin due to the uncertainty of the measurements. Meanwhile, the phase lag ϕ increases with the input amplitude. This nonlinearity phenomenon

Table 4.2: Estimated rotational parameters for Joint 1 and 2.

Joint	J $(kg \cdot m^2)$	B_r $(kg \cdot m^2/s)$	σ_r $(N \cdot m)$	K_r $(N \cdot m/rad)$
1	0.0103	0.0102	0.0008	0.3395
2	0.0073	0.0076	0.0014	0.2847

Table 4.3: Estimated translational parameters at the center of workspace.

Direction	m (kg)	b (kg/s)	σ (N)	k_t (N/mm)
y_w	0.1825	0.190	0.035	0.0085
x_w	0.2575	0.255	0.02	0.0072

disappears as the frequency is higher than 5 Hz. The frequency dependence of the system is plotted in Fig. 4.7b where the gain and the corresponding phase lag at each testing frequency are the linear fit and the average of the results at the testing frequency respectively.

Table 4.2 lists the estimated parameters for Joint 1 and 2 using Eq. (4.20). Generally speaking, their values are small, particularly the Coulomb frictions. As mentioned, the movements of Joint 0, 1 and 2 of the Freedom6S controller decide the translational position of the center of the distal stage in the workspace. Therefore, we may convert these joint parameters to their translational counterparts at the center of the distal stage by dividing a square factor of the distance from the center of the distal stage to the respective joint axis. When the device is at its equilibrium position (all joints are at zero positions), this distance is 0.2 m for both Joint 1 and 2. Table 4.3 lists the resulted translational parameters of effective mass (or tip inertia) m, viscous damping b , stiffness k_t , and Coulomb friction σ along y_w (up and down from Joint 2) and x_w (left and right from Joint 1) directions respectively. The effective mass along the x_w direction is smaller because the upper arm is not involved in Joint 2's motion. To further validate the estimated values, we disassemble the arm of the device. The forearm and the end effector weight about 0.150 Kg in total which is very close to our estimation of the effective mass along the y_w direction.

After these physical parameters are estimated, their induced forces can be evaluated. For example, if the center of the distal stage moves along the $+x_w$ direction with acceleration of 1.0 m/s^2 and velocity of 0.2 m/s, the effective mass and viscous damping induced forces will be 0.2575 N and 0.1275 N respectively. In comparison, the Coulomb friction force, being 0.035 N, is quite small. We also neglect the force induced by the inherent stiffness k_t because it is very small at the center of the workspace. Please note that K_r and k_t are different from K_θ and k. The former two variables describe the dynamics of the joint while the latter two denote the gain of the controller for the joint.

4.4 Proposed Methods

Three methods are proposed to compensate for the effective mass and viscous damping induced forces. We start with the direct compensation using the estimated physical parameters at the center of workspace. The second method focuses on the rotational perception by directly moving the point of application of force. In the third method, the stiffness adaptively changes based on the relative position between the collision point and the holding point. All three methods need the real-time measurements of the holding pivot point on the stylus.

4.4.1 Direct compensation

With this method, the compensation force is added to the feedback force/torque F_t from the haptic model directly. The total requested force/torque $F_t^{'}$ is given by

$$\begin{cases} F_t^{'} & = F_t + (ma + bv)\mathbb{I}\{v_c \cdot n\} \\ \tau_t^{'} & = \tau_t \end{cases}, \tag{4.22}$$

where $a \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ denote the linear acceleration and velocity at the center of the distal stage, $m = [m_x \; 0 \; m_z]$,$b = [b_x \; 0 \; b_z]$, $v_c \in \mathbb{R}^3$ and $n \in \mathbb{R}^3$ are the velocity and the normal of the collision point respectively, and $\mathbb{I}\{s\}$ is an indicator function which enables the compensation only when the penetration depth is increasing. However, the Coulomb friction is not considered for two main reasons. The first one is that we are working on the haptic perception of tapping a virtual object where the incoming velocity and acceleration is usually high. Compared with the Coulomb friction, the inertia and viscous force dominates at high velocity and acceleration. The second one is that the device we are using for the experiments has a very small Coulomb friction of 0.035 N.

After the required joint torque to render F_t' is computed through the Jacobian matrix, an independent gravity compensation for Joint 0 is added. In Sec. 4.3, the compensation torque for Joint 0 was measured to be $0.3855\sin(\theta_0)$. But this torque always tries to keep Joint 0 at its current position and it gives the user some sluggish feelings as he/she manipulates the stylus. After some trials, we lowered the compensation torque to $0.35\sin(\theta_0)$ which gave a good balance between the compensation and the sluggish feelings for Joint 0. Since the device is not equipped with accelerometers, the acceleration information is derived from the position measurements. The adaptive 4-state Kalman filter discussed in Chapter 3 is used to estimate the acceleration where human arm/hand trajectory behavior is considered.

4.4.2 Fixed rotation origin compensation

To avoid the tedious estimation process of the physical parameters, we propose the following two methods which directly work on the non-uniform hardness perception. As discussed above, the control point on the stylus is fixed while the force/torque perceived at the user's holding point will be naturally induced. In Case II (Fig. 4.4b), as the user taps the ball, he/she expects a counterclockwise rotation about the holding point, but the actual motor torque for this rotation (Joint 3) will be clockwise due to its relative position around the control point. Ideally, the expected counterclockwise rotational perception will be generated by the translational force f_c. But this rotation perception is not strong enough to counterbalance the torque component τ_c plus the additional term of $(l + h) \times (ma + bv)$.

To let the users clearly perceive the expected rotational movements around the holding point in Case II, we directly set the desired force/torque $F_h$at the control point. This gives $f_c' = f$ and $\tau_c' = -h \times f$. Now, the induced force/torque around the holding point becomes

$$\begin{aligned} f_h' &= f - ma \\ \tau_h' &= -(2h + l) \times f + (h + l) \times (ma + bv). \end{aligned} \tag{4.23}$$

Referring to Fig. 4.4b, this is equivalent to moving the collision point away from the center of the distal stage by h. Now, the torque from Joint 3 will generate a counterclockwise rotation around the holding point as expected. With this method, we do not

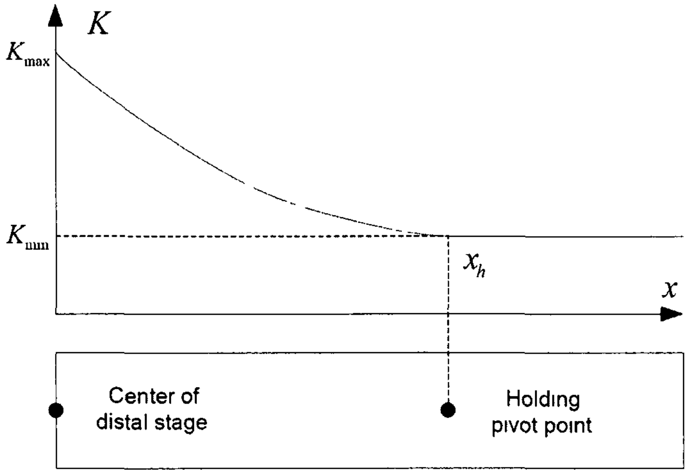


Figure 4.8: Variable stiffness compensation scheme.

modify F_c for Case I. Now users will have a stiffer perception of the virtual object in both cases.

4.4.3 Variable stiffness compensation

In Case I, the user has a harder perception of the simulated object. The further the collision point away from the holding point, the stiffer the perception is. In the third method, instead of assigning a uniform stiffness to the virtual object, we adaptively change the contact linear stiffness based on the locations of the holding point and the collision point as illustrated by Fig. 4.8. Once the collision point is on the other side of the holding point, a uniform stiffness is assigned. The mathematical model for the adaptive stiffness is given by

$$K_x = \begin{cases} K_{\min} + (K_{\max} - K_{\min}\frac{(x-x_h)^2}{x_h^2}) & x \leq x_h; \\ K_{\min} & x > x_h, \end{cases} \tag{4.24}$$

where $K_{\max}$ and $K_{\min}$ are predefined maximum and minimum stiffness. The ratio between $K_{\max}$ and $K_{\min}$ is experimentally determined at each stiffness level to have the best results. For the stiffness levels of 0.1, 0.2 and 0.3 N/m, $K_{\max}/K_{\min} = 2.0$ while for 0.4, 0.5, and 0.6 N/m, $K_{\max}/K_{\min} = 2.25$. This is different from the last method where the direction of the motor joint torque is changed. With this method, the expected rotation is generated through a large translational force at the center of the distal stage.

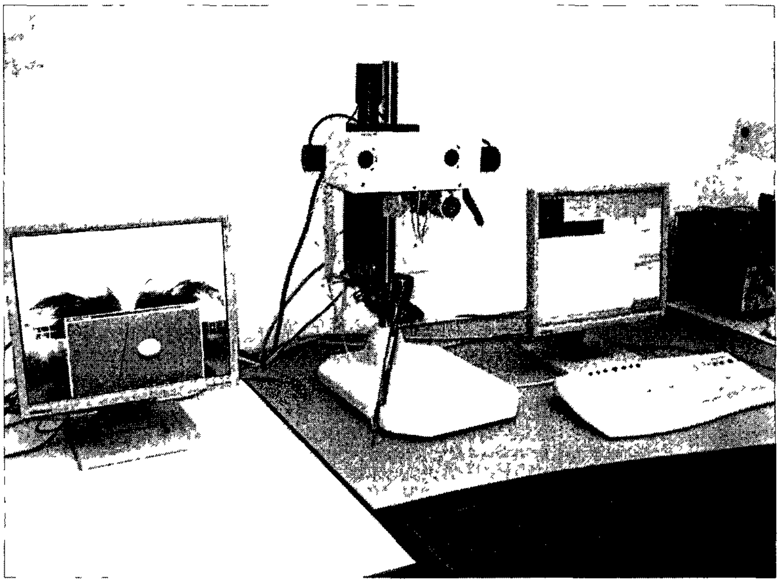


Figure 4.9: Experiment setup for the subjective evaluation.

4.5 Experimental Results and Discussion

4.5.1 Experimental setup and procedure

The experimental setup consisted of a Freedom6S hand controller interfaced with a PC running MS Windows XP. The PC was equipped with an Intel(R) Xeon 1700 MHz processor and 2 GB of RAM. Two 17-inch LCD monitors, sitting side-by-side with the device in between, were configured in a dual-mode as shown in Fig. 4.9. A membrane potentiometer is glued on the end effector of the device. By pressing down on the various parts of the potentiometer, its resistance changes linearly which allows us to accurately measure the position of the holding point. The activation force of the chosen potentiometer is in the range of 0.5-1.8 N, so it can be easily actuated with human fingers without causing distractions from the application itself.

During the experiments, the subject and the operator sat in front of the left and right monitors respectively. The application GUI was spanned into two monitors. Test subjects could only see the graphic rendering of the simulation but not the control panel. Accordingly, they were never informed of the current testing stiffness level or the current compensation method. The control panel of the application, intended for the operator, was displayed on the right monitor. Through the control panel, the operator could modify the testing stiffness, switch among the proposed compensation methods, enable or disable the real-time measurement of the holding point, *etc.* Subjects were instructed to hold the stylus using the thumb and the index fingers, forming a pinch grasp on the end effector. The measured holding point was displayed as three-orthogonal axes in the simulation, so each test subject knew the relative position between the collision point and his/her holding point. To protect the motors, subjects were asked to leave the collision state in case of abnormal vibrations.

Altogether, six stiffness levels ranging from 0.1 to 0.6 N/mm at a step of 0.1 N/mm were tested. The upper limit of the testing stiffness was set such that there are no apparent vibrations as the collision point is far away from the center of the distal stage as discussed in Sec. 4.2. The presentation order of the six stiffness levels to each subject was randomized. At each level, the three proposed compensation methods and the non-compensated one were also randomly presented. Therefore, there are total $4 \times 6 = 24$ tests for each subject. Referring to Fig. 4.4, subjects tapped the stationary ball in two ways: the collision point is between the holding point and the free end of the stylus (Case I), and the collision point is between the holding point and the center of the distal stage (Case II). They were also told to tap the ball with different incoming velocities and accelerations which were recorded for each trial. There is no strict time limit for each test. The subjects were allowed to ask the operator to switch back to the previous method. The velocity and acceleration profiles at the collision point and the center of the distal stage were recorded for each contact transition. At the end of each test, the subjects were asked to select one from the following five options:

A - Case II feels clearly harder than Case I
B - Case II feels slightly harder than Case I
C - Case II feels the same as Case I
D - Case II feels slightly softer than Case I
E - Case II feels clearly softer than Case I

The subjects were also free to add any comments for each trial, such as comparison to the last test, stability, *etc.*

4.5.2 Results and discussion

A total of 10 subjects, all right-handed with no known sensorimotor impairments with their hand or arm, participated in the user study. Three of the subjects use a haptic interface on a regular basis but only one of them had used the Freedom6S. The remaining

Table 4.4: Evaluation results with 10 subjects.

Rating	None	Method 1	Method 2	Method 3
A	0	0	0	0
B	0	2	2	0
C	3	30	20	28
D	9	24	19	25
E	48	4	19	7

seven subjects had no prior experience with haptic devices. Based on the recorded velocity and acceleration profiles, 95% of the incoming velocities and accelerations were in the range of 0.1-0.45 m/s and 0.5-5.4 m/s^2 (absolute value) respectively. Occasionally, the contact transition had a high velocity but a small acceleration, or vice versa.

Table 4.4 lists the number of the test subjects rated for each compensation method in all six stiffness levels. Since there were a total of 10 subjects and six stiffness levels, each column adds up to 60 (6 level × 10 subjects). In general, all three proposed methods perform better than the non-compensated rendering. None of the subjects found Case II to be harder than that in Case I in any trial. For the non-compensated rendering, there were 48 out of 60 who reported Case I to be clearly harder than Case II. This non-uniform hardness perception is even clearer with higher incoming velocities and accelerations. In comparison, only four subjects rated Case I to be clearly harder than Case II with the first method, and seven with the third method. Considering that the subjects had no knowledge of the method being tested, these evaluation results showed again the need for compensation. At the level of 0.1 N/mm, three subjects reported no obvious perceptual differences between Case I and Case II for the non-compensated rendering. This is a sign of increased JND as the assigned stiffness gets lower. When the collision point was more than 75 mm away from the center of the distal stage and the testing stiffness was over 0.4 N/mm, some motor vibrations could be detected. It is possible that part of the stiffer perception for Case I comes from these vibration signals [36] [115]. With the second method, there were still 19 times that the subjects reported clearly different hardness perception in two cases. Thus, the first and third compensation methods perform better than the second one.

To further investigate the results, two-factor analysis of variance (ANOVA) tests were

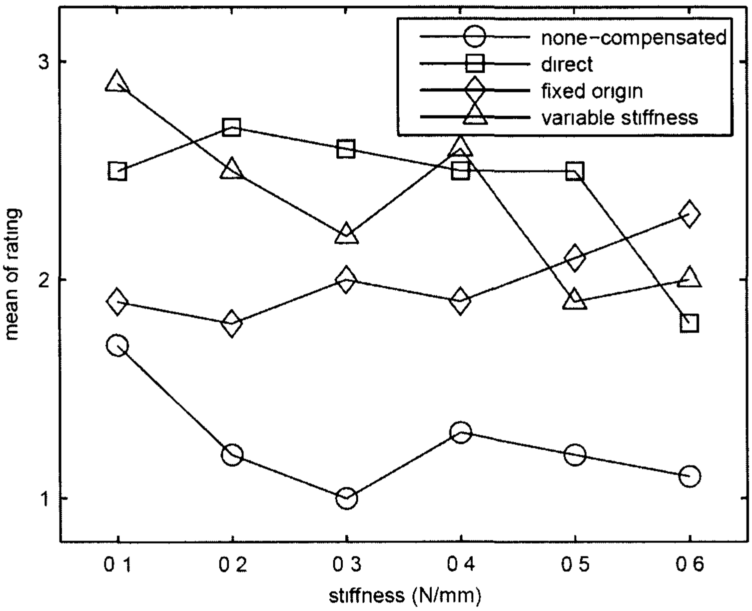


Figure 4.10: Mean values of rating as a function of stiffness levels and compensation methods

applied to the performance of the three proposed methods at all six stiffness levels. We assigned $A = E = 1$ (worst), $B = D = 2$ (acceptable), and $C = 3$ (best). Fig. 4.10 shows the mean values of rating for the different methods at six stiffness levels. It is evident that the non-compensated rendering performed worse than the proposed methods at all six stiffness levels. The ANOVA test confirms that there are significant differences among the none-compensated and the three compensation methods ($F(3, 216) = 41.6$, $p = 0$). If we choose 0.05 as the significant p-value to reject the null hypothesis, the effect of the stiffness level is also significant ($F(5, 216) = 2.26$, $p = 0.0499$), which indicates that the rating decreases a bit as the stiffness level increases. This seems untrue for the fixed origin method where the mean value increases as the stiffness level increases. The significant interaction between the stiffness level and the compensation method ($F(15, 216) = 1.95$, $p = 0.0197$) indicates that the effect of the stiffness level differs depending on the method of compensation. Specifically, it means that increasing the stiffness degrades the performance on some methods and improves performance on other methods. We believe it is also caused by the fixed origin method. To prove this, we did another two-factor ANOVA test but not including the results from the fixed origin method this time. Indeed, the interaction between the two factors is no longer significant ($F(10, 162) = 1.76$, $p = 0.0728$) and the effect of stiffness level also becomes much more significant ($F(5, 162) = 6.07$, $p = 0$). In addition, the fixed origin method gets the largest variance of 0.678 as compared to 0.292, 0.385, and 0.468 for the none-compensated method, the direct method, and the variable stiffness method respectively.

We used separate t-tests between different pairs of compensation methods. A right-tailed t-test shows a significant difference between the direct method and the fixed origin method ($t(118) = 3.255$, $p = 7.395\text{e-}4$). This confirms the mean value of the direct method is significantly higher than that of the fixed origin method. However, a two-tailed t-test does not show a significant difference between the direct method and the variable stiffness method ($t(118) = 0.699$, $p = 0.486$). In the range of 0.1-0.4 N/mm, the direct method performed quite well in terms of the mean and standard deviation, but it degraded a bit at 0.5 and 0.6 N/mm, which implies that the induced forces were under-compensated. Two separate two-tailed t-tests at 0.5 and 0.6 N/mm show no significant difference between the direct method and the fixed origin method ($t(18) = 1.238$, $p = 0.232$ for 0.5 N/mm, and $t(18) = 1.387$, $p = 0.182$ for 0.6 N/mm). We believe that the main cause for this was the inaccuracy of the estimated parameters. We estimated the effective mass and viscous damping at low frequencies (≤ 10 Hz) while collisions with stiffer objects could contain signals with higher frequencies. In addition, since there were no accelerometers installed on the device, the acceleration information had to be derived from the position measurements, which is very challenging as itself. For the first method, all subjects also reported a crisper hard contact perception. This was indeed as expected due to the active compensation of the effective mass and viscous damping during the initial contact period. In [37], Kuchenbecker *et al.* have shown that the contact realism can be improved by exploiting the incoming velocity and acceleration.

For the fixed rotation origin method, the original idea was to enhance the rotational feedback around the holding point for Case II where subjects felt softer contact. However, the method did not perform well at 0.1 N/mm, which was almost the same as the non-compensated one. This is confirmed by a separate two-tailed t-test at 0.1 N/mm between the two methods ($t(18) = 0.490$, $p = 0.63$). At other stiffness levels, both median and mean values indicate Case II is felt slightly softer than Case I, yet the standard deviations are still high. Two subjects totally disfavored this method by rating E at all six testing levels. When resting the stick on the ball or tapping the ball gently, eight subjects reported vibrations in Case II. The vibrations were caused by the mismatch between the force component and the modified torque component as described by Eq. (4.23). There was also a glitch due to the jump in the requested torque as the collision point passes from one side of the holding point to the other. On the other hand, the method performed better as testing stiffness increased. At 0.6 N/mm, it performs as well as the other two compensation methods, confirmed by two separate two-tailed t-tests in between ($t(18) = 1.387$, $p = 0.182$ with the direct method, and $t(18) = 0.896$, $p = 0.382$ with the variable stiffness method).

For the variable stiffness method, a right-tailed t-test shows that its mean is significantly higher than that of the fixed origin method ($t(118) = 2.532$, $p = 0.006$), which indicates a better performance. As aforementioned, there is no significant difference between the variable stiffness method and the direct method. But for the same testing stiffness level, the ball was felt to be softer when compared with the first method. This was also as expected, because the contact linear stiffness decreased as the collision point moved away from the center of the distal stage. Referring to Fig. 4.8, since the stiffness was smoothly changed, there was no glitch felt as the collision point passed the holding point.

4.6 Summary

In this chapter, we studied the influence of a haptic interface's effective mass and viscous damping on the hardness perception of the virtual object in 6-DOF haptic rendering. The device used for the experiments has a pen-like stylus with one end connected to the active end of the device while the other end has nothing to connect to. It was found that distinct hardness can be perceived as a user taps the virtual object with different parts of the tool. Theoretical analysis showed that perceived hardness is related to the locations of the collision point, the user's holding point, and the control point of the haptic interface in 6-DOF haptic rendering.

Assuming a pinch grasp handling posture, we have glued a potentiometer on the stylus of a Freedom6S controller to measure a user's holding point. Three software methods were proposed to compensate for the non-uniform hardness perception. The user study included 10 subjects at six different stiffness levels in the range of 0.1-0.6 N/mm. In general, the first method and the third method perform better than the second one. The first method directly compensates the effective mass and viscous damping induced forces at the active end of the device. It not only improves uniform hardness perception of the contact but also give users a crisp perception during the contact transition period. One drawback of the method is the tedious estimation procedures for those physical parameters. With the second method, the point of the force application is explicitly moved so the rotation around the user's holding point can be clearly perceived. However, there is a clear glitch because of the non-continuity of the requested torque as the collision point moves from one side of the holding point to the other side. With the third method, the simulation stiffness at the collision point varies adaptively based on its relative position to the holding point. Its performance is between the first and the second method.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we studied two issues in haptics: velocity and acceleration estimation for haptic interfaces, and uniform stiffness perception in 6-DOF tool-based haptic rendering. In Chapter 3, we presented a detailed literature review and discussion about the velocity and acceleration estimation from discrete position measurements. These studies showed that to get robust estimates with a high accuracy the filter should be adapted to the characteristics of the trajectory. For that, we reviewed the research results from psychophysical studies on the behaviors of human arm/hand trajectory. These research results show that the human arm/hand trajectory follows a constrained minimum-jerk movement. Hence, we applied this plausible result back to the real engineering problem and proposed a jerk constrained 4-state Kalman filter for the velocity and the acceleration estimation of haptic interfaces. The four states include position, velocity, acceleration, and jerk of the trajectory to be estimated. To honor the characteristic of constrained minimum-jerk movement, the jerk component is set to be varying in a predefine range. If the current distance between the estimated and the measured ones is less than the resolution of the device, the jerk component will be decreased by a predefined step. Otherwise, the jerk component will be increased by a predefine step to catch the transient behavior of the trajectory. In addition, since it is the user that holds the end effector of the device in haptic application, we suggested that the linear velocity and acceleration of the end effector should be estimated in Cartesian space directly instead of converting from the joint space using the device's Jacobian matrix. We also modified the proposed jerk constrained 4-State Kalman filter into a prediction filter to compensate for the network latencies for haptic-enabled telementoring applications.

In the thesis, we discussed the differences between the point-based 3-DOF and the tool-based 6-DOF haptic rendering. We have identified a new problem in 6-DOF haptic rendering: a surface that should feel uniformly stiff will feel softer or harder, depending on which part of the virtual tool it contacts. Essentially, it is caused by the induced forces from the physical parameters of the device such as inertia, viscous damping, Coulomb friction, *etc.* Interestingly, the point-based haptic rendering is not subject to this non-uniform stiffness perception problem. There, the collision point (point of force application) is fixed and the induced force has the same effects, hence the user still gets uniform stiffness perception. However, in the tool-based 6-DOF haptic rendering, the virtual tool takes its real geometric shape in the haptic model and collision points can be anywhere on the tool. In this case, the induced force has different effects on the perceived stiffness which depends on the location of the collision point and its relative location to the user's holding point. We proposed three compensation methods for the problem. The first method directly compensates the effective mass and viscous damping induced forces at the active end of the device. It not only improves uniform hardness perception of the contact but also gives users a crisp perception during the contact transition period. With the second method, the point of the force application is moved so the rotation around the user's holding point can be explicitly perceived. With the third method, the simulation stiffness at the collision point varies adaptively based on its relative position to the holding point. The subjective evaluation results show that the direct compensation method performs the best, following by the third and the second method. Nevertheless, one drawback of the direct compensation method is the tedious estimation procedures for those physical parameters.

5.2 Future Work

With current pervasive communication networks, more and more networked haptic VR applications are being developed. For distributed VR applications, a main challenge is the synchronization of graphical models among the participants due to network delay, latency, packet loss, *etc.* If the haptic is introduced, it becomes even harder because of the additional stability requirements at each participating peer [116]. Research on these networked haptic applications has been conducted in three directions. First, some researches are focusing on the design of new communication protocols in which haptic data are treated as a new medium type and its time critical requirements and relationships with other media are emphasized [13] [117]. Second, some researchers explicitly include users in the design loop at the application level. For example, the current network latency level is visually presented by changing the colors of the interacting virtual objects so the users can adjust their actions accordingly, *e.g.*, slow down the movements in case of a high network delay [118]. The third direction is to apply research results from bilateral teleoperations where the network latencies and packet losses are critical for the performance of the whole system [119] [120] [121]. For these networked haptic applications, our human arm/hand trajectory behavior based velocity and acceleration estimation method can be used for the prediction of the remote user's haptic data which may moderate the effects of the network delays. In the DISCOVER Lab, we have developed a software framework for collaborative hapto-visual applications [19] [105]. This framework has also been used for the evaluation of different haptic data reduction algorithms for haptc-enabled telementoring applications [40] [122]. In the future, we will experimentally evaluate our proposed velocity and acceleration estimation method in such application scenarios.

As pointed out in the chapter, the non-uniform hardness perception can be understood as been caused by the discrepancy between the user's holding point and the active end of the device. Ideally, if we have a 6-DOF haptic interface with such capability, the induced forces from the physical parameters at the active end would go through the holding point and generate no additional torque. But practically it is difficult to do so since the user's handling postures can be diversified and time varying. One possible solution is to design a haptic interface such that both ends of the end effector are actuated such as Quanser 5-DOF haptic wand from Quanser Consulting Inc. As a result, the effects of induced forces from the active ends are symmetrical to the holding point. During the evaluation of the proposed compensation methods, we have also noticed that the visual display of the holding point on the virtual tool provides useful information about the expected rotational movement of the tool. In addition, the visual information of the penetration depth between the tool and the interacting object can also be controlled to change the user's hardness perception as learned by other researchers [79] [123]. More investigations on these topics are left as future work.

Appendix A

Velocity Estimation with Least Square Method

In [87], Janabi-Sharif *et al.* proposes a adaptive window method to estimation the velocity of haptic interfaces using least square method. Here is the derivation for the analytical formula given in the paper.

Let us assume the velocity is constant in a very short period. We then want to find the equation of a straight line $y = \alpha t + \beta$ which passes through all samples measured during this period. α is exactly the velocity to solve. Let's put it in a matrix form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{A.1}$$

where $\mathbf{x} = [\alpha\ \beta]^T$ is the unknowns and $\mathbf{b} = [y_k\ \cdots\ y_{k-i}\ \cdots\ y_{k-n}]^T$ is the measured samples. Data matrix $\mathbf{A}$ is formed as

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -T & 1 \\ \vdots & \vdots \\ -iT & 1 \\ \vdots & \vdots \\ -nT & 1 \end{pmatrix}, \tag{A.2}$$

where T is the sampling interval and n is the number of previous samples used for the current estimation. The estimated solution of $\mathbf{x}$ in the least square sense is

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{b}). \tag{A.3}$$

Therefore,

$$\begin{aligned}
\mathbf{A}^T\mathbf{A} &= \begin{pmatrix} 0 & -T & \cdots & iT & \cdots & -nT \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -T & 1 \\ \vdots & \vdots \\ -iT & 1 \\ \vdots & \vdots \\ -nT & 1 \end{pmatrix} \\
&= \begin{pmatrix} T^2(1^2+2^2+\cdots+n^2) & -T(1+2+\cdots+n) \\ -T(1+2+\cdots+n) & n+1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{T^2}{6}n(n+1)(2n+1) & -T\frac{n(n+1)}{2} \\ -T\frac{n(n+1)}{2} & n+1 \end{pmatrix}
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
|\mathbf{A}^T\mathbf{A}| &= \frac{T^2}{6}n(n+1)^2(2n+1) - \frac{T^2}{4}n^2(n+1)^2 \\
&= \frac{T^2}{6}n(n+1)^2(2n+1-\frac{3}{2}n) \\
&= \frac{T^2}{6}n(n+1)^2(\frac{n}{2}+1) \\
&= \frac{T^2}{12}n(n+1)^2(n+2)
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
(\mathbf{A}^T\mathbf{A})^{-1} &= \frac{1}{|\mathbf{A}^T\mathbf{A}|} \begin{pmatrix} n+1 & T\frac{n(n+1)}{2} \\ T\frac{n(n+1)}{2} & \frac{T^2}{6}n(n+1)(2n+1) \end{pmatrix} \\
&= \frac{1}{\frac{T^2}{12}n(n+1)^2(n+2)} \begin{pmatrix} n+1 & T\frac{n(n+1)}{2} \\ T\frac{n(n+1)}{2} & \frac{T^2}{6}n(n+1)(2n+1) \end{pmatrix} \\
&= \begin{pmatrix} \frac{2}{\frac{T^2}{6}n(n+1)(n+2)} & \frac{n}{\frac{T}{6}n(n+1)(n+2)} \\ E & F \end{pmatrix},
\end{aligned} \tag{A.6}$$

where E and F are not computed since they do not contribute to the final velocity estimate. And

$$\mathbf{A^T b} = \begin{pmatrix} 0 & -T & \cdots & iT & \cdots & -nT \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_k \\ \vdots \\ y_{k-i} \\ \vdots \\ y_{k-n} \end{pmatrix} = \begin{pmatrix} -T\sum_{i=0}^{n} i y_{k-i} \\ \\ \sum_{i=0}^{n} y_{k-i} \end{pmatrix} \tag{A.7}$$

Accordingly,

$$\begin{aligned} \hat{\mathbf{x}} &= (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{b}) \\ &= \begin{pmatrix} \frac{2}{\frac{T^2}{6}n(n+1)(n+2)} & \frac{n}{\frac{T}{6}n(n+1)(n+2)} \\ \\ E & F \end{pmatrix} \begin{pmatrix} -T\sum_{i=0}^{n} i y_{k-i} \\ \\ \sum_{i=0}^{n} y_{k-i} \end{pmatrix} \\ &= \begin{pmatrix} \frac{n\sum_{i=0}^{n} y_{k-i} - 2\sum_{i=0}^{n} i y_{k-i}}{Tn(n+1)(n+2)/6} \\ \\ \text{not useful here} \end{pmatrix} \end{aligned} \tag{A.8}$$

Let $n = M - 1$, where M is the filter window length, the estimated velocity in least square sense can then be computed as

$$\hat{\alpha} = \frac{12\sum_{i=0}^{M-1}(\frac{M-1}{2} - i)y_{k-i}}{TM(M-1)(M+1)}. \tag{A.9}$$

Appendix B

C Source Code for a 4-state Kalman Filtering

```
/*-----------------------------------------------------------------------------
 *  Name:  Thornton_UD_propagation
 *  The detailed explanation and its Matlab code can be found in [104].
 *
 *  Note: C is the output matrix with a form [1 0 0 0]
 *
 *  INPUT:   X_update     state estimate X(k|k)
 *           U_update     Upper unit triangular matrix with D(i) stored on
 *                        the diagonals. this corresponds to the a priori
 *                        covariance P(k|k)
 *           A            state transition matrix
 *           Q            process noise covariance matrix
 *  OUTPUT:  X_a_priori   Predicted state estimate X(k+1|k)
 *           U_a_priori   Predicted upper unit triangular matrix with
 *                        D(i)stored on the diagonals; this corresponds
 *                        to the a priori covariance P(k+1|k)
 *----------------------------------------------------------------------------*/
void Thornton_UD_propagation(float X_update[4], float U_update[4][4],
   float A[4][4], float Q[4][4],
   float X_a_priori[4], float U_a_priori[4][4])
{
   int order = 4;
   float u[4][4] = {{1.0f,0.0f,0.0f,0.0f}, {0.0f,1.0f,0.0f,0.0f},
                    {0.0f,0.0f,1.0f,0.0f},{0.0f,0.0f,0.0f,1.0f}};
   float G[4][4] = {{0.0f,0.0f,0.0f,0.0f}, {0.0f,0.0f,0.0f,0.0f},
                    {0.0f,0.0f,0.0f,0.0f},{0.0f,0.0f,0.0f,1.0f}};
```

```
float D_in[4] = {0.0f,0.0f,0.0f,0.0f};
float D_out[4] = {1.0f,0.0f,0.0f,0.0f};
float sigma;
int i,j,k;

D_in[0] = U_update[0][0];
D_in[1] = U_update[1][1];
D_in[2] = U_update[2][2];
D_in[3] = U_update[3][3];
U_update[0][0] = 1.0f;
U_update[1][1] = 1.0f;
U_update[2][2] = 1.0f;
U_update[3][3] = 1.0f;

// x propagation x_a_priori = a * x_update
for (i = 0; i < order; i++)
{
    X_a_priori[i] = X_update[i];
    for (j = i+1; j<order; j++)
        X_a_priori[i] += A[i][j] * X_update[j];
}

u[0][1] = U_update[0][1] + A[0][1];
u[0][2] = U_update[0][2] + A[0][2] + A[0][1]*U_update[1][2];
u[0][3] = U_update[0][3] + A[0][1]*U_update[1][3] +
                A[0][2]*U_update[2][3] + A[0][3];
u[1][2] = U_update[1][2] + A[1][2];
u[1][3] = U_update[1][3] + A[1][2]*U_update[2][3] + A[1][3];
u[2][3] = U_update[2][3] + A[2][3];

for (i = order-1; i >= 0; i--)
{
    sigma = 0;
    for (j = 0; j<order; j++)
    {
        sigma = sigma + u[i][j]*D_in[j]*u[i][j];
    }
    for (j = 0; j<order; j++)
    {
        sigma = sigma + G[i][j]*Q[j][j]*G[i][j];
    }
    D_out[i] = sigma;
```

```
        for (j = 0; j <= i-1; j++ )
        {
            sigma = 0;
            for (k = 0; k < order; k++)
                sigma = sigma + u[i][k]*D_in[k]*u[j][k];

            for (k = 0; k < order; k++)
                sigma = sigma + G[i][k]*Q[k][k]*G[j][k];

            U_a_priori[j][i] = sigma/D_out[i];

            for (k = 0; k < order; k++)
                u[j][k] = u[j][k] - U_a_priori[j][i]*u[i][k];

            for (k = 0; k < order; k++)
                G[j][k] = G[j][k] - U_a_priori[j][i]*G[i][k];
        }
    }

    U_a_priori[0][0] = D_out[0];
    U_a_priori[1][1] = D_out[1];
    U_a_priori[2][2] = D_out[2];
    U_a_priori[3][3] = D_out[3];
}
```

```
/*----------------------------------------------------------------------
 *  Name: Bierman_UD_update
 *  The detailed explanation and its Matlab code can be found in [104].
 *
 *  Note: C is the output matrix with a form [1 0 0 0]
 *
 *  INPUT:   X_a_priori  A priori state estimate X(k|k-1)
 *           U_a_priori  Upper unit triangular matrix with D(i) stored on
 *                       the diagonals; this corresponds to the a priori
 *                       covariance P(k|k-1)
 *           y           the observation y(k) (the new position measurement)
 *           r           the observation error variance
 *
 *  OUTPUT:  X_update    Updated state estimate X(k|k)
 *           U_update    Updated upper unit triangular matrix with D(i)
 *                       stored on the diagonals; this corresponds to
 *                       the a priori covariance P(k|k)
 *           K           Kalman  gain K(k)
 *---------------------------------------------------------------------*/

void Bierman_UD_update(float X_a_priori[4], float U_a_priori[4][4],
    float y, float r,
    float X_update[4], float U_update[4][4], float K[4])
{
    int order = 4;
    float dz;
    float a[4], b[4], alpha, beta, lambda, gamma, dzs;
    int i, j;

    float D[4]; // diagonal elements
    D[0] = U_a_priori[0][0];
    D[1] = U_a_priori[1][1];
    D[2] = U_a_priori[2][2];
    D[3] = U_a_priori[3][3];
    U_a_priori[0][0] = 1.0f;
    U_a_priori[1][1] = 1.0f;
    U_a_priori[2][2] = 1.0f;
    U_a_priori[3][3] = 1.0f;

    memcpy(U_update, U_a_priori, 16*sizeof(float));

    a[0] = U_a_priori[0][0];
```

```
    a[1] = U_a_priori[0][1];
    a[2] = U_a_priori[0][2];
    a[3] = U_a_priori[0][3];
    b[0] = D[0]*a[0];     // b = D_a_priori*U_a_priori'*H'
    b[1] = D[1]*a[1];
    b[2] = D[2]*a[2];
    b[3] = D[3]*a[3];

    dz = y - X_a_priori[0];
    alpha = r;      //alpha(-1) = r;
    gamma = 1/alpha;
    for (j = 0; j < order; j++)
    {
        beta = alpha; //alpha_(j-1)

        alpha = alpha + a[j]*b[j]; //alpha(j)

        lambda = -a[j]*gamma; //lambda =- a(j)/alpha(j-1)

        gamma = 1/alpha;

        //D_update = D_a_priori*alpha(j-1)/alpha(j)
        D[j] = beta*gamma*D[j];
        for (i = 0; i <= j-1; i++)
        {
            beta = U_update[i][j];
            U_update[i][j] = beta + b[i]*lambda;
            b[i] = b[i]+b[j]*beta;
        }
    }
    dzs = gamma*dz;

    for (i = 0; i < order; i++)
        X_update[i] = X_a_priori[i] + b[i]*dzs;

    U_update[0][0] = D[0];
    U_update[1][1] = D[1];
    U_update[2][2] = D[2];
    U_update[3][3] = D[3];
}
```

```
/*------------------------------------------------------------------------
 *  Name: A_update
 *
 *  Function: This function computes the new matrix A for the new T
 *----------------------------------------------------------------------*/
void A_update(float A_current[4][4], float T)
{
   A_current[0][0] = 1.0;
   A_current[0][1] = T;
   A_current[0][2] = T*T/2.0f;
   A_current[0][3] = T*T*T/6.0f;
   A_current[1][0] = 0.0;
   A_current[1][1] = 1.0;
   A_current[1][2] = T;
   A_current[1][3] = T*T/2.0f;
   A_current[2][0] = 0.0;
   A_current[2][1] = 0.0;
   A_current[2][2] = 1.0;
   A_current[2][3] = T;
   A_current[3][0] = 0.0;
   A_current[3][1] = 0.0;
   A_current[3][2] = 0.0;
   A_current[3][3] = 1.0;
}

/*------------------------------------------------------------------------
 *  Name: Q_update
 *
 *  Function: This function computes the new matrix Q for the new T
 *----------------------------------------------------------------------*/
void Q_update(float Q_current[4][4], float T, float q)
{
   Q_current[0][0] = q*T*T*T*T*T*T*T/252.0f;
   Q_current[0][1] = 0.0;
   Q_current[0][2] = 0.0;
   Q_current[0][3] = 0.0;
   Q_current[1][0] = 0.0;
   Q_current[1][1] = q*T*T*T*T*T/20.0f;
   Q_current[1][2] = q*T*T*T*T/8.0f;
   Q_current[1][3] = q*T*T*T/6.0f;
   Q_current[2][0] = 0.0;
   Q_current[2][1] = Q_current[1][2];
```

```
  Q_current[2][2] = q*T*T*T/3.0f;
  Q_current[2][3] = q*T*T/2.0f;
  Q_current[3][0] = 0.0;
  Q_current[3][1] = Q_current[1][3];
  Q_current[3][2] = Q_current[2][3];
  Q_current[3][3] = q*T;
}

/*-------------------------------------------------------------------------
 *  Name: UD_factorization
 *  INPUT:  P a definite positive matrix
 *  OUTPUT: U an upper unit triangular matrix with D(i) stored on the
 *          diagonals
 *  Function:  P = U * diag(D(1),..., D(n))*U'
 *------------------------------------------------------------------------*/
void UD_factorization_4(float P[4][4], float U[4][4])
{
  int order = 4;
  int i,j,k;
  float alpha, beta;

  memset(U, 0, 16*sizeof(float));

  for(i = order-1; i > 0; --i)
  {
    U[i][i] = P[i][i];        // D[i] = P[i][i]
    alpha = 1/U[i][i];

    for(j=0; j<i; j++)
    {
      beta = P[j][i];
      U[j][i] = alpha * beta; // U[j][i] = P[j][i]/D[i]

      for (k=0; k<j+1; k++)
      {
        // P[k][j] = P[k][j] - U[k][i]*U[j][i]*D[i]
        P[k][j] -= beta * U[k][i];
      }
    }
  }
  U[0][0] = P[0][0];
}
```

```
/*----------------------------------------------------------------------
 *  Name: Kalman_Filtering_computation
 *
 *  Function:  update of Kalman filtering at each time step
 *---------------------------------------------------------------------*/
void Kalman_Filtering_computation(float X_a_priori[4],
   float U_a_priori[4][4], float y, float r,
   float X_update[4], float U_update[4][4], float Gain[4],
   float T, float q, float A[4][4], float Q[4][4])
{
   Bierman_UD_update(X_a_priori, U_a_priori, y, r, X_update,
                           U_update, Gain);
   Q_update(Q, T, q);
   A_update(A, T);
   Thornton_UD_propagation(X_update, U_update, A, Q, X_a_priori,
                                 U_a_priori);
}
```

Appendix C

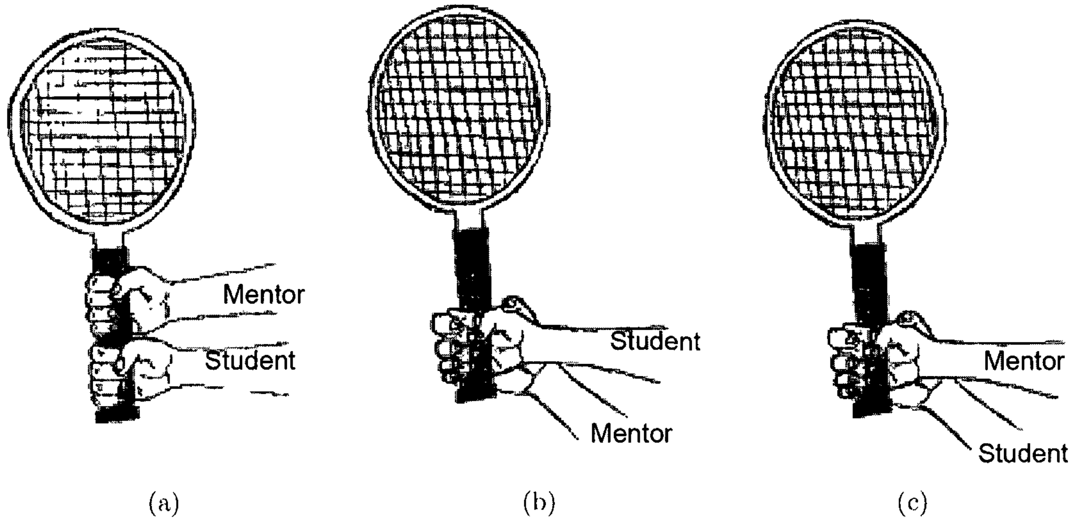
Haptic-Enabled Telementoring

C.1 Benefits of Haptics in Learning of Spatio-temporal Trajectories

There have been some studies on the application of haptic feedback in the learning of spatio-temporal trajectories. Feygin *et al.* call the approach "haptic guidance" [106]. For example, an experienced surgeon's handling trajectory of a scalpel for a cataract surgery can be pre-recorded. Then in the training system, the haptic device can be programmed to playback this pre-recorded trajectory while trainees are able to follow and feel the experienced surgeon's movements by holding the device end effector.

In [107], Gillespie *et al.* design a "virtual teacher" for the purpose of facilitating the acquisition of a manual skill. The design is based on three common paradigms to teach manual skills in real-world as illustrated in Fig. C.1. These three paradigms are distinguished by the arrangements of mechanical contacts between the mentor's hand, the trainee's hand, and the tool.

In the first paradigm, the mentor and the trainee grasp at different locations on the tool. Since there is no direct contact between the mentor and the trainee, the motion of the mentor is felt by the trainee indirectly through the tool. The mentor also feels the combined dynamics of the trainee and the tool. In the second paradigm, the mentor grasps the trainee's hand which in turn grasps the tool. This is a common way to teach the manual skills in the real-world, such as teaching kids how to write English alphabets. Since the trainee experiences two distinct contacts: one with the tool and the other with the mentor, the trainee not only feels the dynamics of the environment through the tool but also follows the correct path due to the guiding force from the mentor. In the third



(a) (b) (c)

Figure C.1: Three teaching paradigms, distinguished by the arrangement of mechanical contact between mentor's hand, trainee's hand, and tool: (a) Indirect Contact Style; (b) Double Contact Style; and, (c) Single Contact Style [107].

paradigm, the trainee grasps the mentor's hand which in turn grasps the tool. Since the trainee indirectly grasps the tool, he/she is able to monitors the motions of the mentor but only can feel the mentor-filtered dynamics between the tool and the environment. In the case of comparable mechanical impedances between the environment and the mentor's hand, the trainee will probably not have correct perception of the contact impedance. However, it is most natural for the mentor to manipulate the tool. The pilot study presented in the paper is designed to test the hypothesis that a virtual teacher could teach a human operator the optimal strategy to perform a crane-moving task. The experimental results show that the virtual teacher is able to effectively demonstrate and encourage the optimal strategy to the trainees although the average task completion time does not improve much. Some subjects report that the virtual teacher has an advantage over the human teacher because of its accuracy and consistency.

In [108], Williams *et al.* evaluate the usefulness of a haptic playback system in the learning of palpation diagnosis. The studied system includes two working modes. In the first mode, the trainee holds the device's stylus passively and is guided by a PD controller which plays back a pre-recorded trajectory. In the second mode, the feedback forces from the virtual environment are enabled and the trainee is instructed to follow the visually displayed pre-recorded trajectory. If the current position error to the pre-recorded trajectory is larger than a predefined threshold, a spring force will pull the trainee back to the correct path. The experimental results suggest that the passive haptic playback is helpful for the learning of some specific movements, but it is not so obvious that the force motor skill can be effectively learned in this manner. They also conclude that the performance with the second mode is enhanced when preceded by the first mode. In other words, the active learning is more effective than the passive learning. Similarly. Basdogan *et al.* program a haptic device to provide controlled forces to trainees along a pre-recorded trajectory tunnel for motor skills' training [124]. Within this tunnel, the trainees should feel the same feedback forces from the virtual environment as the expert feels. As the trainees move out of the tunnel, a correcting force will bring them back. There are two modes in their haptic guidance system as well: time-dependent and time-independent. In the time-dependent mode, a graphical guiding tool is displayed on the screen and moves along the same path and speed as the pre-recorded trajectory. The controlled force will bring the trainee back to the correct trajectory in case of either path or instantaneous speed mismatch. In the time-independent mode, there is no need for the trainees to match the velocity profile of the trajectory, so they can more concentrate on the instantaneous force feedback and the path shape. Indeed, this is very important for the learning of manual skills in the real-world. We do not expect babies to run before they can even walk. Although it is not explicitly mentioned. the experiments were designed with the active learning in mind.

In [15], Morris *et al.* design a haptic training system with particular emphasis on the role of haptic guidance in the learning of a sequence of one-dimensional forces along a position trajectory. The system has three training modes: haptic training only, visual training only. and combined hapto-visual training. The performance measure is the accuracy of the actual recalled force against the pre-recorded force trajectory. As expected, the experimental results show that the hapto-visual training mode significantly outperforms the visual only mode or the haptic alone mode. However, the haptic training alone is significant less effective to visual training alone.

In [125], Corno *et al.* study two closely correlated trajectories accomplishing a force motor skill: movement and accompanying user's exerting force trajectory. Indeed. these two flows form a closed-loop system between the interacting environment and the user in the real-world. The movement trajectory is produced by muscle forces while the interacting forces in turn modify the muscle forces with the user's intention. The authors claim that the most effective way to teach force motor skills is to display both the movement and the force trajectories simultaneously.

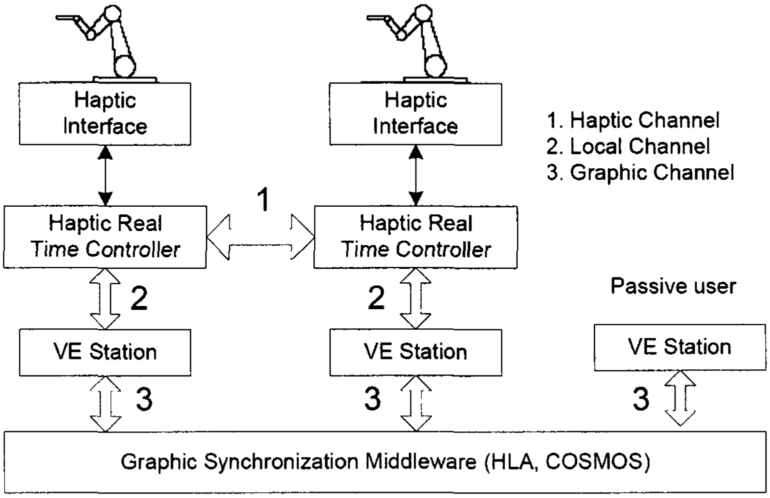


Figure C.2: C-HAVE system architecture.

In summary, in a haptic guidance learning system, trainees are either actively or passively guided through the expert's trajectories by holding the stylus of a haptic interface. This gives the trainees a kinesthetic understanding of what is required for a specific force motor skill. From all these studies on the haptic guidance/playback learning, it supports a possible role for haptics in the training of perceptual force motor skills within virtual environments. We may summarize the results from these studies as follows:

1. Haptic guidance helps in the learning of force sensorimotor skills.
2. Temporal aspects of both position and force trajectories are more effectively mastered through haptic guidance.
3. Visual guidance is essential for the learning of sensorimotor skills, particularly important for the teaching of position trajectory shape.
4. Active haptic learning is more effective than passive haptic learning in terms of making errors and developing strategies to accomplish the task.
5. Performance is usually measured by position, shape, and temporal accuracy.

C.2 Haptic-Enabled Telementoring System Architecture

A minimum haptic-enabled telementoring system at least needs the following components: two computers running VE applications at two sites, two haptic interfaces to control the virtual tool inside the VE application at both sites, and some communication networks linking the two workstations.

Shen *et al.* proposed a generic system architecture for Collaborative Hapto-Audio-Visual Visual Environments(C-HAVE) in [105]. As illustrated in Fig. C.2, a haptic node within C-HAVE comprises three components: a haptic device, a Haptic Real Time Controller (HRTC), and a Virtual Environment (VE) station. The HRTC controls the local haptic device through a device driver. A local HRTC is linked with remote HRTC(s) through a haptic channel and communicates with its local VE station through a local channel. The VE stations are interconnected over HLA/RTI for updates of all shared graphical objects for graphic synchronization or audio/video communications among the distributed users. Since haptic-enabled telementoring is a special application scenario of C-HAVE, we implemented all the functionalities required for the telementoring inside the HRTC so that the same system architecture is still suitable.

In the whole architecture, the HRTC is the key component. Ideally, it should rely on a real-time underlying operating system such as Windows-CE or RT-Linux to have a high resolution timer and scheduling mechanism. The main functionalities of a HRTC are listed in the following:

1. A haptic servo loop thread to read measurements from the local device and set commanded forces to the device. The thread may also include the interpolation method for successive forces from the local VE station or the telementoring controller if the update rate does not match.

2. A communication thread to transmit/receive haptic data to/from the remote HRTC. For such kind of real-time data, User Datagram Protocol (UDP) is the popular choice of transport layer protocols. The thread also includes time delay compensation for the inherent network latency so as to provide synchronized haptic data to the local VE stations or the telementoring controller.

3. A communication thread to transmit/receive data to/from the local VE station. This communication channel is implemented with TCP for reliable data transmission.

4. A process to monitor current network latency and synchronize the timers of distributed HRTCs in every certain amount of time.

5. A module for haptic-enabled telementoring controller.

C.3 Time Delay Effects

In haptic-enabled telementoring, haptic data describing the trajectory, cutaneous or kinesthetics information are packetized and sent over the network at a certain transmission rate. The controllers at both sites keep the two distributed haptic devices following each other in terms of position and velocity. Due to network impairments such as latency and packet loss, the received signal is quite different from the original signal. To show this, let x_m and x_s denote the measured position at the mentor and the trainee side, x_{dm} and x_{ds} be the delayed position, and the data is transmitted at every T_s seconds. At the receiver, the application actively reads the input buffer every T_s seconds. The received signals become

$$\begin{aligned} x_{dm}(t_k) &= x_m(t_k - L_k T_s) I\{V_s(t_k) > 0\} \\ x_{ds}(t_k) &= x_s(t_k - M_k T_s) I\{V_s(t_k) > 0\}, \end{aligned} \tag{C.1}$$

where I is an indicator function defined by

$$I(S) = \begin{cases} 1, & \text{if the statement is true;} \\ 0, & \text{otherwise.} \end{cases} \tag{C.2}$$

and $V(t_k)$ denotes the number of packets coming in during time interval $[t_{k-1}, t_k)$. As shown in Fig. C.3, for the same original signal, $V(t_{k_3}) = 3$, while $V(t_{k_0}) = V(t_{k_1}) = V(t_{k_2}) = 0$, *etc.*

The proportional and derivative gains for the controller are determined according to the device and the test results. To maintain the stability of the closed-loop control, both gains should adapt to the measured network latency. Without compensation, the instant position error used is proportionally increased with the network latency, so we have to decrease K_p to improve the stability phase margin. Intuitively, it is to prevent the output forces from exceeding the motor's limit or exciting the oscillation. The direct effect of network latency is the sluggishness felt by the users. The consequences of this sluggish response are twofold. On the one hand, it explicitly gives mentor the current network latency information and he/she can adjust the movement accordingly. On the

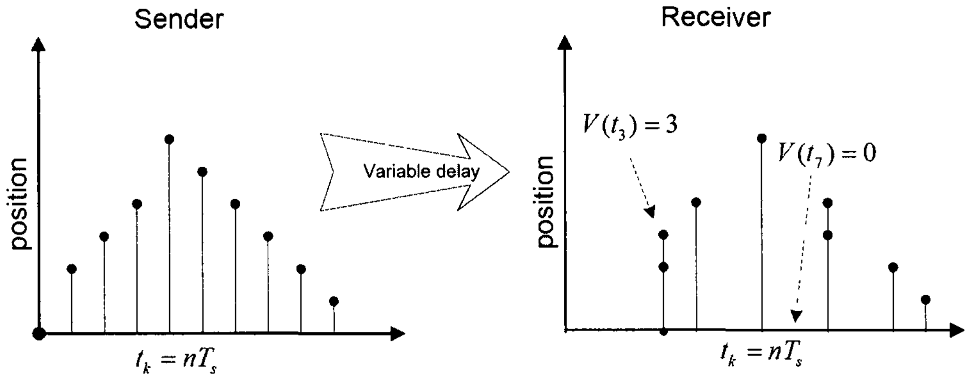
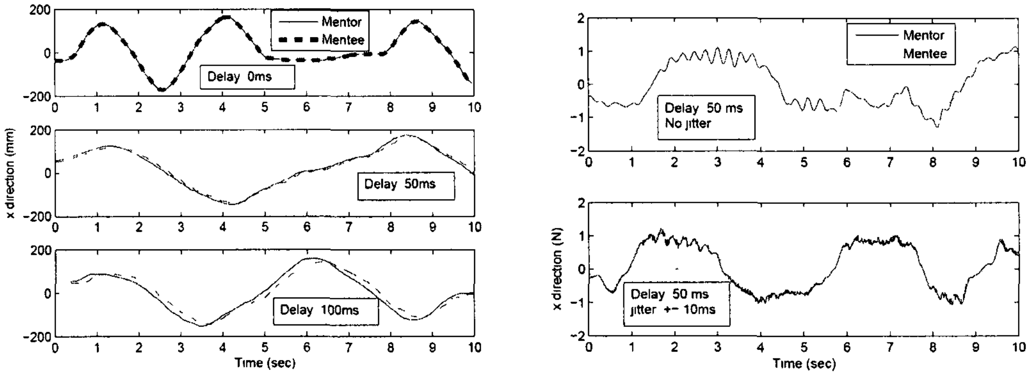


Figure C.3: A distorted haptic signal due to network delays.



(a)

(b)

Figure C.4: Time delay effects on the tele-mentoring controller gain: (a) constant time delay effects and (b) jitter effects.

other hand, this feeling impedes the freedom of the mentor's movement and tires the mentor after a very short time.

Fig. C.4a shows the experimental results of telementoring under constant time delays. To avoid the effect of time synchronization, we connect both haptic devices to the same computer and insert the delay artificially. The experiments assume the trainee acting passively. Fig. C.4b shows the experimental results of a 50 ms mean delay with a 10 ms jitter. The same PD controller gains are applied for the experiments with and without jitter. It is obvious that the output forces with constant delay are much smoother than that of variable delay. The high frequency components in the output of variable delay cause annoying whine noise because of the limited bandwidth of the motor. If the same

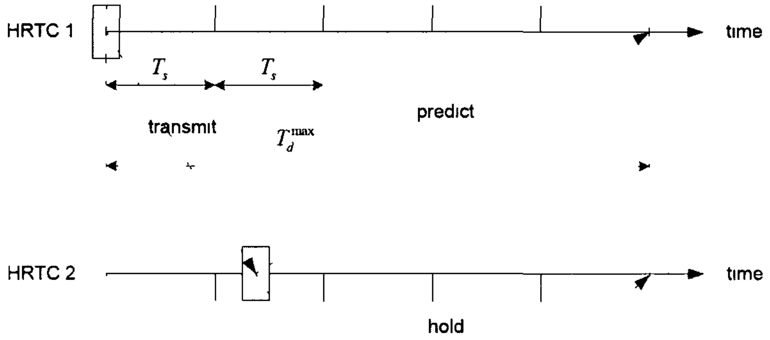


Figure C 5: Timing of events in transmission, processing and prediction. (adapted from [39])

controller gains are applied for a 20 ms jitter, the system will start vibrate and lose the stability.

C.4 Implementation of Time Delay Compensation

In the audio/video streaming, one way to deal with the network latency is to provide a playout buffer at the receiver side Once the buffer depth reaches a predefined playout start threshold, the receiver begins pop the data from the buffer to the user The idea behind this is to convert variable delays into constant delays. However, this playout buffer mechanism introduces additional time delay. For haptic-enabled telementoring, this additional delay may drive the system into unstable region and limit it to very slow motions with sluggish feelings One feasible solution is to convert the variable delay jitter into constant delay with a jitter buffer first and then predict the states of remote device based on the previous received data and the current time delay. If the dynamics of the motion is known or easy to predict, this approach will solve the problem of both inherent network latency and the artificial delay caused by the buffering mechanism. Let us assume samples are sent every T_s seconds and $T_d^{max} = NT_s$ be maximum time delay between two HRTCs as described in the system architecture. This maximum delay T_d^{max} can be preset or determined by on-line measurements. At each transmission instant, a packet containing the motion (position, velocity, acceleration, and jerk for both directions of motion) and the timestamp information is sent between two HRTCs. As studied by Caradima in [39], one of the following situations may occur:

Case I The transmission delay is smaller than $T_d^{\max}$. At the standpoint of the HRTC

2, once a sample from the HRTC 1 is arrived, the state information will be fed into the predictor to predict the states of device at HRTC 1 at time $t_1^1 + T_d^{\max}$. The predicted state will be then held in a playout buffer until the $T_d^{\max}$ has elapsed. At $t_1^1 + T_d^{\max}$, the predicted state of HRTC 1 will be fed into the controller of HRTC 2 as reference for position tracking. The processing sequence is shown in Fig. C.5

Case II: It is possible that due to temporary congestion traffic or communication blackout, the transmission delay is larger than $T_d^{\max}$. A counter m with a maximum value M is preset in the software to control how many time steps without new measurements are tolerated. Let T' be the time delay. If $T' < MT_s + T_d^{\max}$, the last available data will be either used directly or predicted over another m steps. Once $T' \leq M \times T_s + T_d^{\max}$ happens, the HRTC notifies the user about the interruption in communication, cleans all the buffers, and stops the tele-mentoring.

Case III: Occasionally, the packet may get lost since UDP does not guarantee a reliable transmission. For example, there are gaps inside the playout buffer. The same mechanism used in case 2 with a preset value m can solve the problem. For consecutive packets, if they are received before their scheduled playout time, the out-of-order is recovered by the holing buffer. If they are received after the scheduled playout time, they are simply discarded.

Appendix D

Subjective Evaluation Results for Uniform Hardness Perception

D.1 ANOVA Test Results

Table D.1: Two-way ANOVA table with non-compensation results included.

Source	SS	Df	MS	F	Prob. >F
Stiffness	4.733	5	0.9467	2.26	0.0499
Method	52.35	3	17.45	41.6	0
Interaction	12.3	15	0.82	1.95	0.0197
Error	90.6	216	0.4194		
Total	159.983	239			

Table D.2: Two-way ANOVA table without none-compensation results.

Source	SS	Df	MS	F	Prob. >F
Stiffness	9.778	5	1.9556	6.07	0
Method	52.344	2	26.1722	81.22	0
Interaction	5.656	10	0.5656	1.76	0.0728
Error	52.2	162	0.3222		
Total	119.978	179			

Table D.3: One-way ANOVA tests with non-compensation results included.

Stiffness (N/mm)	$F(3,36)$	p
0.1	5.935	0.002
0.2	12.261	1.118e-5
0.3	11.914	1.439e-5
0.4	10.966	2.925e-5
0.5	6.698	0.001
0.6	5.639	0.003

Table D.4: One-way ANOVA tests without non-compensation results.

Stiffness (N/mm)	$F(2,27)$	p
0.1	5.561	0.009
0.2	5.912	0.007
0.3	1.8	0.18
0.4	3.949	0.03
0.5	1.762	0.191
0.6	1.089	0.351

D.2 Subjective Evaluation Results with 10 Subjects

Table D.5: Subjective evaluation results with 10 subjects.

Method	Stiffness (N/mm)						Method	Stiffness (N/mm)					
	0.1	0.2	0.3	0.4	0.5	0.6		0.1	0.2	0.3	0.4	0.5	0.6
None	D	E	E	E	D	E	1	C	D	C	D	D	D
	C	C	E	E	D	D		D	C	D	C	D	C
	D	D	E	E	E	E		C	C	D	C	D	E
	E	E	E	E	E	E		D	D	D	D	C	E
	E	E	E	D	E	E		C	C	C	C	D	E
	D	E	E	D	E	E		C	C	C	D	D	D
	C	E	E	E	E	E		C	C	C	C	C	C
	E	E	E	E	E	E		D	D	D	D	C	D
	E	E	E	D	E	E		B	C	C	B	C	D
	E	E	E	E	E	E		D	C	C	C	C	E
2	C	E	C	D	D	C	3	C	D	D	D	C	D
	C	C	E	D	C	C		C	C	C	D	E	D
	D	C	C	D	C	C		C	D	C	C	D	E
	C	B	C	C	C	D		C	C	D	C	D	C
	E	E	E	B	E	E		C	C	D	C	D	D
	E	D	D	D	D	D		C	C	D	C	D	D
	E	D	E	E	E	E		C	C	C	C	C	C
	E	E	C	D	D	C		D	D	D	E	E	D
	C	D	D	D	C	C		C	D	D	C	D	D
	E	E	E	E	E	D		C	D	E	C	E	E

Appendix E

Acronym

ANOVA	ANalysis Of Variance
API	Application Programming Interface
CD	Collision Detection
CNS	Central Nervous System
DOF	Degree-Of-Freedom
FFT	Fast Fourier Transform
HCI	Human Computer Interface
HIP	Haptic Interface Point
HRTC	Haptic Real-Time Controller
JND	Just Noticeable Difference
KF	Kalman Filtering
NP	Newton Predictor
NPEKF	Newton Predictor Enhanced Kalman Filtering
OBB	Oriented-Bounding-Box
RMS	Root Mean Square

UDP	User Datagram Protocol
VR	Virtual Reality
VE	Virtual Environment

Bibliography

[1] E. Petriu, W. McMath, S. Yeung, and N. Trif, "Active tactile perception of object surface geometric profiles," *IEEE Trans. Instrum. Meas.*, vol. 41, no. 1, pp. 87–92, Feb. 1992.

[2] P Payeur, C Pasca, A.-M. Cretu, and E. Petriu, "Intelligent haptic sensor system for robotic manipulation," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. 1148–1159, Aug. 2006.

[3] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: taxonomy, research status, and chanlleges," *Comput. Graph.*, vol. 21, no. 4, pp. 393–404, 1997.

[4] A. E. Saddik, "The potential of haptics technologies," *IEEE Instrum Meas. Mag.*, vol. 10, no. 1, pp. 10–17, Feb. 2007.

[5] S. J. Lederman and D. Pawluck, "Lessons from the study of biological touch for robotic haptic sensing," in *Advanced Tactile Sensing for Robotics*, H. Nicholls, Ed. World Scientific, 1992.

[6] R. L. Klatzky and S. J. Lederman, "Touch," in *Exp. Psychol.*, A. F. Healy and R. W. Proctor, Eds., 2002, pp. 147–176.

[7] J. Cha, I. Oakley, Y.-S. Ho, Y. Kim, and J. Ryu, "A framework for haptic broadcasting," *IEEE Multimedia Mag.*, vol. 16, no. 3, pp. 2–13, Jul.–Sep. 2009.

[8] M. Eid, S. Andrews, A. Alamri, and A. Saddik, "Hamlat: A haml-based authoring tool for haptic application devleopment," in *Proc. 6th Int. Conf. Haptics: Perception, Devices and Scenarios*, 2008, pp. 857–866.

[9] S. Aukstakalnis, D. Blatner, and S. F. Roth, *Silicon Mirage: The Art and Science of Virtual Reality.* Peachpit Pr., 1992.

[10] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE Comput. Graph. Appl*, vol. 24, no. 2, pp. 24–32, Mar.–Apr. 2004.

[11] B. Knörlein, M. D. Luca, and M. Harders, "Influence of visual and haptic delays on stiffness perception in augemented reality," in *Proc. Int. Symp. Mixed and Augmented Reality (ISMAR 2009)*, Orlando, USA, Oct.19–22 2009, pp. 49–52.

[12] S. Dodeller and N. D. Georganas, "Transport layer protocols for telehaptics update messages," in *Proc. 22nd Biennial Symp. Commun.*, 2004.

[13] H. A. Osman, M. Eid, and A. E. Saddik, "Evaluating ALPHAN: a communication protocol for haptic interaction," in *Proc. Symp. Haptic Interfaces for Virtual Environ. and Teleoperator Syst.*, 2008, pp. 361–366.

[14] A. Liu, F. Tendick, K. Cleary, and C. Kanufmann, "A survey of surgical simulation: Applications, technology, and education," *Presence: Teleoperators Virtual Environ.*, vol. 12, no. 6, pp. 599–614, Dec. 2003.

[15] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, "Haptic feedback enhances force skill learning," in *Proc. 2nd Joint EuroHaptics Conf. and Symp. Haptic Interfaces for Virtual Environ. and Teleoperator Syst.*, Washington, DC, USA, Mar.22–24, 2007, pp. 21–26.

[16] O. A. J. van der Meijden and M. P. Schijven, "The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review," *Surg. Endosc.*, vol. 23, no. 6, pp. 1180–1190, Jun. 2009.

[17] T. Coles, N. W. John, D. A. Gould, and D. G. Caldwell, "Haptic palpation for the femoral pulse in virtual interventional radiology," in *Proc. 2nd Int. Conf. Advances in Computer-Human Interactions*, Feb. 2009, pp. 193–198.

[18] B. Bayart, A. Drif, A. Kheddar, and J.-Y. Didier, "Visuo-haptic blending applied to a tele-touch-diagnosis application," in *Proc. HCI Int. 2007*, Beijing, China, Jul.22-27 2007, pp. 617–626.

[19] X. Shen, J. Zhou, A. Hamam, S. Nourian, N. R. El-Far, F. Malric, and N. D. Georganas, "Haptic-enabled telementoring surgery simulation," *IEEE Multimedia*, vol. 15, no. 1, pp. 64–76, 2008.

[20] R. F. Boian, J. E. Deutsch, C. S. Lee, G. C. Burdea, and J. Lewis, "Haptic effects for virtual reality-based post-stroke rehabilitation," in *Proc. 11th Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst. (HAPTICS03)*, LA, California, USA, Mar.22–23 2003, pp. 247–253.

[21] A. Barghout, A. Alamri, M. Eid, and A. E. Saddik, "Haptic rehabilitation exercises performance evaluation using automated inference systems," *Int. J. Advanced Media Commun.*, vol. 3, no. 1/2, pp. 197–214, 2009.

[22] G. Sankaranarayanan, S. Weghorst, M. Sanner, A. Gillet, and A. Olson, "Role of haptics in teaching structural molecular biology," in *Proc. 11th Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst. (HAPTICS03)*, LA, California, USA, Mar.22–23 2003, pp. 363–366.

[23] N. L. Bernstein, D. A. Lawrence, and L. Y. Pao, "Friction modeling and compensation for haptic interfaces," in *Proc. 1st Joint Eurohaptics Conf. Symp. Haptic Interfaces for Virtual Evnironment Teleoperator Syst. (WHC05)*, Mar. 2005, pp. 290–295.

[24] A. Gregory, M. Lin, S. Gottschalk, and R. Taylor, "A framework for fast and accurate collision detection for haptic interaction," in *Proc. Virtual Reality Conf. 1999*, Mar. 1999, pp. 38–45.

[25] M. Glencross, R. Hubbold, and B. Lyons, "Dynamic primitive caching for haptic rendering of large-scale models," in *Proc. World Haptic Conf.*, Pisa, Italy, Mar. 18–20, 2005, pp. 517–518.

[26] Y. Adachi, T. Kumano, and K. Ogino, "Intermediate representation for stiff virtual object," in *Proc. IEEE Virtual Reality Annual Int. Symp.*, Research Triangle Park, NC, 1995, pp. 203–210.

[27] D. Constantinescu, S. E. Salcudean, and E. A. Croft, "Haptic feedback using local models of interaction," in *Proc. 11th Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst. (HAPTICS'03)*, 2003, pp. 411–416.

[28] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. Int. Conf. Intelligent Robots Syst. (IROS95)*, vol. 3, 1995, pp. 146–151.

[29] D. D. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proc. 24th Annual Conf. Computer graphics and interactive techniques*, Aug. 1997, pp. 345–352.

[30] "Reachin API 3.0 programmer's guide," Reachin Technologies Inc., 2002.

[31] L. N. Verner and A. M. Okamura, "Force & torque feedback vs force only feedback," in *Proc. 3rd Joint Eurohaptics Conf. and Symp. Haptic Interfaces for Virtual Environ. and Teleoperator Syst.*, Salt Lake City, UT, USA, Mar.18–20 2009, pp. 406–410.

[32] C. H. Ho, C. Basdogan, and M. A. Srinivasan, "Ray-based haptic rendering: Force and torque interactions between a line probe and 3d objects in virtual environments," *Int. J. Robot. Res.*, vol. 19, no. 7, pp. 668–683, 2000.

[33] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *ASME J. Dyn. Syst., Meas. Control*, vol. 104, pp. 65–77, Mar. 1982.

[34] J. Zhou, N. D. Georganas, E. M. Petriu, X. Shen, and F. Marlic, "Modelling contact forces for 3d interactive peg-in-hole virtual reality operations," in *Proc. I²MTC 2008 - IEEE Int. Instrum. Meas. Technol. Conf.*, Victoria, Canada, May 11–16, 2008, pp. 1397–1402.

[35] L. A. Jones and S. J. Lederman, *Human Hand Function.* Oxford University Press, Apr. 2006.

[36] A. M. Okamura, M. R. Cutkosky, and J. T. Dennerlein, "Reality-based models for vibration feedback in virtual environments," *IEEE/ASME Trans. Mechatronics*, vol. 6, pp. 245–252, 2001.

[37] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer, "Improving contact realism through event-based haptic feedback," *IEEE Trans. Visualization and Comput. Graph.*, vol. 12, pp. 219–230, 2006.

[38] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 614–637, Oct. 1993.

[39] C. Caradima, "Time delay compensation in teleoperation over the internet," Master's thesis, University of Waterloo, Waterloo, Canada, 1999.

[40] N. Sakr, J. Zhou, N. D. Georganas, and J. Zhao, "Prediction-based haptic data reduction and transmission in telementoring systems," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1727–1736, Aug. 2009.

[41] D. Sha, J. Patton, and F. A. Mussa-Ivaldi, "Minimum jerk reaching movements of human arm with mechanical constraints at endpoint," *Int. J. Computers, Syst. Signals*, vol. 7, no. 1, pp. 41–50, 2006.

[42] V. Hayward and O. R. Astley, "Performance measures for haptic interfaces," in *Proc. 7th Int. Symp. Robot. Res.*, G. Giralt and G. Hirzinger, Eds. Springer Verlag, 1996, pp. 195–207.

[43] M. Mahvash and A. Okamura, "Friction compensation for enhancing transparency of a teleoperator with compliant transmission," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1240–1246, Dec. 2007.

[44] M. Mahvash, J. Gwillian, R. Agarwal, B. Vagvolgyi, D. Y. L. Su, and A. M. Okamura, "Force-feedback surgical teleoperator: controller design and palpation experiments," in *Proc. 16th Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Mar. 2008, pp. 465–471.

[45] D. A. Lawrence, L. Y. Pao, A. M. Dougherty, M. A. Salada, and Y. Pavlou, "Rate-hardness: a new performance metric for haptic interfaces," *IEEE Trans. Robot. Autom.*, vol. 16, no. 4, pp. 357–371, Aug. 2000.

[46] D. Constantinescu, S. E. Salcudean, and E. A. Croft, "Impulsive forces for haptic rendering of rigid contacts," in *Proc. 35th IRS Int. Symp. Robot.*, Paris, France, 2004, pp. 1–6.

[47] L. A. Jones, "Kinesthetic sensing," unpublished.

[48] H. Z. Tan, N. I. Durlach, G. L. Beauregard, and M. A. Srinivasan, "Manual discrimination of compliance using active pinch grasp: the roles of force and work cues," *Percept. Psychophys.*, vol. 57, no. 4, pp. 495–510, 1995.

[49] K. R. Boff, L. Kaufman, and J. P. Thomas, Eds., *Handbook of Perception and Human Performance.* Wiley-Interscience, Apr. 1986, vol. I.

[50] G. C. Burdea, *Force and Touch Feedback for Virtual Reality.* New York: Wiley, 1996.

[51] M. A. Srinivasan and J. Chen, "Human performance in controlling normal forces of contact with rigid objects," *ASME Dyn. Syst. & Control: Advances in Robot., Mechatronics and Haptic Interfaces*, vol. DSC-49, pp. 119–125, 1993.

[52] L. A. Jones and I. W. Hunter, "A perceptual analysis of stiffness," *Exp. Brain Res.*, vol. 79, pp. 150–156, 1990.

[53] L. A. Jones, I. W. Hunter, and R. J. Irwin, "Differential thresholds for limb movement measured using adaptive techniques," *Percept. Psychophys.*, vol. 52, pp. 529–535, 1992.

[54] N. Forrest, S. Baillie, and H. Z. Tan, "Haptic stiffness identification by veterinarians and novices: A comparison," in *Proc. 3rd Joint EuroHaptics Conf. Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Salt Lake City, UT, USA, Mar.19–20 2009, pp. 646–651.

[55] H. Z. Tan, M. A. Srinivasan, B. Eberman, and B. Cheng, "Human factors for the design of force-reflecting haptic interfaces," *ASME Dyn. Syst. Control*, vol. 55-1, 1994.

[56] G. Degersem, "Kinaesthetic feedback and enhanced sensitivity in robotic endoscopic telesurgery," Ph.D. dissertation, Department of Mechanical Engineering, Catholic University of Leuven, 2005.

[57] G. Westling and R. S. Johansson, "Responses in glabrous skin mechanoreceptors during precision grip in humans," *Exp. Brain Res.*, vol. 66, no. 1, pp. 128–140, 1987.

[58] P. D. Neilson, "Speed of response or bandwidth of voluntary system controlling elbow position in intact man," *Med. & Biol. Eng.*, vol. 10, pp. 450–459, 1972.

[59] E. Kunsech, F. Binkofski, and H.-J. Freund, "Invariant temporal characteristics of manipulative hand movements," *Exp. Brain Res.*, vol. 78, pp. 539–546, 1989.

[60] J. E. 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.

[61] N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. Salisbury, and C. Melchiorri, "The effect of quantization and coulomb friction on the stability of haptic rendering,"

in *Proc. 1st Joint Eurohaptics Conf. Symp. Haptic Interfaces for Virtual Environ. and Teleoperator Syst. (WHC05)*, 2005, pp. 237–246.

[62] D. S. Kwon and K. Y. Woo, "Control of the haptic interface with friction compensation and its performance evaluation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robot. Syst. (IROS 2000)*, Takamatsu, Japan, Oct.31 2000, pp. 1620–1625.

[63] G. Borghesan and C. Melchiorri, "Model and modelless friction compensation application to a defective haptic interface," in *Proc. 6th Int. Conf. Haptics: Perception, Devices and Scenarios*, Madrid, Spain, Jun.10–13 2008, pp. 94–103.

[64] D. Bi, Y. Li, S. Tso, and G. Wang, "Friction modeling and compensation for haptic display based on support vector machine," *IEEE Trans. Industrial Electronics*, vol. 51, no. 2, pp. 491–500, Apr. 2004.

[65] T. H. Massie and K. J. Salisbury, "Phantom haptic interface: a device for probing virtual objects," in *Proc. ASME Winter Annual Meeting, Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Chicago, IL, USA, Nov.6–11 1994, pp. 295–299.

[66] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004.

[67] K. J. Kuchenbecker, "Characterizaing and controlling the high-frequency dynamics of haptic interfaces," Ph.D. dissertation, Stanford University, Jun. 2006.

[68] V. Hayward and K. E. MacLean, "Do it yourself haptics, part I," *IEEE Robot. Autom. Mag.*, vol. 14, no. 4, pp. 88–104, Dec. 2007.

[69] B. Hannaford, "DC motors for robotic manipulators and haptic devices," The University of Washington, Lecture/Class, 2005.

[70] "Haptic master user manual," FCS Control Systems, 2004.

[71] J. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice Hall, 1991.

[72] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 465–474, 1999.

[73] K. A. Kaczmarek, J. G. Webster, P. B. y Rita, and W. J. Tompkins, "Electrotactile and virotactile displays for sensory substitution systems," *IEEE Trans. Biomed. Eng.*, vol. 38, no. 1, pp. 1–16, Jan. 1991.

[74] S. Salcudean and T. Vlaar, "On the emulation of stiff walls and static friction with a magnetically levitated input/output device," *J. Dyn. Syst., Meas. Control*, vol. 119, no. 1, pp. 127–132, 1997.

[75] G. Han, S. Jeon, and S. Choi, "Improving perceived hardness of haptic rendering via stiffness shifting: an initial study," in *Proc. 16th ACM Symp. Virtual Reality Software and Technology*, Kyoto, Japan, Nov.18–20 2009, pp. 87–90.

[76] D. Constantinescu, S. E. Salcudean, and E. A. Croft, "Haptic rendering of rigid body collisions," in *Proc. IEEE Int. Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Chicago, USA, 2004.

[77] ——, "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 309–323, Jun. 2005.

[78] B. Birch and D. Constantinescu, "Modulated impulsive forces for haptic rendering of rigid contact," in *Proc. IMECE 2007*, Seattle, Washington, USA, Nov.11–15 2007.

[79] M. A. Srinivasan, G. L. Beauregard, and D. L. Brock, "The impact of visual information on the haptic perception of stiffness in virtual environments," in *Proc. ASME Dyn. Syst. Control Division*, 1996, pp. 555–559.

[80] A. H. C. Gosline and V. Hayward, "Eddy current brakes for haptic interfaces: Design, identification, and control," *IEEE Trans. Mechatronics*, vol. 13, no. 6, pp. 669–677, 2008.

[81] M. C. Çavuşoğlu, D. Feygin, and F. Tendick, "A critical study of the mechanical and electrical properties of the phantom haptic interface and improvements for high-performance control," *Presence: Teleoperators Virtual Environ.*, vol. 11, no. 6, pp. 555–569, Dec. 2002.

[82] B. Friedland, "Optimum steady-state position and velocity estimation using noisy sampled position data," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-9, no. 6, pp. 906–911, Nov. 1973.

[83] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerospace Electronic Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[84] P. R. Bélanger, "Estimation of angular velocity and acceleration from shaft encoder measurements," in *Proc. 1992 IEEE Int. Conf. Robot. Autom.*, Nice, France, May 1992, pp. 585–592.

[85] P. R. Bélanger, P. Dobrovolny, A. Helmy, and X. Zhang, "Estimation of angular velocity and acceleration from shaft encoder measurements," *Int. J. Robot. Res.*, vol. 17, no. 11, pp. 1225–1233, Nov. 1998.

[86] G. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Prentice-Hall, 2002.

[87] F. Janabi-Sharif, V. Hayward, and C. J. Chen, "Discrete-time adaptive windowing for velocity estimation," *IIEEE Trans. Control Syst. Technol.*, vol. 8, no. 6, pp. 1003–1009, Nov. 2000.

[88] G. Liu, "On velocity estimation using position measurements," in *Proc. American Control Conf.*, Anchorage, AK, May 8–12, 2002, pp. 1115–1120.

[89] J. D. Han, Y. Q. He, and W. L. Xu, "Angular acceleration estimation and feedback control: an experimental investigation," *Mechatronics*, vol. 17, no. 9, pp. 524–532, Nov. 2007.

[90] S. J. Ovaska, "Newton-type predictors-a signal processing oriented viewpoint," *Signal Processing*, vol. 25, no. 2, pp. 251–257, Nov. 1991.

[91] N. Hogan, "Controlling impedance at the man/machine interface," in *Proc. IEEE Int. Conf. Robot. Autom.*, Scottsdale, AZ, USA, May 14–19, 1989, pp. 1626–1631.

[92] W. Abend, E. Bizze, and P. Morasso, "Human arm trajectory formation," *Brain*, vol. 105(Pt 2), pp. 331–348, Jun. 1982.

[93] W. L. Nelson, "Physical principles for economics of skilled movements," *Biol. Cybern.*, vol. 46, pp. 135–147, 1983.

[94] N. Hogan, "An organizing principle for a class of voluntary movements," *J. Neurosci.*, vol. 4, no. 11, pp. 2745–2754, Nov. 1984.

[95] T. Flash and N. Hogan, "The coordination of arm movements: an experimental confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, Jul. 1985.

[96] P. Viviani and C. Terzuolo, "Trajectory determines movement dynamics," *Neuroscience*, vol. 7, no. 2, pp. 431–437, 1982.

[97] P. Viviani and T. Flash, "Minimum-jerk, two-third power law, and isochrony: converging approaches to movement planning," *J. Exp. Psychol.: Hum. Percept. Perform.*, vol. 21, no. 1, pp. 32–53, 1995.

[98] P. Viviani and R. Schneider, "A developmental study of the relationship between geometry and kinematics in drawing movements," *J. Exp. Psychol.: Hum. Percept. Perform.*, vol. 17, no. 1, pp. 198–218, 1991.

[99] D. Sternad and S. Schaal, "Segmentation of endpoint trajectories does not imply segmented control," *Exp. Brain Res.*, vol. 124, pp. 118–136, 1999.

[100] N. Bernstein, *The Coordination and Regulation of Movements.* Oxford: Pergamon Press, 1976.

[101] K. Mehrotra and P. R. Mahapatra, "A jerk model for tracking highly maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 4, pp. 1094–1105, Oct. 1997.

[102] G. J. Bierman, *Factorization methods for discrete sequential estimation.* Academic Press, 1977.

[103] C. L. Thornton, "Triangular covariance factorizations for kalman filtering," Ph.D. dissertation, University of California at Los Angeles, School of Engineering, 1976.

[104] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 2nd ed. Wiley-Interscience, 2001.

[105] X. Shen, J. Zhou, A. E. Saddik, , and N. D. Georganas, "Architecture and evaluation of tele-haptic environments," in *Proc. 8th IEEE Int. Symp. Distributed Simulation and Real-Time Applications (DS-RT04)*, Budapest, Hungary, Oct. 2004, pp. 53–60.

[106] D. Feygin, M. Keehner, and F. Tendick, "Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill," in *Proc. 10th Symp. Haptic Interfaces for Virtual Envir. Teleoperator Systs.*, Orlando, FL, USA, Mar. 2002, pp. 40–47.

[107] R. B. Gillespie, M. S. O'Modhrain, P. Tang, D. Zaretzky, and C. Pham, "The virtual teacher," in *Proc. ASME Dyn. Syst. Control Division*, Anaheim, CA, 1998, pp. 171–178.

[108] R. L. Williams II, M. Srinvastava, R. R. Conatser Jr., and J. H. Howell, "Implementation and evaluation of a haptic playback system," *Haptics-e*, vol. 3, no. 3, 2004.

[109] M. A. Otaduy and M. C. Lin, *High Fidelity Haptic Rendering.* Morgan & Claypool Publishers, 2006.

[110] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., 1994.

[111] L. Jandura and M. A. Srinvasan, "Experiments on human performance in torque discrimination and control," *ASME Dyn. Syst. Control*, vol. 55-1, pp. 369–375, 1994.

[112] B. Taati, A. M. Tahmasebi, and K. Hashtrudi-Zaad, "Experimental identification and analysis of the dynamics of a phantom premium 1.5a haptic device," *Presence: Teleoperators Virtual Environ.*, vol. 17, no. 4, pp. 327–343, Aug. 2008.

[113] D. Ma and J. M. Hollerbach, "Identifying mass parameters for gravity compensation and automatic torque sensor calibration," in *Proc. 1996 Int. Conf. Robot. Autom.*, Minneapolis, Minnesota, USA, Apr. 1996, pp. 661–666.

[114] W. Levine, "System identification when noise is negligible," in *The Control Handbook.* CRC Press IEEE Press, 1995.

[115] A. M. Okamura, J. T. Dennerlein, and R. D. Howe, "Vibration feedback models for virtual environments," in *Proc. 1998 IEEE Int. Conf. Robot. Autom.*, vol. 1, Leuven, Belgium, May 1998, pp. 674–679.

[116] P. Buttolo, R. Oboe, and B. Hannaford, "Architecture for shared haptic virtual environments," *Comput. Graph.*, vol. 21, pp. 421–429, Oct.–Aug. 1997.

[117] R. Wirz, M. Ferre, R. Marin, J. Barrio, J. M. Claver, and J. Ortego, "Efficient transport protocol for networked haptics applications," in *Proc. 6th Int. Conf. Haptics: Perception, Devices, and Scenarios*, Madrid, Spain, 2008, pp. 3–12.

[118] A. Boukerche, S. Shirmohammadi, and A. Hossain, "Prediction-based decorators for distributed collaborative haptic virtual environments," *Int. J. Computer Appl. Technol.*, vol. 29, no. 1, pp. 81–88, Jul. 2007.

[119] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," *IEEE J. Ocean. Eng.*, vol. 16, no. 1, pp. 152–162, Jan. 1991.

[120] G. Sankaranarayanan and B. Hannaford, "Experimental comparison of internet haptic collaboration with time-delay compensation techniques," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May19–22 2008, pp. 206–211.

[121] Z. Li and D. Constantinescu, "Remote dynamic proxies for wave-based peer-to-peer haptic interaction," in *Proc. IEEE World Haptics Conf.*, Salt Lake City, USA, 2009, pp. 553–558.

[122] N. Sakr, J. Zhou, N. D. Georganas, J. Zhao, and E. M. Petriu, "Robust perception-based data reduction and transmission in tele-haptic systems," in *Proc. World Haptics, 3rd Joint Eurohaptics Conf. Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Salt Lake City, USA, Mar.18–20 2009, pp. 214–219.

[123] N. Gurari, K. J. Kuchenbecker, and A. M. Okamura, "Stiffness discrimination with visual and proprioceptive cues," in *Proc. World Haptics, 3rd Joint EuroHaptics Conf. Symp. Haptic Interfaces for Virtual Environ. Teleoperator Syst.*, Salt Lake City, USA, Mar.18–20 2009.

[124] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 55–64, Mar./Apr. 2004.

[125] M. Corno and M. Žefran, "Haptic playback: Modeling, controller design, and stability analysis," in *Proc. Robot. Sci. Syst.*, Philadelphia, USA, Aug.16–19, 2006.