1-2007

COMPARATIVE STUDY OF HAPTIC AND VISUAL FEEDBACK FOR KINESTHETIC TRAINING TASKS

Ravikiran Singapogu
Clemson University, rsingap@clemson.edu

COMPARATIVE STUDY OF HAPTIC AND VISUAL FEEDBACK FOR KINESTHETIC TRAINING TASKS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Ravikiran Singapogu
May 2007

Accepted by:
Dr. Timothy Burg, Co-Chair
Dr. Sam Sander, Co-Chair
Dr. Darren Dawson

ABSTRACT

Haptics is the sense of simulating and applying the sense of human touch. Application of touch sensations is done with haptic interface devices. The past few years has seen the development of several haptic interface devices with a wide variety of technologies used in their design. This thesis introduces haptic technologies and includes a survey of haptic interface devices and technologies. An improvement in simulating and applying touch sensation when using the Quanser Haptic Wand with proSense™ software is suggested in this work using a novel five degree-of-freedom algorithm. This approach uses two additional torques to enhance the three degree-of-freedom of force feedback currently available with these products. Modern surgical trainers for performing laparoscopic surgery are incorporating haptic feedback in addition to visual feedback for training. This work presents a quantitative comparison of haptic versus visual training. One of the key results of the study is that haptic feedback is better than visual feedback for kinesthetic navigation tasks.

DEDICATION

"The fear of the LORD is the beginning of knowledge…", Proverbs 1:7

To my parents, Vijay and Sukanya Bhaskar Singapogu. Their constant love and support has made this possible.

ACKNOWLEDGMENTS

I would first like to thank my advisors, Dr. Tim Burg and Dr. Sam Sander, for their support and guidance throughout this work. Their example has taught me to be a better student and strive always to learn more. I would like to thank Dr. Darren Dawson for serving on my committee and encouraging me with his "hope you can graduate" words. Special thanks go to my assistantship supervisor, Barbara Weaver, for her understanding during the thesis writing phase. I would like to thank Dr. JeAnne Burg for her help with statistical analysis. I'm also grateful to William Cobb, MD, for his inspiration behind this project.

I would like to thank Donbin Lee for his guidance regarding my work. Gigi Grizzard helped me understand the scripting capabilities of MATLAB; I'm grateful for her help. I would also like to Ninad Pradhan and Hariprasad Kannan, in the Controls and Robotics group, for their thoughtful conversations and altruism.

My heartfelt thanks go to Rachie for her perennial support throughout my graduate experience. Her serene encouragement was a big sustaining force behind this work. Finally, this acknowledgement would not be complete without thanking my parents and my siblings, Steny and Shiny, for their constant prayer and support throughout my life.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER ONE
INTRODUCTION

Haptics is a rapidly emerging field spanning computer science, biology, psychology, physics and engineering. The discipline of haptics deals broadly with the simulation of the human sense of touch. The proof of its gaining popularity is the availability of a wide range of commercially sold haptic interface devices. This indicates the growth of haptics considering the field was insignificant just a few years ago with only a few research professionals working in the arena. The field of haptics shows much promise for mechanisms and capabilities that can be added to the computer interface devices of tomorrow. Haptics can be seen in the field of gaming, to add another dimension to immersive game environments [1]. The Novint Falcon®™ is a gaming joystick device that lets users "feel" characters and objects on the monitor [2].

This masters thesis presents an algorithm extending the use Quanser Inc.,'s 5 degree-of-freedom (DOF) Haptic Wand and studying haptics as a means of surgical training. This document is organized into four chapters. The following is an explanation of each of the three remaining chapters with mention of application areas for this work.

Chapter 1 presents a general introduction to the field of haptics and overviews haptic interface. The first section presents basic definitions and terms. The distinctions between the three types of haptic feedback are explained. An introduction to haptic interfaces is given as well as some common mechanisms for haptic interface technology. A haptic interface is defined to be the mechanical link between the user and computer simulated touch. It is a

mechanical device built to emulate touch based on principles of robotics. A brief survey of haptic interfaces, commercial and research, is presented in this chapter. One of the goals of Chapter 2 is to help describe the key components in a haptic system. The role of software in haptic and visual rendering is discussed. The several areas of application of haptics relevant to this research are also discussed.

In Chapter 3, the Quanser 5 DOF Haptic wand is introduced. The haptic interface used Quanser Inc.'s 5 DOF Haptic Wand. A brief introduction to relevant technical details of this device is presented. The software component, MATLAB/Simulink with Handshake Inc.'s proSense™ toolbox is described. This toolbox enables the building of ready-to-use VRML worlds. A concise tutorial of haptic world building using this platform is presented in Appendix A. It is shown that proSense software's haptic rendering is limited to 3 DOF even for a 5 DOF device. A multi-point torque method is proposed to extend this 3 DOF functionality to 5 DOF. Using the multi-point torque rendering approach, forces are rendered in three linear dimensions and in two additional dimensions. This capability enhances haptic realism. The detailed algorithm for 5 DOF rendering presented. Finally, this algorithm is demonstrated using a test model. The results from test observations suggest that the 5 DOF rendering algorithm functions as expected.

Haptics for surgical training tasks is the subject of Chapter 4. A brief literature review for haptics based surgical training and simulations are provided. Haptics has found favor in the medical community because of its high quality of feedback in training for Minimally Invasive Surgical (MIS) procedures. Laparoscopy is a procedure where small incisions are made on the patient and special instruments and cameras perform a comparatively less invasive

surgery. Current generation laparoscopic trainers use visual feedback inspite of the fact that the actual surgery relies heavily on kinesthetic feedback. An experiment was conducted at Clemson University's Controls and Robotics lab to quantitatively compare haptic versus visual feedback. The initial hypothesis was made that haptic feedback is better than visual feedback. Participants in this research were trained using either visual feedback or haptic feedback. After the training period, participants were asked to retrace the path they learned during training. The results of this research are presented along with a detailed data analysis. The research confirms the initial hypothesis that haptic feedback is indeed better than visual feedback. Other results are also presented relevant to the analysis.

Appendices B and C provide additional information to support the experiments described in this thesis.

CHAPTER TWO

INTRODUCTION TO HAPTICS TERMINOLOGY AND TECHNOLOGY

The field of haptics has seen tremendous advances in the past decade or so. Because of its interdisciplinary nature, with fields of interest including robotics (including telerobotics), computer graphics, engineering, psychophysics and neurosciences, there has been considerable intellectual investment. In this chapter a brief overview of the field of haptics is presented along with an introduction to common pervasive terms. The motivation for understanding haptic technology is to explore better ways of haptic rendering, thus increasing realism in "feeling" and thus expanding the range of applications. Also, as presented in Chapter 4, haptics can be a great asset for surgical training which is one of the motivations of this work.

2.1 What is Haptics?

The word "haptic" comes from the Greek word "*haphe*" which means "pertaining to the sense of touch" [3]. The "haptic" sense is the first to develop as a fetus. The relationship of haptic sensory modes in relationship to other modes (sight, auditory, etc) is a topic of great interest in the present psychophysics community. For a broad definition of haptics, we turn to a definition by Salisbury *et al.*, where it is defined as:

> "touch interactions (physical contact) that occur for the purpose of perception or manipulation of objects" [4].

It should be noted that these interactions can be man-machine (human and robot interface), man-object (human and a real object) or man-simulated object (human interacting with a

virtual, simulated object). All of the aforementioned interactions are considered broadly under the umbrella of "haptics". Interest in the field of haptics in the 21st century began in the year 1968, with studies to understand human touch perception and interactions [20]. In parallel, efforts were being made to build electromechanical systems that were capable of providing force feedback to their human users; that is, to provide new user interfaces to machines [7]. Some of the earliest examples of efforts along these lines include: Corliss & Johnson (1968) and Mosher (1964) with their design of the "Handyman", an electrohydraulic device with two 10 degrees-of-freedom (DOF) arms [5] [6]. Thring (1983) later improvised this device into the "Hardiman" featuring a "whole- body" exoskeleton [7]. It is interesting to note that these early complex explorations would lead to the motivation and current understanding of haptics. On the one hand was the natural science reseacher's quest to understand human touch interactions with real rigid and non-rigid bodies while on the other hand was the interest in the robotics community to design and build force reflecting (*i.e.*, force feedback) devices. For example, when researchers were working on building a dexterous robotic hand for manipulating in hostile environments they were faced with "[how building the device] *was much more complex and subtle than their initial naive hopes had suggested"* [4].

Before delving into the technicalities of haptic feedback it would serve well to define the types of haptic feedback in modern technology and their inherent distinctions. Often the three kinds of feedback are seen as one and therefore no distinction among them is seen. That view is incorrect. Although seemingly trivial, these distinctions become all the more important when attempting to artificially create them. Three haptic feedback terms are now introduced with their meanings.

(a.) Tactile Feedback : This refers to the sensation felt by the skin during touch or external contact. Tactile feedback is sensed by high bandwidth (50-350 Hz) receptors placed very close to the skin [8]. These receptors are responsible for tactile information, *i.e.*, geometry, "slipiness", temperature, and texture details. Tactile feedback can be used for haptic symbol generation. For example, Braille symbols can be communicated using tactile haptic feedback because of the role of the skin in Braille symbol perception.

(b.) Kinesthetic Feedback: We have noted the presence of receptors very close to the skin responsible for tactile perception. Deeper in the body, there are other receptors present in muscle tendons and joints that sense actual force. That is, these receptors sense actual force exerted by contact, compliance and weight. Kinesthetic feedback can be defined as the force mediated by muscles, tendons and joints when stimulated by bodily movements and tensions [8]. It has also been suggested that kinesthetic forces may include knowing the locations of body parts with respect to each other [9].

The reader should note that in this thesis, when haptic feedback is mentioned, we mean the kinesthetic force feedback.

(c.) Proprioceptory Feedback: Simply stated, proprioception is kinesthetic sense plus the sense of balance or equilibrium. Burdea refers to it as "stimuli arising within the organism" [8]. Proprioception provides feedback related to body posture and the location of body parts with respect to each other. This information is sensed by receptors in the skeletal joints, the inner ear and from the central nervous system [9]. One distinction that can be made here from kinesthetic feedback is that proprioceptive feedback includes the sense of balance.

Early haptic research was directed towards understanding sensory modes and the more basic concepts of human haptic sensation. Until the late 1990's, haptics dealt with the study of "real" objects, mostly rigid. However, with greater capacity processors and cheaper and larger memory, there was the idea or virtual haptics. The term "Computer Haptics" was coined to describe the human interaction with a computer model. Researchers developed virtual objects that had haptic properties assigned to them. In this approach, when a user "touched" a virtual object with the aid of a haptic interface device there would be equivalent computer generated "forces" felt by the user through the haptic interface. This created considerable interest in the computer software community and held the promise for an efficient way to simulate physics based interactions [10].

2.2 Haptic Interface Technology

There are two primary components to any computer haptic system: a software component that describes the behavior of the virtual object and the user interface device. These components define haptic interaction. A high-level system diagram is shown in Figure 1.

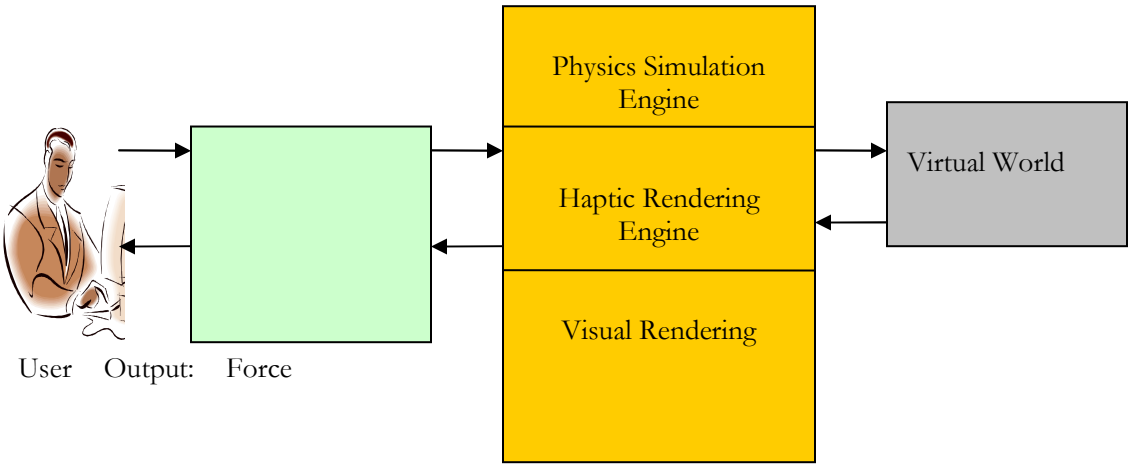

Figure 1 : Overview of a computer haptic system.

2.2.1 Haptic Rendering

"[Haptic] Rendering refers to the process by which desired sensory stimuli are imposed on the user to convey information about a virtual haptic object." [4]. In other words, haptic rendering deals with assigning certain haptic properties to the object, such that when the user "feels" it through a haptic interface, a realistic feel of the object is produced. Consider the example of a ball rendered in 3D using two different techniques: simple shading and ray-tracing. Just as both will be visually different, even so, objects rendered with different haptic rendering techniques or objects rendering with different haptic fields will "feel" different. Haptic rendering, therefore, deals with providing realistic feel to computer simulated objects when they are manipulated using haptic interfaces [11]. The task of the haptic interface is to convey the computer-controlled forces to the user [12].

Haptic Rendering can again be broken down into two basic operations:

(a) Collision Detection, and

(b) Collision Response.

Collision detection deals with knowing the position of the virtual end-effector in the virtual world. The generic haptic interface user input device (stylus, pantograph, probe) is assigned an avatar in the virtual world. As this avatar moves in the virtual world, the position of the end effector is sensed. Collision detection deals with tracking the location of the avatar in relation to haptic objects in the virtual world. If the avatar is in free space and not colliding or touching a virtual object then the resultant contact forces on the interface will be zero. However, if the avatar is indeed touching a virtual object, there should be resultant, computer-controlled forces felt by the user touching the end-effector. There are several collision detection techniques that are in use today to detect the overlap of two objects (or,

object with the virtual haptic interface) [13] [14]. The choice of approaches depends on the resolution of haptic forces and rate of haptic and visual rendering.

Collision response, on the other hand, calculates the appropriate forces that need to be passed onto the haptic interface device. Each virtual object is assigned certain haptic parameters and properties. Also, the algorithm for calculating forces caused by collision with haptic objects is predetermined. Based on these object properties and position of the haptic interface (how far is it into the object?), the penalty forces are calculated. As a general rule, regardless of the method used to represent the interface, the reaction force is calculated using

$$F = Kx \qquad \text{(generic)},$$

where K is the stiffness of the object and x is the penetration depth (into the object). There are several collision response algorithms that propose a realistic way to generate penalty forces; a few examples are [15] [16]. In this thesis, the response algorithm contains an additional viscous damping term, $\dot{x}$, to render stiffness realistically. This also reduces oscillations due to collision, thus enabling smoother, time-stable rendering. The above equation thus becomes

$$F = Kx + D\dot{x}$$

2.3 Haptic Interfaces

An interface, according to the Merriam-Webster online dictionary, is

"a : the place at which independent and often unrelated systems meet and act on or communicate with each other (the man-machine interface)

b : the means by which interaction or communication is achieved at an interface" [17].

Specifically, a human-computer interface provides the means for the user to exchange meaningful information with the computer. A trivial but all-pervading example of a simple interface [18] device is a mouse. A mouse is used to manipulate and transmit user intent to the computer. Conventionally, interfaces have been non-programmable in that their mechanical properties do not change while interacting with the machine. As can be demonstrated by the mouse example, there is no change in the mechanical properties of the interface.

The past decade has seen an accelerated effort to make use of greater number of human sensitivities. Haptic perception, as a result, is being researched as one of the foremost potentially useful arenas in expanding realism and quality of user interaction experience. Many, including Ivan Sutherland, regarded as the pioneer in virtual reality systems, have recognized the potential of haptic interaction. He is quoted as saying, "human kinesthetic sense is as yet another independent channel to the brain, a channel whose information is assimilated quite subconsciously" [19].

The goal of haptic interfaces, then, is to have programmable mechanical devices that will change their properties according to user interaction. A haptic device will create a realistic "feel" when interacting with a virtual world [20].

2.3.1 Examples of Haptic Devices

This section will provide a brief survey of haptic devices both commercially available and built for research purposes. Probably, the pioneering haptic devices were exoskeletons, devices worn on the body. Bergamasco *et al.*, built a prototype exoskeleton device capable of

permitting actions such as driving simulation. Burdea *et al.*, built a pneumatically actuated glove that can simulate grasping of virtual objects as shown in Figure 3. Another class of devices are desktop game controllers and joysticks. Haptic knobs were developed by Maclean *et al.*, and later commercialized by immersion Corporation for use in vehicles. Logitech has made available a range of force feedback joysticks for a wide range of gaming and simulation applications. There are certain haptic devices that make use of magnetic levitation: Salcudean *et al.*, designed a small 6 DOF voice coil levitated joystick making use of Lorentz actuators [20] (shown in Figure 2).

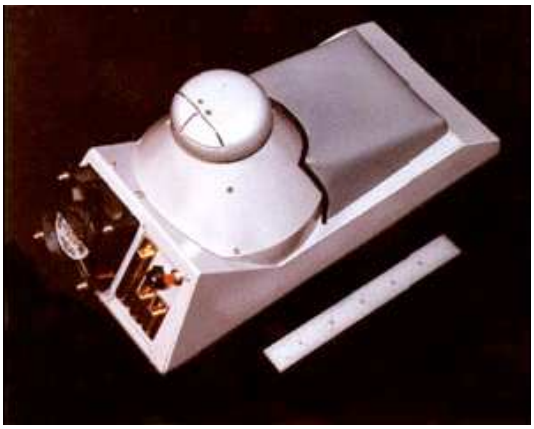

Figure 2: A Magnetic Levitation haptic device.

Figure 3: Haptic grasper developed at Rutgers University.

Arguably the most widely accepted purely haptic device in modern haptic research is the Sensable Phantom™. Though the first devices were 3 DOF, more recent models have 6 DOF capabilities. The Phantom device made the point interaction paradigm of haptic rendering well know. Position is sensed in three dimensions and forces are reflected in three dimensions giving the force vector (F_x, F_y , F_z). This greatly simplifies calculation and processing speed. However, realistic feel is compromised in these cases. The interface avatar in the virtual world should be rendered as a point for accuracy; however, any other rigid body rendered as a point would be unrealistic.

The 5 DOF Haptic Wand made by Quanser Inc., is an example of an interface that has more than 3 DOF, thus increasing the rendering capabilities of the device. While most 3 DOF devices are point-based, wherein the forces are felt at the tip of the end effector, the Haptic Wand's end effector is cylindrical. Because of this geometry, torque rendering is made possible. Also, typically, forces for the Haptic Wand are resolved at the center of the end effector. This, again, can be changed because of the device's end effector geometry. The Haptic Wand is built using the Twin Pantograph mechanism developed by Dr. Salcudean at the University of British Columbia [22]. The Haptic Wand end effector's top is connected to the top pantograph and the bottom is connected to the bottom pantograph. The Haptic Wand, which is the haptic interface device used in this thesis, is discussed in more detail in [22] [21].

2.3.2 Models of Haptic Interaction

Almost every haptic interface is considered as a robot, the specialized function of this class of robots being interaction with humans. The user "feels" a virtual world through the haptic

interface. The virtual world is a simulated environment with haptic properties assigned to each member object. The haptic interface "displays" force sensations on interactions with the remote virtual world. One of the primary areas of research in present-day haptics is directed toward making simulations "feel" more realistic. If, in the virtual world, the user encounters free space, zero resistance to the motion of the end-effector (of the haptic interface) would be expected[1]. However, as will be seen later, there are physical limitations to hardware capabilities which limit realism in interaction.

The reader will now be introduced to the two basic kinds of haptic interfaces based on their energy exchanges. All devices, whether natural or man-made can fall into two broad categories: active or passive. Passive devices (also known as inert devices) can only dissipate mechanical energy. However, this dissipation can be controlled by programming it as a function of position or time. For example, consider a mechanical device with constant elasticity. If the elastic behavior of the device can be computer programmed to reflect a realistic pattern, this kind of model will be considered a passive haptic device [20].

On the other hand, there are active devices (also known as animate devices). The distinction from passive devices is that the "energy exchange between a user and the machine is entirely a function of the feedback control applied" [20]. In other words, active devices are capable of generating energy based on haptic interaction. Active haptic interfaces can be further classified into two categories based on their feedback control mechanism: impedance control and admittance control. The closed-loop control problem for a haptic interface poses a considerable challenge in performance and stability. In impedance control devices, the input

[1] Considering the device is impedance-controlled. This will be discussed in detail later.

is the position of the end-effector and the output is the forces on it. That is, the user moves the device, which will react with forces, if necessary. The actuators will provide the necessary force output. In the virtual world, if the end effector moves in free space, then the resulting force output will be zero. If a haptic object with certain stiffness is encountered, then this will be reflected by the appropriate forces. Impedance devices have stability issues when rendering high-stiffness objects. If the end-effector is in contact with a very stiff wall, then small change in position should cause a very high reaction force(s). However, there is a maximum force that can be applied due to hardware limitations. This causes the device to go unstable and is an area of research.

Admittance control is the "dual" of impedance control. In this paradigm, the user exerts forces on the interface, which is measured and is the input to the device. The response, or output, is in the form of motion (acceleration, velocity or position). In this case, to simulate free space, the device will have to respond with very high change in position. Herein is the cause of stability issues for these devices. Simulating low mass implies having a high control gain which is constrained by hardware limitations. Conversely, to simulate a hard wall, there should be zero change in position which is also not very feasible. Therefore, depending on the nature of the simulated environment, an appropriate scheme of control and type of device can be selected. The end-user application will dictate choice of control.

In this chapter, general haptics terminology and technology was presented. The reader will note the wide variety of applications for haptic technology. This thesis is particularly interested in surgical training applications of haptic technology. A survey of haptic interfaces

is also provided along with some classification criteria. Chapter 3 will describe the use of a specific haptic interface and workstation.

CHAPTER THREE

THE QUANSER 5 DOF HAPTIC WAND AND A MULTI POINT HAPTIC RENDERING ALGORITM

In this chapter, the Quanser 5 degree-of-freedom (DOF) Haptic Wand, introduced in Chapter 2, will be examined. This haptic interface device is used throughout this work. First, a hardware overview is presented, followed by a detailed description of important system specifications. The software component of the haptic workstation is next described. The MATLAB™ environment is used as the basis for real-time operation of the haptic workstation. The reader is introduced to Handshake's proSense™ toolbox for building haptic models. In the second section, a 5 DOF, multi-point haptic rendering algorithm is presented. The major components of the algorithm are discussed in detail. Finally, in the third section, testing of the multi-point rendering algorithm and results are provided.

3.1 Quanser 5 DOF Haptic Wand : Hardware Introduction and Software Implementation

This section is organized into four subsections. First, a brief overview of the Quanser 5 DOF Haptic Wand hardware is presented. This section will contain information about the Wand's development and capabilities. The second section contains selected system parameters of interest. The third section will present the reader with coordinate frames and transformations between frames used to describe the position and orientation of the Wand. Finally, the software to build haptic systems is presented. The Handshake proSense™ toolbox for MATLAB is presented for haptic application building.

3.1.1 Hardware System Presentation

Figure 4: Quanser Inc.'s 5 DOF Haptic Wand.

The 5 DOF Haptic Wand (Figure 4 with top pantograph highlighted) is a product of Quanser Inc., and is based on the research work of Professor Tim Salcudean [21]. Salcudean and colleagues attempted to "mimic any passive environment that a human hand can distinguish" [22]. To do this it was proposed that a manipulator be designed with "most of its actuators at its base" and the components used have low mass and friction [22] [23]. After researching various designs to achieve this mechanism they discovered that the Twin-Pantograph (a pantograph is the five-link closed kinematic chain seen in Figure 4) platform seemed most suitable due to its dexterity in workspace area and static force reflecting capabilities. The original design consisted of "two 3-DOF linkages that are actuated about their folding or waist joints to provide five degrees-of-freedom to a cylindrical end effector" [22]. While the original design made use of seven actuators, the commercial haptic wand makes use of six actuators as shown in Figure 5 [21].

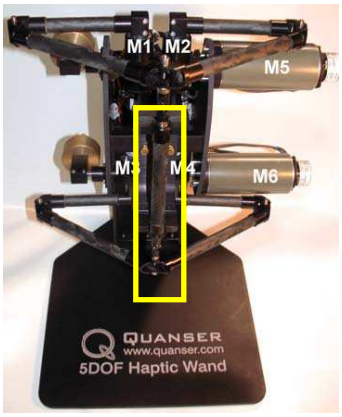

Figure 5: Haptic Wand motor nomenclature.

The Haptic wand consists of two pantographs that are linked together by a cylindrical end-effector. The end-effector is in reality the "wand" and is controlled by the cumulative effect of both pantographs. Each pantograph is driven by two motor pairs, (M1, M2 for the upper section) and (M3, M4 for the lower section), which are in turn driven by a more powerful motor at the shoulders (M5 for the upper shoulder, M6 for the lower shoulder). In addition to these, each pantograph is provided with adjustable weights or counterbalances to minimize the weight of the end-effector. The wand is thus able to output forces in three dimensions and torques about two dimensions (roll and pitch). The third torque, yaw, is passive. Power each of the motors for actuation is supplied by linear power amplifiers. The sensing of position of the end-effector is done using high-resolution optical encoders which read the position of the motor shafts. It should be noted that this is six-dimensional information: position and orientation. The control and operation of the system is via a standard PC equipped with the Quanser Q8 Hardware-In-Loop (HIL) board.

Real-time control for the haptic wand is achieved using MATLAB and Simulink from MathWorks Inc. Quanser provides WinCon (v 5.0) software to build control algorithms. WinCon operates in the MATLAB environment and installs itself as a Simulink toolbox. In

this work, however, WinCon is not used because of its focus on developing control algorithms. The purpose of this work is the building of ready to use haptic worlds and not control algorithms for the device. WinCon does not provide any collision detection or collision rendering algorithms. proSense, on the other hand has ready to use algorithms for force response and collision detection. proSense also offers "hapticizing" a VRML format virtual would with no programming required. For these reasons, Handshake's proSense™ toolbox (v 2.0, 2.1, 2.2) was chosen over WinCon. This software also runs in the MATLAB environment. proSense ™ toolbox has readily available blocks for haptic world building. The graphic file format used for proSense is VRML2. VRML ("*.wrl") files can be readily "hapticised" using custom blocks or haptic shapes can be built using individual shapes. proSense™ accounts for basic shapes such as box, cylinder, cone and also advanced custom shapes using IndexFaceSet and Extrusion blocks. V-realm Builder 2 was used for creating VRML worlds. This software was provided by MathWorks Inc. with the Virtual Reality Toolbox for MATLAB. The interface to the device (HIL Board) and design software is through MATLAB's Real time Workshop (v 6.5).

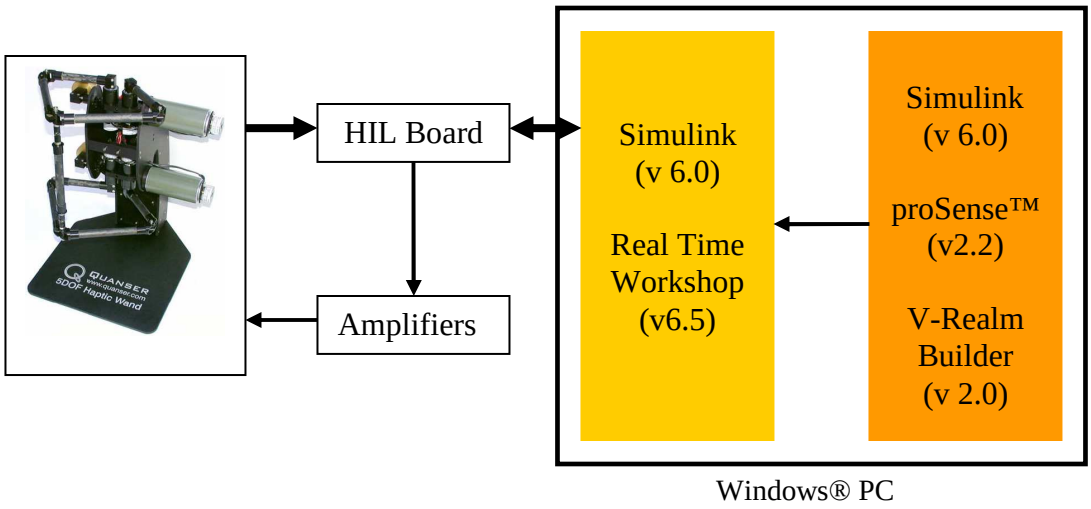

Figure 6: Haptic Wand high level system diagram.

3.1.2 System parameters

The workspace of the wand is 480mm × 250mm × 450 mm in the x, y and z directions, respectively. The device is first calibrated by placing the end-effector in the calibration jig. Once this is done and the calibration program executed, the device is calibrated at [0, 0, 123] mm. There is a non-zero value in the z-direction because of the inherent translation of the jig. Based on this initial calibration value, the workspace boundaries are shown in Table 1.

Table 1: Haptic Wand workspace parameters.

Translation along X	± 240 mm
Translation along Y	85 −335 mm
Translation along Z	- 215 −235 mm

These values are to be noted because of their part in the placement of haptic objects and scaling and resolution considerations. Roll and Pitch ranges are ±85°and ±65° respectively. Also of interest are the maximum continuous forces exerted on the wand.

Table 2 lists important force and torque values for the Haptic Wand. The system disables the amplifiers if these values are exceeded. These values play an important role in the design of haptic objects and their assigned haptic properties like stiffness, friction, etc. A well designed haptic environment will take into consideration the above listed values for optimum design and realistic feel [21]. For a complete listing of Haptic Wand parameters, please refer to the Wand's manual [21].

Table 2: Haptic Wand force parameters.

Maximum Continuous Exertable Force Along X	2.3 N
Maximum Continuous Exertable Force Along Y	2.1 N
Maximum Continuous Exertable Force Along Z	3.0 N
Maximum Continuous Exertable Torque About X	230 N.mm
Maximum Continuous Exertable Torque About Y	250 N.mm

3.1.3 Kinematics

The kinematic modeling equations for the Haptic Wand can be found in the modeling worksheets provided by Quanser Inc. The forward kinematics, inverse kinematics and velocity kinematics are derived from robotics principles. For detailed derivations please refer [24]. These equations are also provided as ".c" and ".m" files to be readily used. The dynamics of the system were not publicly available at the time of writing of this thesis.

3.2 3D Space Notation Overview

The haptic interface device used for our haptic applications is the Quanser 5 DOF Haptic Wand. From the name itself one can infer that it can reflect forces and/or torques in five degrees of freedom. The 5 DOF Wand is capable of rendering torques in two degrees of freedom in addition to forces in three degrees of freedom. In Figure 7, the Haptic Wand is shown with a coordinate frame attached to the movable input stylus, i.e., the "Wand". The reader should note that the z-axis is pointed outward, that is, toward the point of view of the user. The direction of torques are about the x and z axes according to the right hand rule. It

should also be noted that positive directions of torques are assigned according to the right hand rule. Key parameters for understanding the operation of the Haptic Wand are presented the "System Parameters" section.

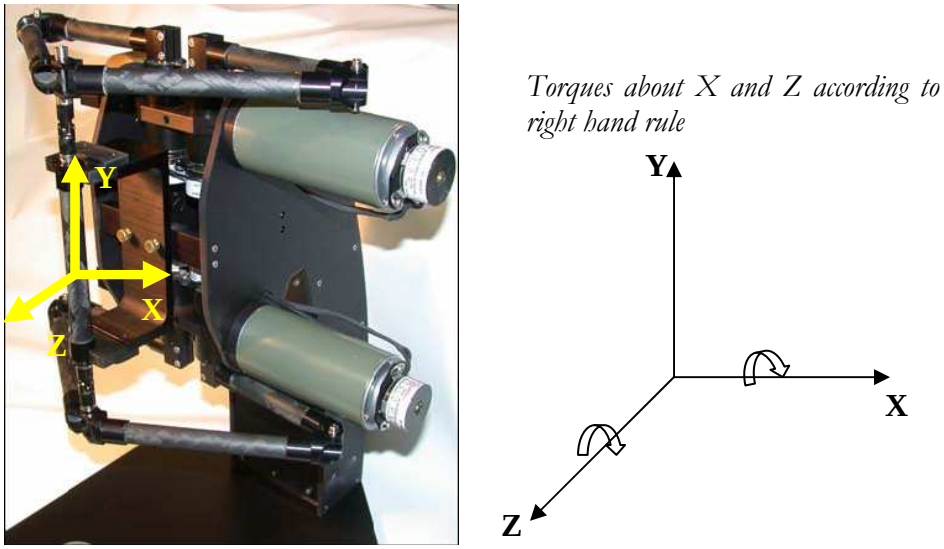

Figure 7 : Haptic Wand with coordinate frame attached to the end effector.

For the purpose of analyzing the Haptic Wand and its workspace, two frames of reference are defined in Figure 8:

1. the **Inertial** Frame (or "world" frame), denoted by **[I]**
2. the **Wand** Frame (or end-effector frame), denoted by **[W].**

In general robotics terminology, the actual "wand" portion of the device called is the end-effector for the robot. The end-effector has its own frame of reference and axes associated with it. As the end-effector is moved, so do the axes move. This is known as the translation of the end-effector axes. The end-effector axes are also capable of rotating following the orientation of the wand.

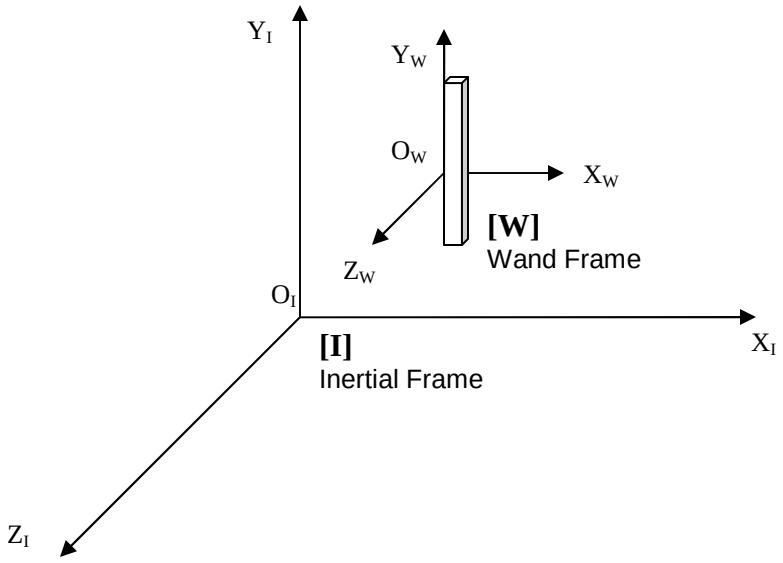

Figure 8 : Inertial and Wand coordinate frames of reference.

In Figure 8, the set-up of the two systems of axes are shown. The inertial frame, **[I]**, is fixed on the base of the wand and the intersection of the three individual lines is assigned the 3D coordinate [x=0, y=0, z=0]. This is called the origin of the frame, and will be denoted by $\mathbf{O_I}$. In the following sections, the relationship between these two frames of reference and equations that connect them will be examined.

3.2.1 Notation for Wand Frame

Having defined the existence of an independent frame for the wand, we now derive equations for representing a point vector in this frame and relationships with the inertial frame. The translation of a point from the origin in any three-dimensional frame is given by a vector with three components. The translation vector is usually denoted by ${}^{I}T$

(notation is used throughout this work) is then

$$^{I}\mathbf{T} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

This notation also specifies that the translation vector is relative to the inertial frame of reference (denoted by the superscript I on the left of **T**). As the wand (end-effector) moves in the workspace of the device, the wand frame also moves accordingly. The origin of the wand frame, which was defined to be variable during motion, represents the translation of the wand. We consider the center point of the wand to be the origin for the wand frame. Accordingly, the translation vector of the center of the wand, in the inertial frame gives the origin of the wand frame, $\mathbf{O_{W}}$..

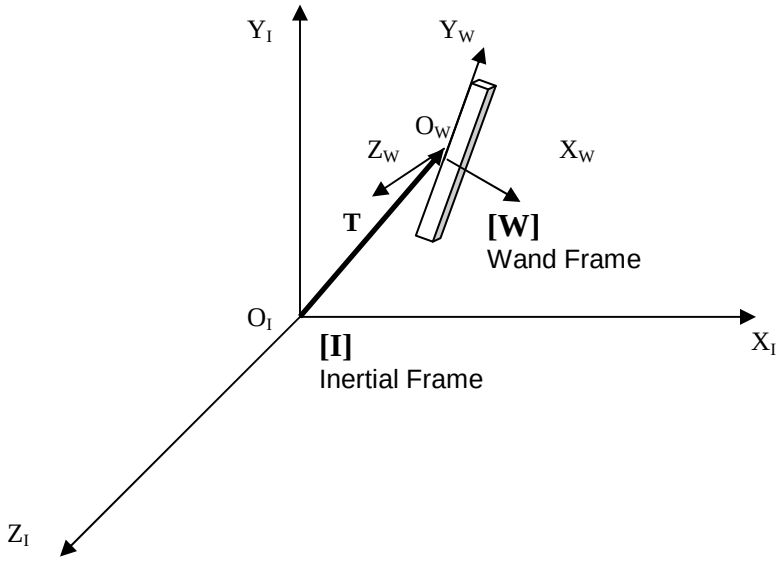

Figure 9 : Rotated Wand Frame.

It is also necessary to formalize the idea of a frame. It is generally agreed upon that a frame is an entity with four vectors giving, position and orientation information as

$$^{W}\mathbf{I} = \{[\mathbf{R}]_{3\times 3}, [\mathbf{T}]_{3\times 1}\}$$

Here, the frame, ${}^{w}\mathbf{I}$, is described by a 3×3 Rotation matrix, **R**, and a 3×1 Translation matrix, **T.**

3.2.2 Rotation of the Wand Frame

Until this point, effort was made to explicitly define the wand frame and to know its relative position with respect to the inertial frame. If the orientations of both inertial and wand frames are the same, then the only distinction metric between the two would be the translation of the origin in the inertial frame. The wand frame not only translates with the wand, but also orients with the rotation of the wand. In Figure 10, the wand is rotated along the z-axis. The wand frame according orients accordingly, the x-axis and y-axis rotate in the x-y plane.

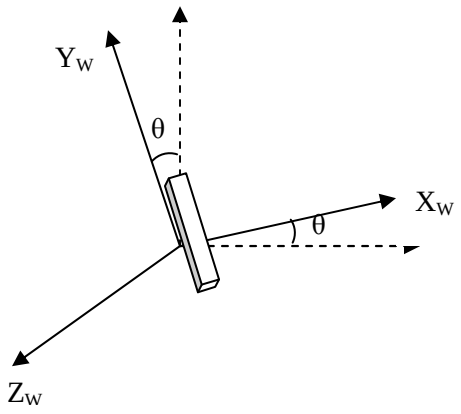

Figure 10 : Wand frame rotated about the z-axis.

Because of the significance of expressing a vector or a point with respect to different frames, an explanation for conversion from one frame to another is presented.

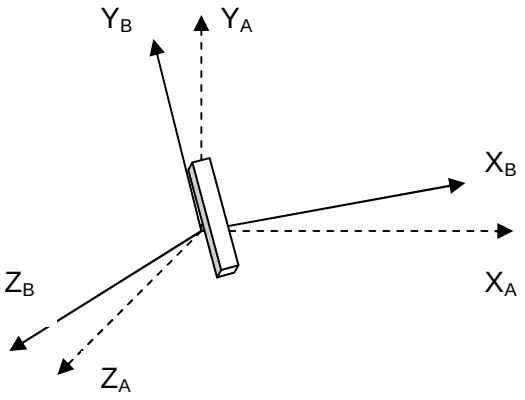

Figure 11 : A frame rotated about all three coordinate axes.

In Figure 11, it is assumed that the two frames, A and B, are described by two sets of 3-dimensional unit vectors. All three unit vectors are mutually perpendicular to each other. Let X_B, Y_B, Z_B , be unit vectors describing the frame B. In order to represent a vector in frame B with respect to frame A, we need to find the representation of each of these unit vectors in frame A

Let vectors ${}^{A}X_B$, ${}^{A}Y_B$ and ${}^{A}Z_B$ represent these transformed vectors. Recall the notation denotes A as the new axes and B as the old axes; the vectors are sought to be defined in frame A. A 3×3 matrix is built, the columns of the matrix containing the aforementioned vectors

$$ {}^{A}\mathbf{R}_B = [{}^{A}\mathbf{X_B}, {}^{A}\mathbf{Y_B}, {}^{A}\mathbf{Z_B}]; $$

$$ {}^{A}\mathbf{R}_B = \begin{bmatrix} \mathbf{X_B}\cdot\mathbf{X_A} & \mathbf{Y_B}\cdot\mathbf{X_A} & \mathbf{Z_B}\cdot\mathbf{X_A} \\ \mathbf{X_B}\cdot\mathbf{Y_A} & \mathbf{Y_B}\cdot\mathbf{Y_A} & \mathbf{Z_B}\cdot\mathbf{Y_A} \\ \mathbf{X_B}\cdot\mathbf{Z_A} & \mathbf{Y_B}\cdot\mathbf{Z_A} & \mathbf{Z_B}\cdot\mathbf{Z_A} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

Some properties of the rotation matrix are listed below; the mathematical proofs for these properties can be found elsewhere. [25]

1. Rotational matrices are orthogonal: the inverse of the rotation matrix is its transpose.

$$[{}^{A}\mathbf{R}_{B}] = [{}^{B}\mathbf{R}_{A}]^{-1} = [{}^{B}\mathbf{R}_{A}]^{T}.$$

2. The column vectors of ${}^{B}\mathbf{R}_{A}$ are of unit length and also mutually orthogonal. The components of the direction matrix (or rotation matrix) are also referred to as direction cosines because the dot product between two vectors yields the cosine of the angle between them [25].
3. The determinant of rotation matrices are ±1.

3.3 The Axis Angle Representation of Rotation

Among the various forms of representing rotation in frames, the axis-angle representation is commonly used. Rotation in this notation is described about a unit vector, $\mathbf{k} = \begin{bmatrix} k_x & k_y & k_z \end{bmatrix}$ and the magnitude of rotation is described by an angle θ (Figure 12).

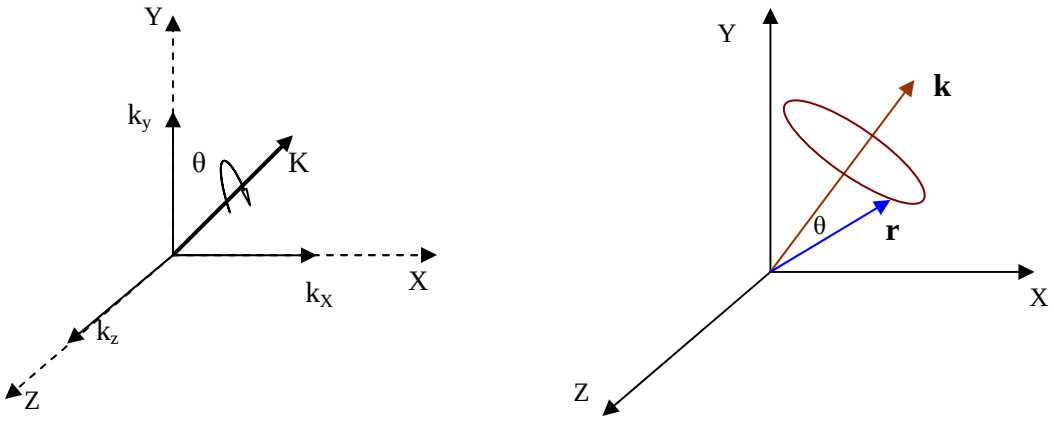

Figure 12 : Axis Angle Representation of rotation around vector k.

It is general practice to represent rotation is axis angle form as ${}^{A}\mathbf{R}_{B}(\mathbf{k},\theta)$ (the orientation of frame B relative to frame A, rotated about the unit vector **k** by an angle θ, **Error! Reference source not found.**, Figure 13). Detailed description of this representation is presented in [25] [26].

3.3.1 Derivation of Matrix from Axis Angle Representation

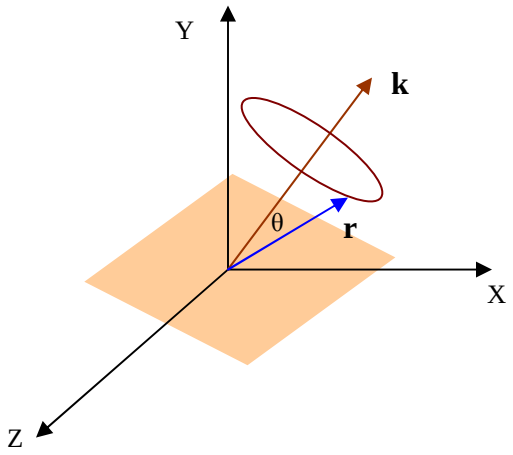

Figure 13 : Axis angle representation : Plane of rotation passing through origin.

From the definition of axis angle representation, rotation is defined about the rotation axis (a vector, **k**). In figure 11, θ is the measure of rotation, the direction given by the right hand rule. The locus of all θ is shown as a circle. The goal of this section is to represent a rotation given in axis-angle parameters in rotation matrix, R, form. A step-by-step approach is followed for this purpose. For a more thorough treatment of this topic, the reader is referred to [27] from which this procedure is credited.

Step 1: A plane is defined as containing the locus of angles of rotation. This plane does not pass through the origin.

Step 2: For ease of finding equations that describe the plane, it is shifted such that it now passes through the origin.

Step 3: It is now sought to find equations that describe the circle in this shifted plane. Each vector (3D point) on the circle can be represented by two basis vectors. The primary goal is to find these basis vectors, $\mathbf{B_1}$ and $\mathbf{B_2}$.

Step 4: The second basis vector is calculated by making use of the definition of cross product. $\mathbf{k} \times \mathbf{r}$ will produce a vector perpendicular to both **k** and **r**.

$$\mathbf{B_2} = (\mathbf{k} \times \mathbf{r})$$

The first basis vector is found by projecting the vector **r**, onto the shifted plane. By definition this vector will be perpendicular to k and $basis_2$.

$$\mathbf{B_1} = \mathbf{B_2} \times \mathbf{r}$$

$$\mathbf{B_1} = (\mathbf{k} \times \mathbf{r}) \times \mathbf{r}$$

This is sufficient to define the circle with $\mathbf{B_1}$ and $\mathbf{B_2}$ as:

$$\mathbf{B_1 Cos\theta + B_2 Sin\theta}$$

Step 5: Define the circle using shifted plane information

We now define **r'** as the transform of **r**. To "get back" the undisplaced circle, we add shift offset to the circle equation.

$$\mathbf{o = r - B_1}$$

$$\mathbf{r' = o + B_1 Cos\theta + B_2 Sin\theta}$$

It is sought to mathematically, reduce the above equation in terms of k, **r** and **θ.**

Substituting for o in r':

$$\mathbf{r' = r - B_1 + B_1 Cos\theta + B_2 Sin\theta}$$

Rearranging and modifying,

$$\mathbf{r' = r + (1 - Cos\theta)(k \times k \times r) + Sin\theta(k \times r)}$$

Step 5: The representation thus far has been in vector algebra. It will now be converted into matrix algebra using standard theorems. An important property is that

$\mathbf{X = A \times B}$ can be written as $\mathbf{X = [A\hat{}]B}$

where, $\mathbf{A\hat{}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$

Using this notation, the equation for **r'** can be rewritten as:

$$\mathbf{r' = [I + (1 - Cos\theta)(A\hat{})^2 + Sin\theta(A\hat{})]r}$$

This is in the form, **r' = [R],** where, R is the rotation matrix,

$$\mathbf{R = I\cos\theta + (1\text{-}\cos\theta)} \begin{bmatrix} k_1^2 & k_1k_2 & k_1k_3 \\ k_2k_1 & k_2^2 & k_2k_3 \\ k_3k_1 & k_3k_2 & k_3^2 \end{bmatrix} \mathbf{+ \sin\theta} \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}$$

On simplification, $R = \begin{pmatrix} tx^2 + \cos\theta & txy - z\cos\theta & txz + y\sin\theta \\ txy + z\sin\theta & ty^2 + \cos\theta & tyz - x\sin\theta \\ txz - y\sin\theta & tyz + x\sin\theta & tz^2 + \cos\theta \end{pmatrix}$.

The purpose of deriving the R matrix is to facilitate representation of orientation in one format throughout the analysis of the model. In this work, the practical implementation software used the axis angle notation whereas in order to design a point generating algorithm, wherein multiple points would be rendered on the wand, the matrix

representation was needful. This above derivation forms the basis for the Point Generation subsystem discussed later.

3.3 5DOF Rendering Algorithm

The Quanser 5DOF wand is capable of generating forces in three dimensions and torques in two dimensions (for a quick graphical explanation refer to Figure 7). It has also been explained that the proSense software used to design haptic models and control the device is capable of linear force rendering only (at the time of writing this thesis). In order to make use of the full potential of the device, it was necessary to design an algorithm in the present software architecture that renders forces and torques.

To further explain the motivation behind developing a 5DOF rendering algorithm, the simplistic 3DOF force rendering only model is examined. Figure 14 shows the Simulink model for single point system. This model has the "HVR World" block as the haptic and graphic rendering subsystem (for further details, please refer to Chapter 2). The output Cartesian Position from the Wand block (labeled "Quanser 5DOF Haptic Wand" in Figure 14) is input to the Rendering system. The HVR World block solves for collision detection and response with the virtual world and calculates forces based the actual position of the wand and collisions in the virtual world.

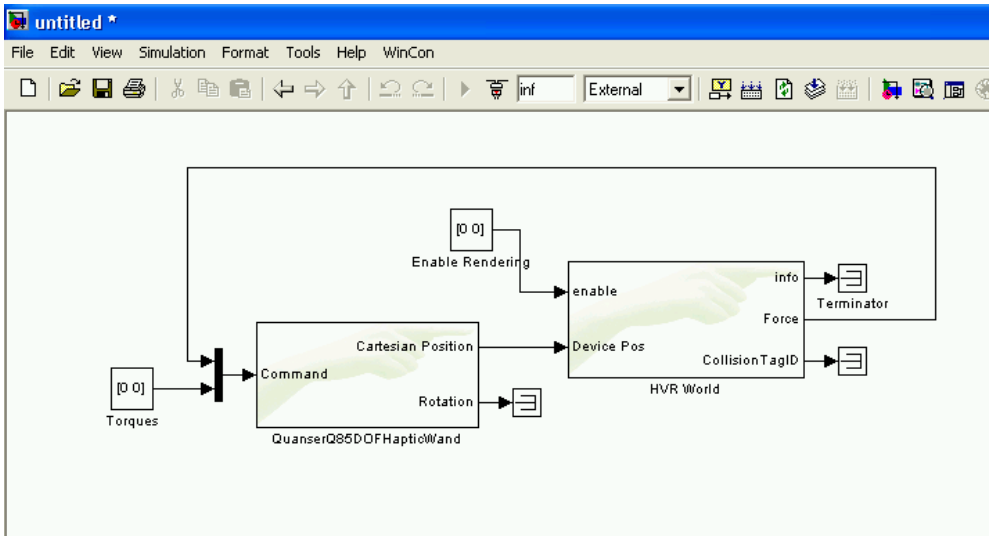

Figure 14 : Simulink diagram of a haptic model for the Wand.

Because of the single point nature of proSense™ software, forces are calculated at that single position point. (The position coordinates give the position of the center of the Wand in inertial frame)

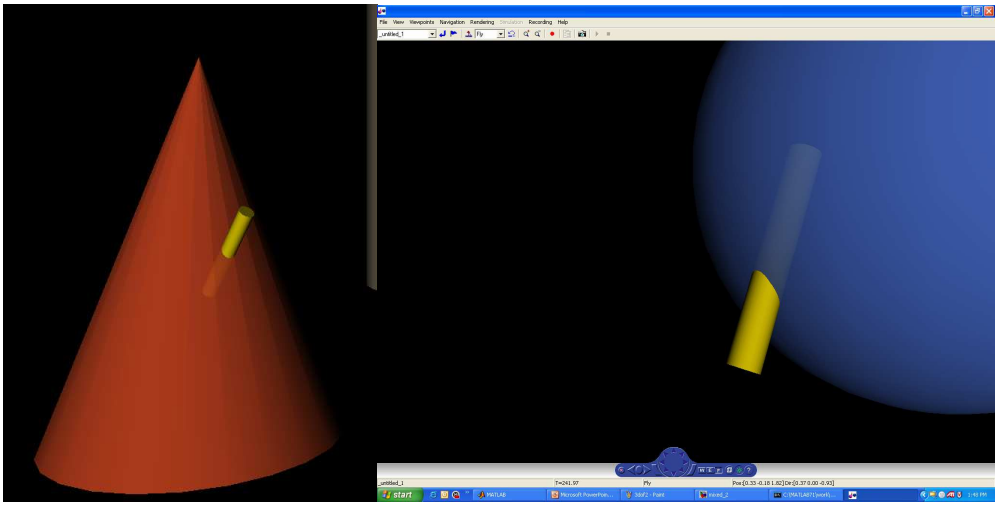

Figure 15: Single point haptic rendering.

Figure 15 demonstrates the unrealistic rendering using the single point interaction method. In both models, the wand is represented by a cylinder in keeping with the shape of the wand.

However, since the point of contact for rendering is the center of the wand, the virtual world "avatar" is able to penetrate into the object without any force rendering. Also, once the center of the avatar is on the surface of the haptic object, it is possible for the ends to penetrate into the object without haptic feedback. This gives an unrealistic haptic rendering caused due to single point based rendering. The cylindrical shaped Wand is being treated as a single point.

In response to the above problem, a multiple-point based rendering algorithm is proposed, capable of reflecting forces at multiple locations on the avatar and therefore, the Wand itself. Also, from these points, we will be able to calculate torque based on principles of rigid body physics. The proposed model, therefore, can be used for 6 DOF rendering. However, because the Quanser Haptic Wand available is only capable of 5 DOF rendering, we will test for results accordingly. The overview for the model is shown in Figure 16 where three subsystems are shown.

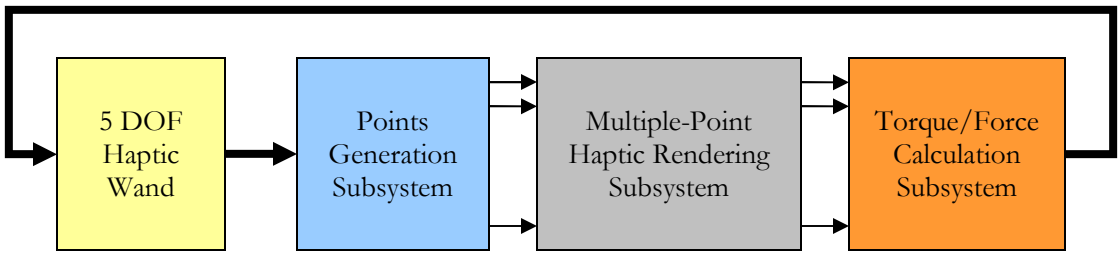

Figure 16: Multi-point haptic rendering algorithm system diagram.

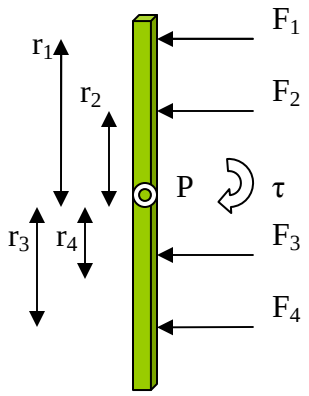

Figure 17: Derivation of torque for two-point model.

Figure 15 shows the forces and distances from the center of a hypothetic wand to calculate torque for a four point rendering system. The Torque subsystem takes in the distance and force data and applies the torque law as shown above. It can also be noticed that $\tau = \tau_x + \tau_y + \tau_z$ is a vector with three dimensions, each specifying torque about that particular axes. Since torque can be rendered only about the x and z axes, τ_x, τ_z are taken from τ.

For a two-point system, the rendering system will output two forces based on their respective locations and collisions. The torque can be calculated from these two forces. Torque, τ, is calculated with respect to the center of the wand. The distances r_1 and r_2 are specified by the Point Generation subsystem. (This can be made variable depending upon the resolution needed for haptic rendering).

The following three sections describe each system of the 5 DOF haptic rendering algorithm in detail.

3.4.1 Point Generation Subsystem

The inputs to this block are the Cartesian Position of center of wand and Rotation of the end-effector (wand). The purpose of this block is to generate multiple points on the wand, displaced by fixed distances from the center of the wand. These points will then "follow" the rotation of the wand. Throughout this thesis the number of points chosen is two. The convention of points is: "Top Point" (representing the top end of the wand) and "Bottom Point" (representing the bottom end of the wand). It is left to the discretion of the user and the force resolution desired to set these values.

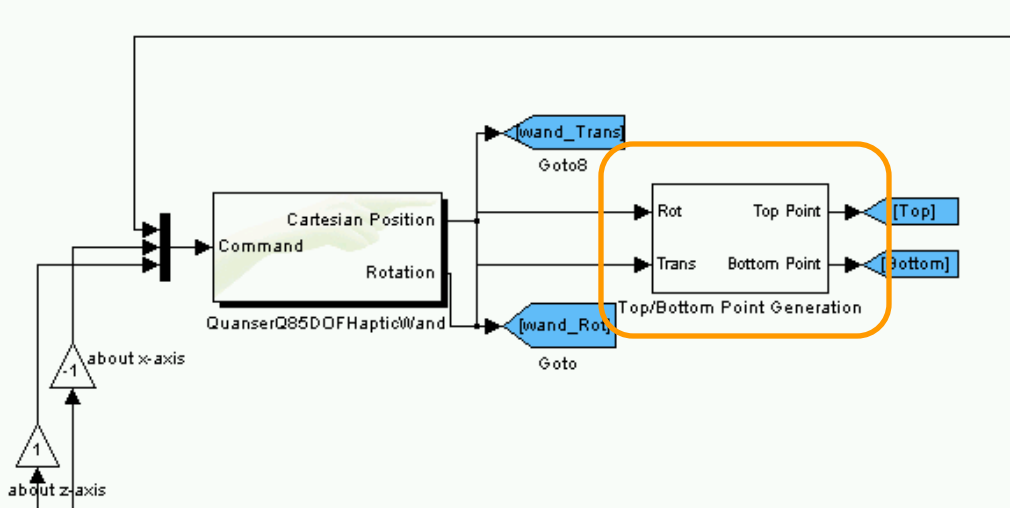

Figure 18: Simulink diagram of Point generation subsystem for two points.

Once the number of points is set and their distance with respect to the center is calculated, the purpose of the block is to generate these points on the wand avatar. These points should follow wand translation and rotation. It should also be noted that rotation of the wand is given in axis-angle form in the end-effector frame (wand frame). The equation to find the cartesian position of the point each point, displaced by a translation, T_i , is given by

$$P_i = [R]Ti + P_w$$

$$R = \begin{pmatrix} tx^2 + \cos\theta & txy - z\sin\theta & txz + y\sin\theta \\ txy + z\sin\theta & ty^2 + \cos\theta & tyz - x\sin\theta \\ txz - y\sin\theta & tyz + x\sin\theta & tz^2 + \cos\theta \end{pmatrix}$$

$$t = (1 - \cos\theta)$$

The values of x, y, z, θ are found from the input rotation vector. P_W ($=\begin{bmatrix} x_w & y_w & z_w \end{bmatrix}$) is the Cartesian position of the center of the wand. The vector T_i specifies the ith point in the wand coordinate frame. The length of the end effector handle is 0.159m [21]. It can, therefore, be approximated that the top most point on the wand handle is at [0, 0.08, 0]. Similarly, the bottom most point of the wand is approximated as [0, -0.08, 0]. It should be carefully noted that this is true only when the wand center position is fed to the subsystem without any gain. This varies with the resolution of position of end effector. It is highly suggested that for best use of this algorithm that users arbitrarily choose point locations based on practical observation and resolution and scaling of 3d world parameters. In the present model, we will consider only two points due to present limitation insystem performance and rendering tools. The equations for top and bottom point are:

$$^{I}P_{top} = [R]T_{top} + P_w$$

$$^{I}P_{bottom} = [R]_{bottom} + P_w$$

The outputs for the wand are the Cartesian positions of the specified multiple points. Throughout this thesis, the number of points chosen is two for aforementioned reasons. Having more than one point also enables the calculation of torques for rendering which will be dealt in detail later. A portion of the Simulink model is shown in figure 17.

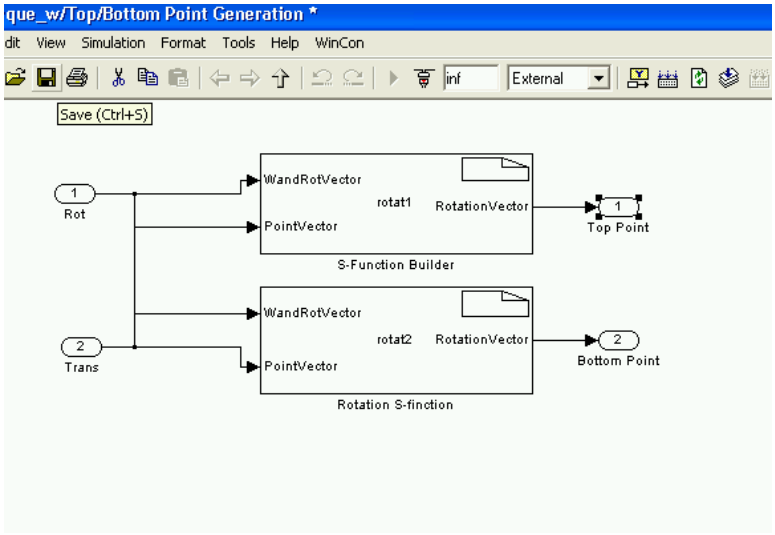

Figure 19: Point Generation subsystem.

3.4.2 Multiple-point Haptic Rendering Subsystem

It is assumed in this work that for rendering of forces there exists an algorithm that inputs position in Cartesian coordinates and outputs forces (proSense™ for MATLAB is used in this work). For the purpose of working with the Wand, proSense software qualifies the above condition. The proSense library of haptic blocks contains a rendering block performing both haptic and graphic rendering. This block accepts a VRML file, parsing it for nodes. Nodes in VRML contain the 3D parameters of object including shape, size, color, translation and rotation. proSense™ also features the "hapticizing" (3D objects are assigned haptic properties) of this 3D file with an easy to use GUI for specification of haptic parameters like stiffness, friction, etc.

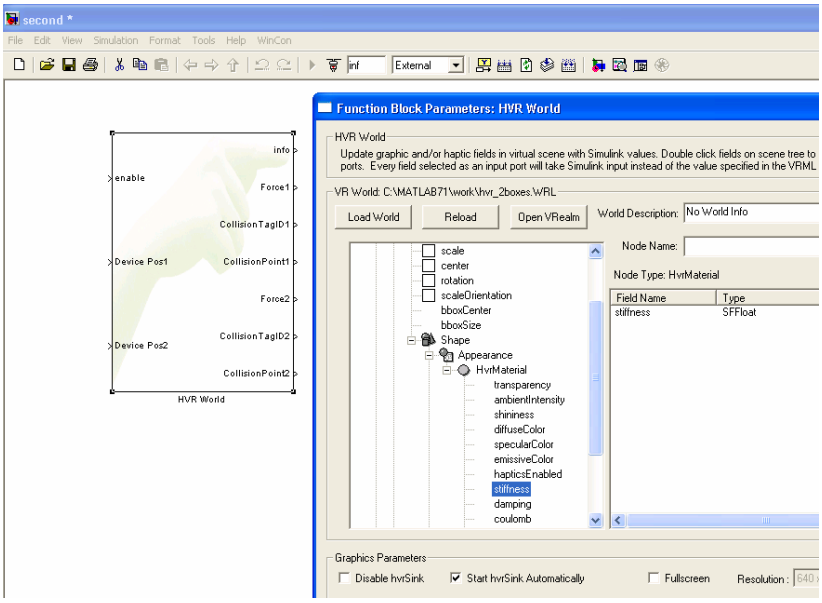

Figure 20: Simulink diagram of HVR World (haptic parameters).

The Haptic Rendering subsystem is responsible for haptic (and graphic) calculations. It is sometimes necessary to reverse-engineer and specify the multiple point locations (described in the previous chapter), resolution of position (gain in Wand center position), etc, based on the adjustments in this subsystem. For example, if the 3D world to be rendered is a surgical environment where greater precision is required, then the resolution of position will be large. There will also be a force adjustment (divided by the gain in position) commensurate with position resolution. The algorithm and method chosen for haptic rendering can also be of critical importance in force and position critical applications. However, since this topic is not the focus of this section, it is not discussed in detail here.

3.4.3 Torque Application subsystem

The goal of this algorithm is to facilitate 5DoF rendering; it is hence necessary to compute torques along with the forces. The reader will recall that the haptic device used is capable of torque rendering in two dimensions. In this section, the torque about the x-axis and the torque about the z-axis is computed and rendered.

A "first principles" approach is taken for torque calculation. The wand is modeled as a cylindrical rod, as shown in Figure. Consider two forces acting at the end points of the rod. Based on the direction of the forces, a torque is applied to the rod. For example, if the forces are in opposite directions, a rotational torque is produced. To calculate the magnitude and direction of torque a pivot point must be established. In many cases, the pivot point is the center of mass of the body. All torques are calculated about this point. It is well know that if a force, F, acts at a s=distance, r, from the pivot point, torque is calculated as $\tau = r \times F$. Since torque is a vector, the direction is described by the vector product (perpendicular to r and F). Similarly, if there are multiple forces and hence, multiple points of contact, the resultant torque is given by

$$\tau = \sum (r_i \times F_i).$$

3.5 Simulation Results

This section provides some test results from the 5 DOF haptic rendering algorithm implemented using two points, one at each end of the wand. The first section briefly describes model building and setup. Following this, the experiment data is described and the data is analyzed and presented in the graphical form. Finally, conclusions are drawn and some improvements to the model are suggested as future work.

The 5DOF haptic rendering algorithm model is built using Simulink, proSense toolbox for MATLAB, and V-Realm builder. In this example, a three dimensional cube is designed and haptic properties are assigned. The Haptic Wand is virtually rendered as a two-point model, having a top and a bottom point separated by equal distance from the center. Figure 21 shows the cube and identifies the two points (red and green balls) used to implement the torque model. The goal of this experiment is to collect data when either point is touching the wall of the cube as well as both points. For simplicity, only one face of the cube is considered for analysis (the face parallel to the y-axis in inertial frame). Data is then analyzed to check for accuracy of torques.

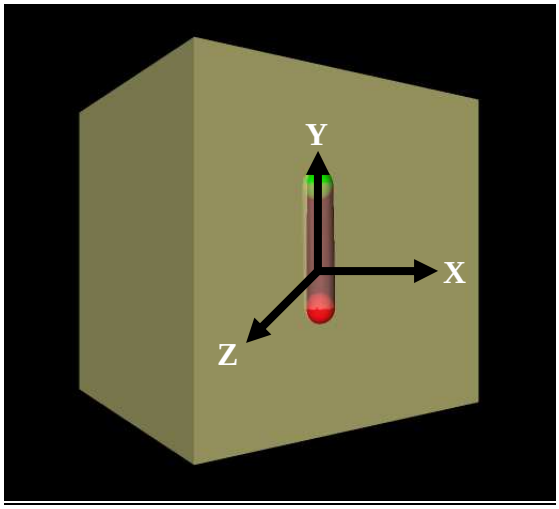

Figure 21: Cube model for 5 DOF algorithm testing.

The Simulink model contains three main subsystems. The point Generation subsystem is designed to create two points: top point and bottom point. These points represent the top and bottom of the Haptic Wand. The mechanism of this block was discussed earlier. For the Haptic Rendering subsystem, the VRML world containing the cube is "hapticized". This is done using proSense™'s "HVR World" block. The block is set to dual configuration mode, thus enabling two input positions for top and bottom points. Finally, the Torque calculation subsystem calculates torques based on the theory discussed previously.

The graphical analysis of data collected is shown below. Figure 23 contains force data. Each component force is plotted for both top and bottom forces. It can be noticed that forces are significant only in the z component. This is because the wand is pushed against the face parallel to the z-axis. The second plot shows the torque about the x axis (measured in N-mm meters). In Figure 22, the three cases for analysis of torque are shown. In cases one and two, only one point of the wand is with the haptic cube. In case three, both points of

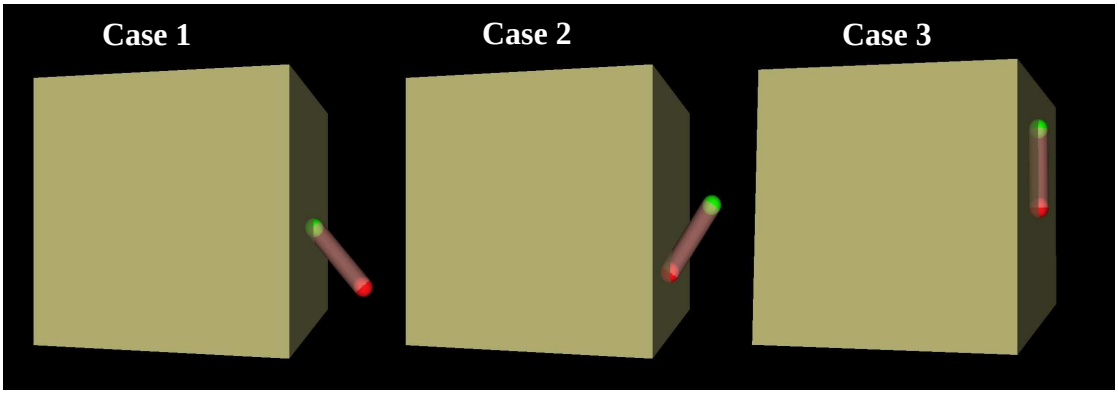

Figure 22: Three scenarios for torque analysis.

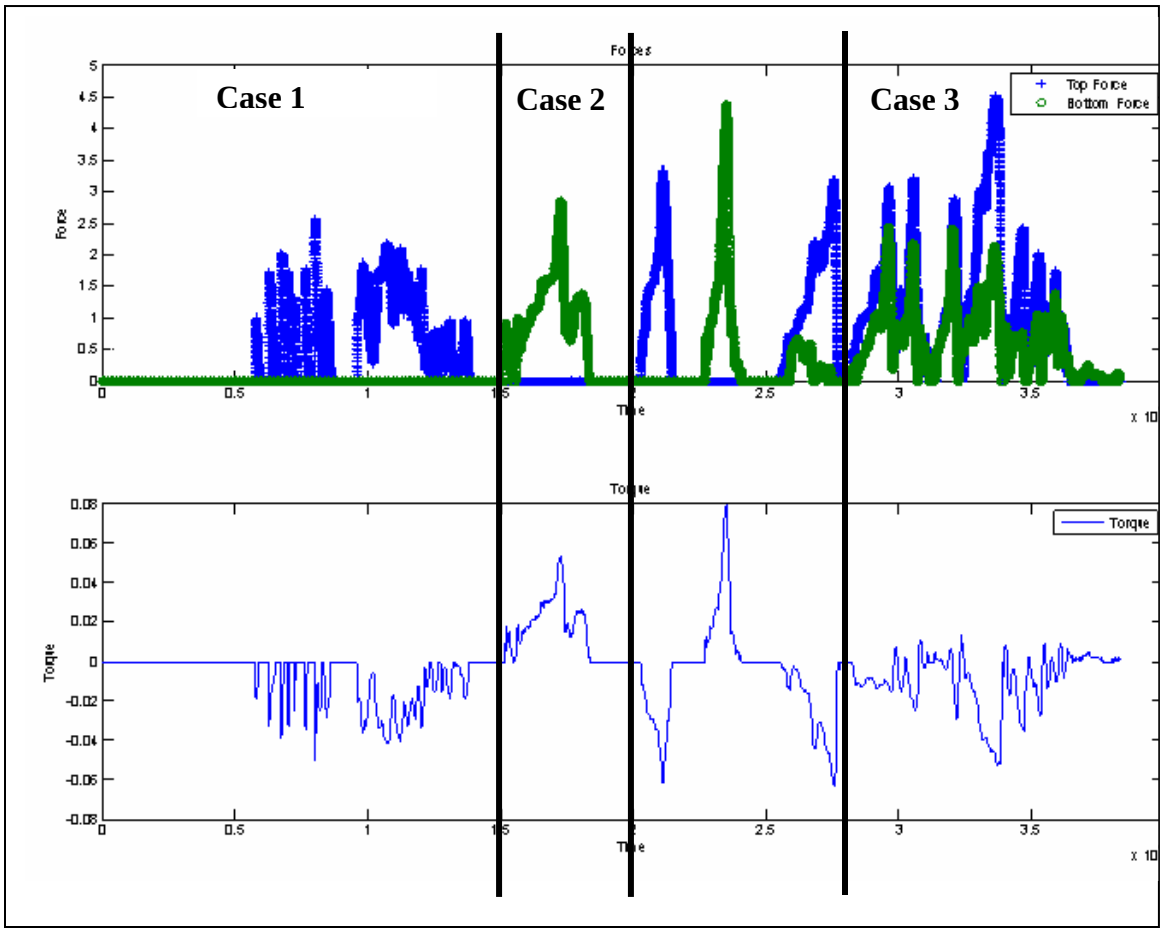

Figure 23: Force and torque comparison for 5 DOF algorithm.

the wand are in contact with the haptic cube simultaneously. Force and torque results for their respective cases are shown in Figure 23. When the top point is in contact with the haptic cube, the result is a negative torque about the x-axis. Conversely, when the bottom point is in contact with the haptic cube result is a positive torque about the x-axis. When both points of the wand touch the haptic cube simultaneously, the results are show in Figure 23.

Figure 24 shows torques about each component. It will be noticed that the magnitude of torque is significant only in one direction (about x).

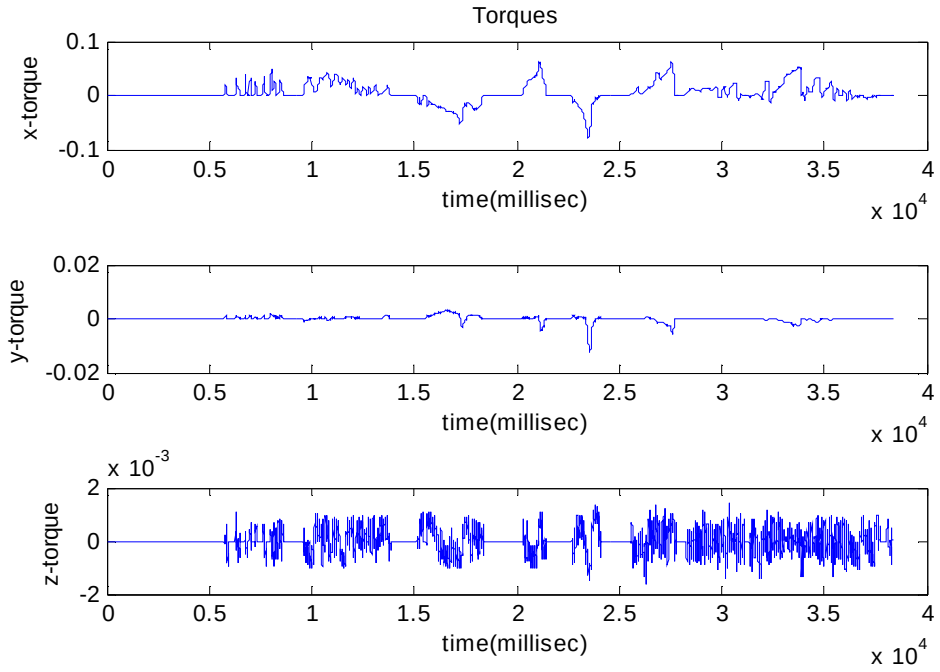

Figure 24: Torques about three axes.

Figure 23 compares torques and forces. It can be seen from this graph that the magnitude and direction of torque are as expected.

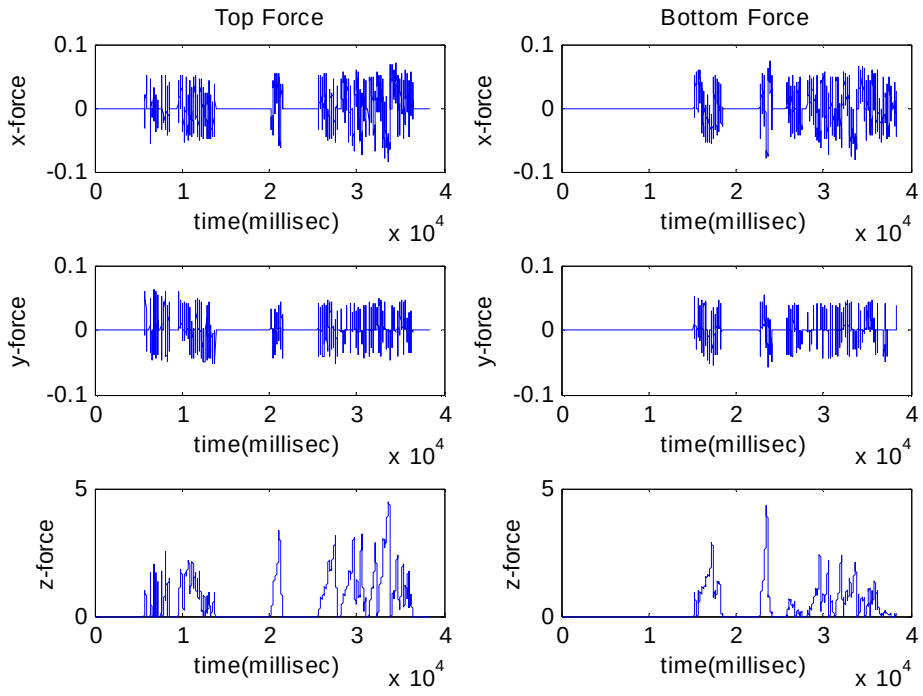

Figure 25: Top and Bottom point forces.

3.6 Conclusions and Future work

The drawbacks of 3 DOF haptic rendering were presented. A 5 DOF haptic rendering algorithm is presented based on rendering multiple points for the end effector (in this case the Haptic Wand) and calculation of torques using the cross product. The model is tested using a proSense model with the Quanser 5 DOF Haptic Wand. Finally, results for torques and forces on the test model are examined. The algorithm is demonstrated to render force and torques in the correct direction and of the correct magnitude..

This work can have a vast number of applications. Haptic exploration is the "feeling" of objects and surfaces with a haptic interface. In this technique, it is important to have multiple degrees-of-freedom and torque rendering is known to especially increase

effectiveness in haptic feeling [28]. Torques can also be used for assistive force feedback techniques such as rehabilitation and writing. For example, elementary school students can be taught handwriting using 5 DOF rendering [29]. Improvements to this model can include rendering the Haptic Wand (any end effector) as more than two points. This will further increase realism in haptic feedback.

CHAPTER FOUR
STUDY OF HAPTIC AND VISUAL FEEDBACK FOR KINESTHETIC TRANING TASKS

This chapter compares the relative effectiveness of visual and haptic feedback training as preparation for kinesthetic navigation tasks like laparoscopic surgery. First, the motivation for conducting the experiment is elucidated along with literature review and a survey of available visio-haptic trainers. In the next section, the experimental setup is presented. The tools and methodology employed to build visual and haptic models for the test experiment are detailed. Data collection methods are explained including the recruitment of human subjects. The penultimate section deals with the analysis of data. Three main hypotheses are tested and discussed. Finally, conclusions drawn from results that include the conclusion that haptic training for kinesthetic learning tasks is better than visual training are presented. Suggestions for future work with applications to laparoscpoic surgery in particular are presented.

4.1 Motivation and Background

Laparoscopic surgery, a Minimally Invasive Surgery (MIS) technique, has seen major advances since its early beginnings in the 1960s. It is performed through small incisions, less than 10mm in diameter, made on the patient. A laparoscope is inserted and the abdominal cavity is inflated. The surgeon uses special instruments and miniature cameras for this procedure. Surgery is performed using the camera view and feel from the instruments. The surgeon, therefore, has to be better trained to use haptic (sense of touch) cues when performing laparoscopic surgery because of the lack of traditional visual cues in this method. Since this procedure requires smaller incisions, there is a significant reduction in hospital stay

and recovery time. Perhaps the most common laparoscopic procedure is gall bladder removal called Cholecystectomy. In this procedure, the gall bladder is drained of bile, cut and removed through small cuts made in the abdominal region [30]. This research experiment focuses on training techniques that may help prepare surgeons to perform laparoscopic surgery.

Due to the advantages in patient recovery time and comparative procedural ease, laparoscopic procedures have become more common. Commensurate with this demand has been the need to train surgeons to perform these procedures. Early computer-based training for laparoscopic surgery consisted of purely visual feedback. These trainers used virtual patients and models, generating realistic visual human anatomy and responses [31] [32]. Recent trainers have been more focused on adding haptic feedback. For example, Marvick, Lango *et al.*, designed a laparoscopic pointer for 3-D image guided surgery [33]. Feintuch *et al.*, in their research showed the effectiveness of haptic feedback for large-scale haptic navigation [34]. Tendick *et al.*, developed a virtual environment tested for visio-haptic training [35], while Cavusoglu *et al.*, developed a haptic Telesurgical trainer [36]. Among commercially available haptic trainers, Immersion Corporation's Medical CathSim Vascular Access Simulator, developed to train nursing students for intravenous procedures, is probably the pioneer. Figure 26 shows the Immersion Corp. laparoscopic trainer which is designed and commercially available to perform laparoscopic training.

Figure 26 : Immersion Corporation's laparoscopic trainer [37].

While these trainers and model have used a combination of visual and haptic feedback mechanisms, this experiment seeks to quantitatively understand the relationship between haptic and visual feedback used for kinesthetic navigation tasks (which include surgical tasks). Kinesthetic navigation may be most important outside the field of view of the laparoscope and thus neglected by many simulators.

4.2 Materials and Methods

Minimally invasive surgery techniques like laparoscopic surgery depend on the surgeons' skill and experience to perform kinesthetic tasks – tasks involving precise limb control. In this experiment we seek to compare the effectiveness of haptic feedback versus visual feedback in preparing subjects for kinesthetic navigation tasks. To illustrate, a black path is shown in Figure 27. If a subject is given a pencil and asked to learn the path using two methods: trace over the path while looking at the path (visual) and close his eyes while the pencil is guided by a hand, the goal is to find which method is more efficient for learning the path. The colored lines would then represent the users attempt to reproduce the path. In this work, the path will be three dimensional. In the following sections the experiment is presented from organization to analysis.

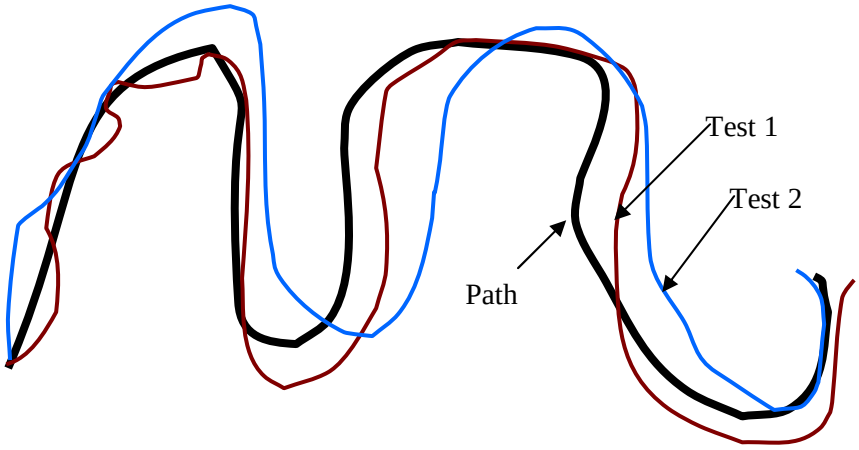

Figure 27: Simple explanation of haptic experiment.

4.2.1 Experiment Design

For the experiment a 3D traceable path needs to be constructed for the Clemson University LaparoWand (see Section 4.3). A virtual world is built with a hollow tube for the purpose of providing a reference path for training and performance evaluation. The walls of the tube provide boundaries within which the user will train to position the avatar. The tube is designed to optimally fit in the Haptic Wand's workspace (Figure 7, Table 1). This three-dimensional tube (Figure 28, Figure 29) is used for training the user to learn a prescribed avatar trajectory. In accordance with the goal of comparing feedback paradigms, two training methods are selected: haptic and visual. Initially, users are randomly assigned to either group.

In haptic training, users train with only haptic feedback from the tube; there is no visual feedback. As users navigate through the tube with the Clemson University LaparoWand, they "feel" forces represented by the walls of the tube. In haptic training, initially users slide along the walls to make a mental map of the tube, ideally the accuracy of this map increases with each training iteration.

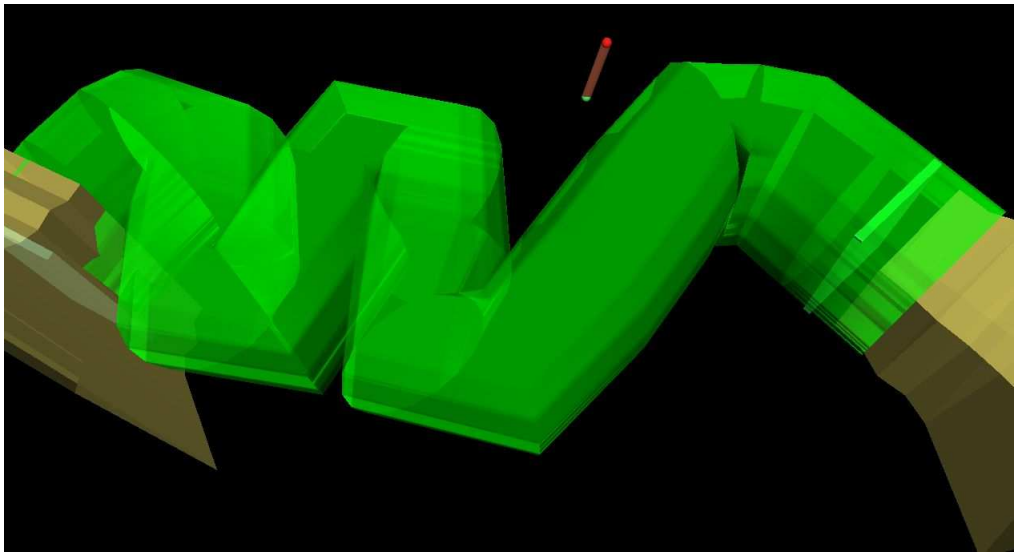

Figure 28: Top view of 3D navigation tube.

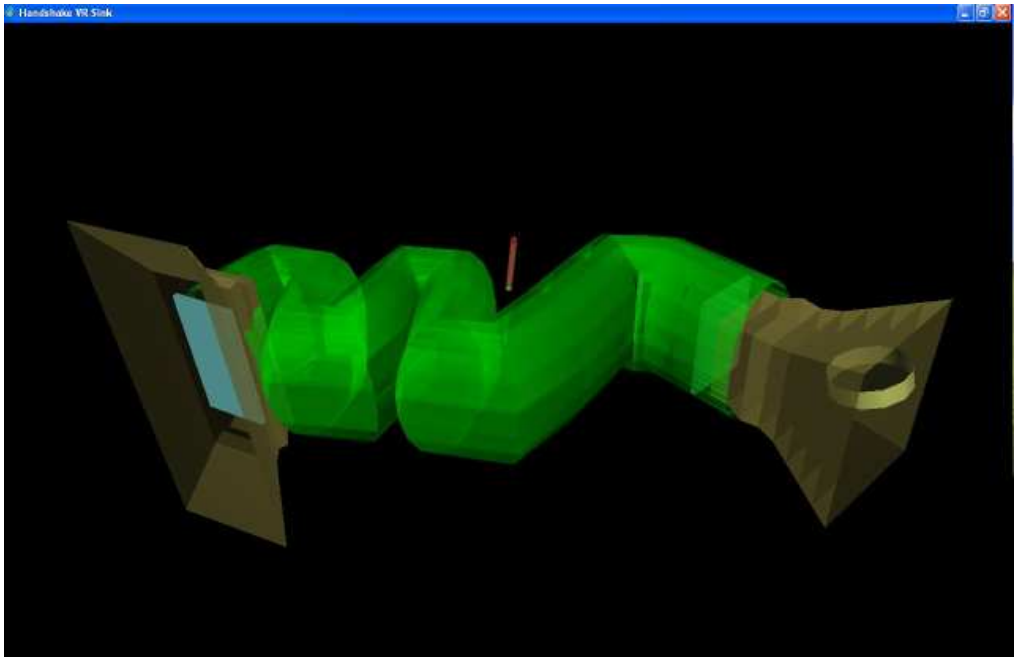

Figure 29: Front view of the experiment environment.

In visual training, users learn the path of the tube only through visual feedback. An overhead screen provides the three-dimensional picture of the tube and avatar position inside the tube in real time during the training phase. Although the Haptic wand is used to navigate through the tube, no haptic feedback is present. When the wand avatar is in contact with the walls of

the tube, the color of the tube changes. Using these visual cues, users learn the path of the tube. It should also be noted that because entry and exit to the tube presented a significant challenge without feedback, visio-haptic "funnels" were provided at the start and end of the tube. These funnels were made available to both training groups. To enter the tube, users will first enter the start funnel which will lead them into the tube. Similarly, the end funnel will guide users out of the tube. Also, start and end "Caps" were provided to signal proper entering and exiting the tube. The color change of the caps signaled proper entering or exit.

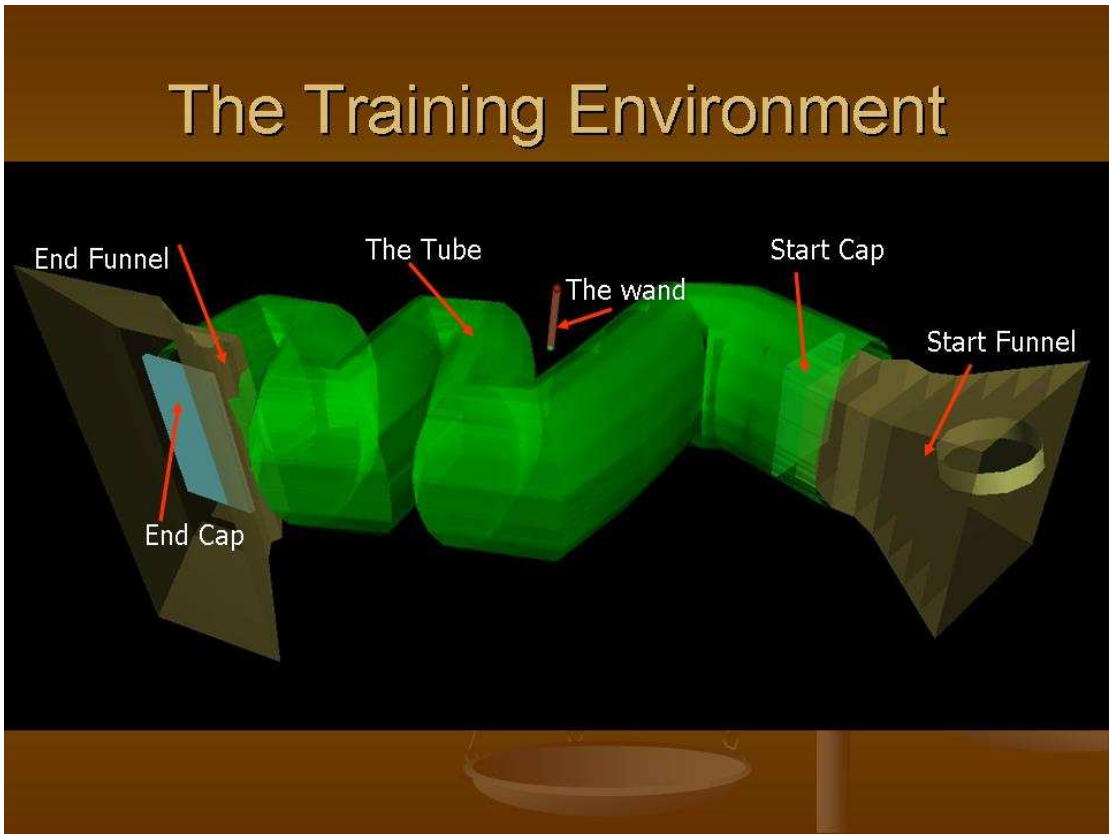

Figure 30: Labeled diagram of virtual world for navigation experiment.

Each training group was provided with two timed training sessions, interactive and non-interactive, totaling a period of seven minutes. At the end of this session, users were invited to test. The goal of testing is for users to trace the path of the tube, learned during their

training, without any feedback. Both training groups tested on the same path. The entry and exit mechanisms (funnels and caps) were still made available to the user (Figure 30, Figure 31) as testing for proper entry is not part of this study. An overhead screen with displaying the funnels and caps but no tube was provided to users. Also, during navigation with the Haptic Wand, no haptic forces were sensed. Data was collected for users during testing for two trials. Data analysis was performed offline once data from all users was collected. A flow chart of the experiment is shown in Figure 32.

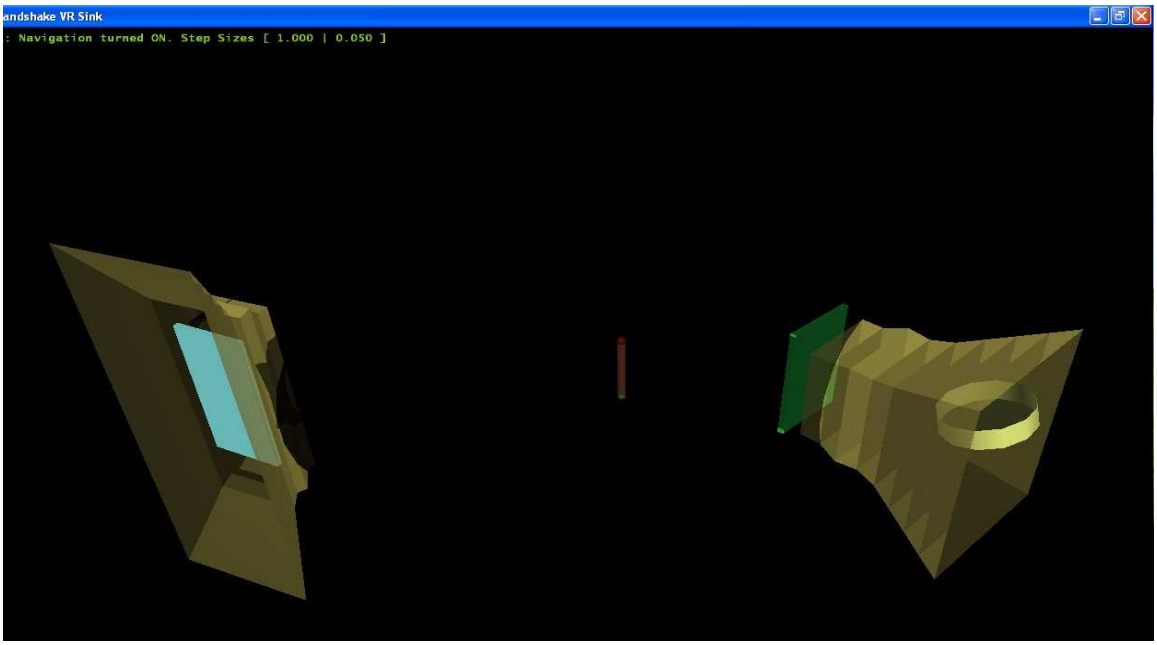

Figure 31: Testing environment.

4.2.2 Data collection

The central hypothesis for this experiment is that the use of haptic feedback to train users for kinesthetic navigation tasks in laparoscopic surgery simulations is more effective than visual training.

The "Null Hypothesis" tested in this experiment is that haptic training is more effective than visual training for the navigation tasks

$$H_o = E_h > E_v .$$

An alternate hypothesis is that haptic training is equivalent or less effective than visual training

$$H_o = E_h \leq E_v$$

Effectiveness, E, is defined as:

$$E = (K_1)(TaskTime) + (K_2)(AverageDeviation)$$

where, K_1, K_2 and K_3 are arbitrary scaling factors chosen to scale units appropriately, and, TaskTime and MaximumDeviation are calculated from real-time measurements collected.

The subjects for this experiment were 10 students at Clemson University. Subjects were evenly divided between male and female and ethnic groups. The test session begins by assigning each participant with a unique identification number. No private information of the participant is recorded. Participants were pre-assigned randomly to either the Haptic or Visual Training group. A one minute introduction was given about the purpose and plan of the experiment. Participants were then asked to read a presentation prepared to guide them through the details of the experiments. Information such as entering and exiting the tube and the philosophy of the experiment were explained. Once this was completed, participants were invited to perform a three minute interactive training session. In this session, they were free to ask questions about any part of the experiment operation and guidance is given by the administrator. After this session ended, participants were shown the test model. This was done so that they would be aware of the environments used for testing. This session lasted for approximately one minute. Following this was the non-interactive training session. Participants trained using their assigned feedback method for a fixed time period of five minutes (Figure 33). No data was recorded during training sessions. After this, participants

were invited to test. Testing was not time constrained. Two trials were requested based on proper exit (*i.e.*, the end cap changes color). Data was recorded only during testing. The measurements collected during test are as shown in Table 3.

Table 3: Data collection parameters during testing.

BottomDeviation	*Bottom Point position – Bottom collision Point*
BottomPoint	*Bottom point translation*
CIDBottom	*Bottom point Collision ID*
CIDTop	*Top point Collision ID*
CPBottom	*Bottom collision point*
CPTop	*Top collision point*
ForceBottom	*Bottom 3d forces*
ForceTop	*Top 3d forces*
Forces	*Sum of Top and Bottom forces*
TopDeviation	*Top point position – Top collision point*
TopPoint	*Top point translation*
WandRotation	*Wand rotation (axis-angle)*
WandTranslation	*Wand center translation*

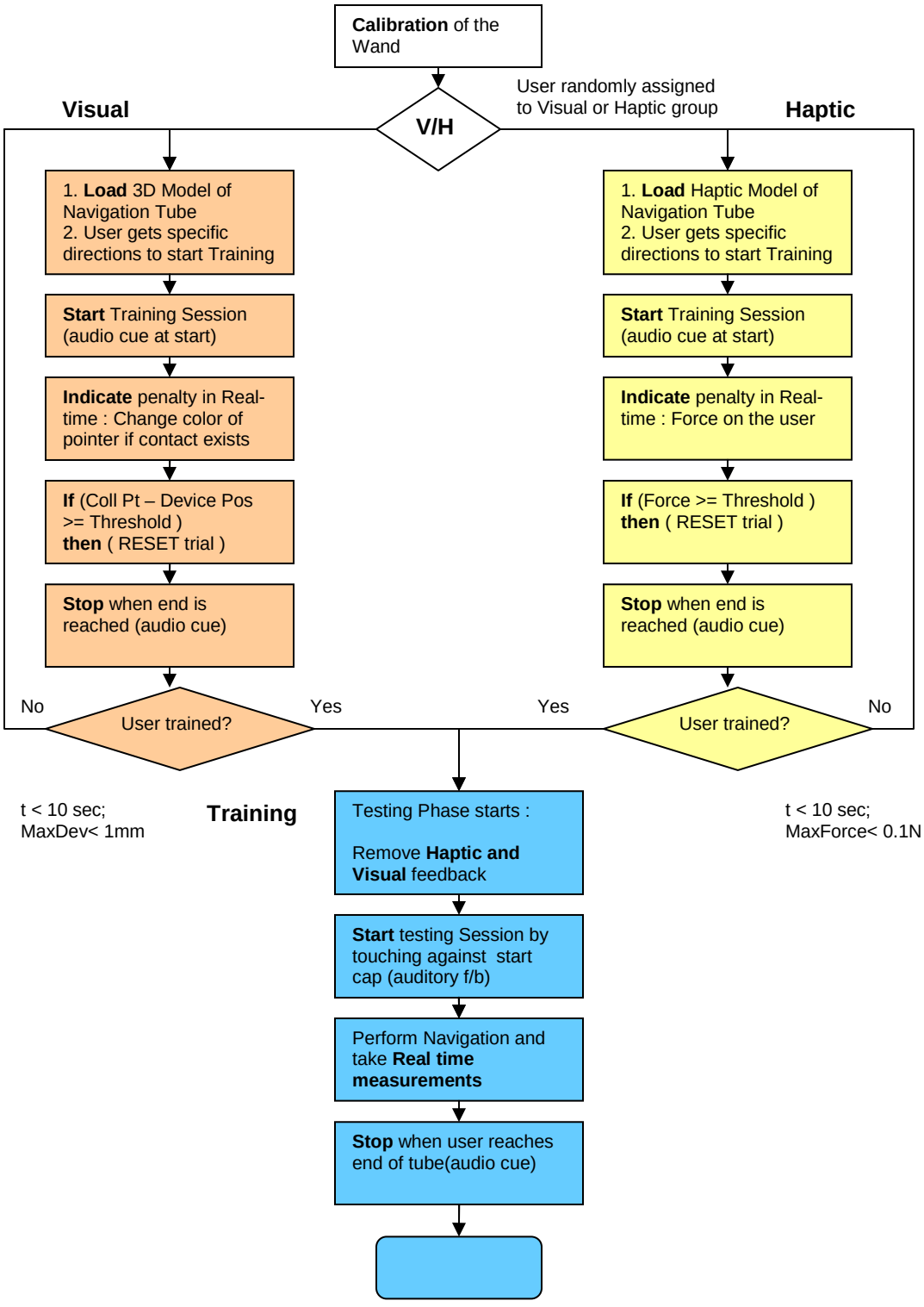

Figure 32: Flow chart for the experiment

Figure 33: Participant training with the visual model.

4.3 CU LaparoWand

4.3.1 Hardware

The central piece of equipment in this experiment is the force-feedback device. Quanser's 5-Degree-of-freedom (DOF) Haptic Wand is used for this purpose. The hardware is modified so that the user interacts with the wand via the handle of a standard laparoscopic tool as shown in Figure 34.

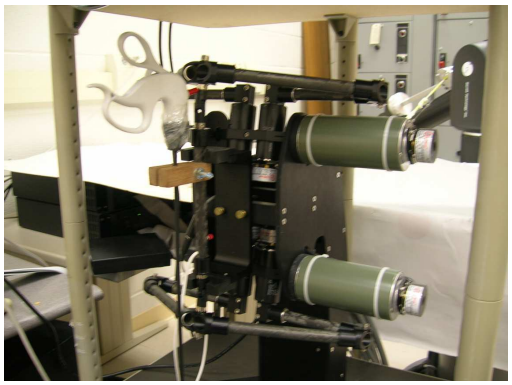

Figure 34: The CU LaparoWand.

The device presents no risk for the user because of its low force output. The maximum force it can exert in one direction is about 2N. It is, therefore, safe for even novices to use this device and presents zero risks or after effects. The Wand is a commercial device designed for this purpose. The control and programming of this device is done using MATLAB (7.1) software with Real Time Workshop (v 6.5). Design of Haptic objects is done with ProSense toolbox (v 2.1) for MATLAB.

4.3.2 Building of Models

In order to carry out this experiment three models are required: the Haptic Training model, the Visual Training Model and the Test Model. This section will present the methods and tools used to build these models. First, the core subsystems common to all models are described. Subsequently, changes to the core model to accommodate for training and testing methods are presented.

The first step was to build a three dimensional tube for navigation. Since the platform used in this system is VRML based, a ".wrl" file was built for the tube. V-Realm Builder (v2.0) is a standalone editor that creates vrml files in VRML2 format. For the purpose of this experiment an extrusion node was selected. The cross section of the extrusion is circular. The spine of the tube specifies the direction of the tube. The spine was designed as shown in Figure 35 using Extrusion Editor. Each line in the spine field specifies a vertex (location in 3D space). Since entry and exit mechanisms are provided for the user, virtual objects are also created for these. For the funnel, an extrusion node is created. The cross section of the funnel is rectangular and it is coded that the width and height of the cross section decrease

linearly from start to end until a fixed area is reached. The start and end caps are created using the Box node as a thin sheet.

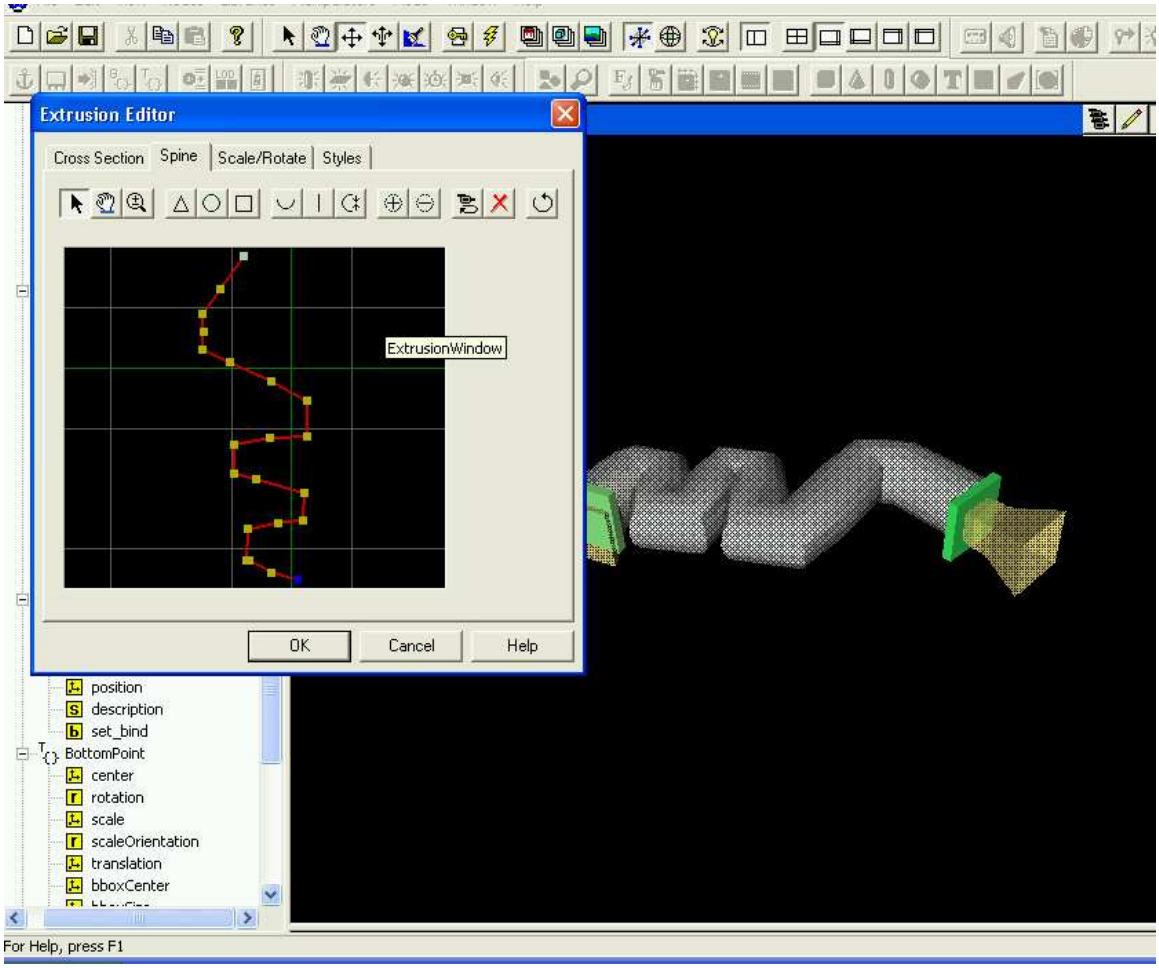

Figure 35: VRML construction of navigation tube.

The next step is the Simulink modeling of the model. To increase realism the two-point haptic rendering method described in Chapter 3 is used for visual and haptic rendering. Two points, equally spaced are used to represent the top and bottom of the wand. Henceforth in this thesis, they will be referred to as the "Top Point" and the "Bottom Point".

The Rendering subsystem visually and haptically renders the model. Input to the subsystem is a "*.wrl" file. A readily available proSense™ block called "HVRWorld" is used in the

Simulink model. Since two points, the Top and Bottom points, are inputs for position, the block is used in dual configuration mode. This block is assigned the pre-built tube model with the entry and exit enhancements. Once this is done, proSense will let the user assign and set haptic parameters. Also placing for the different objects (tube, funnels, caps) are fine tuned in real time. Any node can be made variable by assigning a name to it. Once this is done, input to the node can be given using Simulink's wide assortment of constants, functions and user-defined functions. For haptic parameters, proSense offers mainly four variables: stiffness, damping, Coulomb friction and Coulomb velocity. Each of this parameters can be set as a constant during simulation or be made variable using the above described methods. The outputs to this block are two forces. These forces are summed and fedback to the wand input. (adjustments are made to force values depending on resolution of wand translation).

For the haptic model, the tube is turned invisible by setting tube color to black. Haptic parameters are fine tuned and set to a constant value. No data is collected during haptic training. The stiffness value is set so that though the user feels a constant resistance when in contact with the walls, persistent contact will not cause instability in the device. For the visual model, all haptic parameters are set to zero. Thus, users will not feel any forces as they traverse the tube. However, tube color is set to green. When either the top or bottom point comes in contact with the tube, tube color changes to red. An embedded function is defined for this purpose. This visual cue will let the user stay in the path and therefore, train accordingly. In both cases, the start and end caps are visual, with the colors changing when both points are in contact.

In the test model, the entry and exit mechanisms are left intact. The funnels are both visual and haptic and the start and end caps give visual feedback when contacted with. However, the tube is not present in this model. Thus, when users test, they are "drawing" in free space, attempting to retrace the tube path. Data is recorded in this model. Users are requested for two set of data (two trials) to test for consistency in data pattern.

4.4 Results and Discussion

The test data for each participant is stored on disk. For analysis, this data is run offline with a Simulink model that has haptic parameters assigned. This method gives the ability to make precise force and deviation calculations using proSense™. After this model is run, variables are collected in the workspace and a MATLAB script file is run to calculate effectiveness, time taken to complete the task, top point deviation mean, bottom point deviation mean and the total deviation mean. Each trial has tens of thousands of data points for multiple variables. The goal of this analysis is to compare the three dimensional path of participants and test the hypothesis. The key variables in this analysis are "*time*" (time taken to complete task from start cap to end cap) and "*deviation*". "*Deviation*" is calculated as *Point Position* – *Collision Point* . This value is three dimensional and to get a single value the mean square root of the variables is found; hence, $"deviation" = \sqrt{x^2 + y^2 + z^2}$. The mean top point and bottom point deviations are found and denoted as D_{top} and D_{bottom} .

Below is an example of the list of calculated parameters.

TimeTaken = 3.3487
TopDevMean = 0.0704
BottomDevMean = 0.1134
Total = 18.3844
Effectiveness = 81.6156

Shown in the following figures are graphs pertaining to key variables. Figure 36 contains a sample path traced by the user from start cap to end cap. The two lines indicate the top and bottom point trajectories. Figure 37 indicates the distance deviations of top point in x, y and z axes individually. Figure 38 shows the aforementioned for the bottom points. **Error! Reference source not found.** and **Error! Reference source not found.** show the force deviations for top and bottom points respectively.

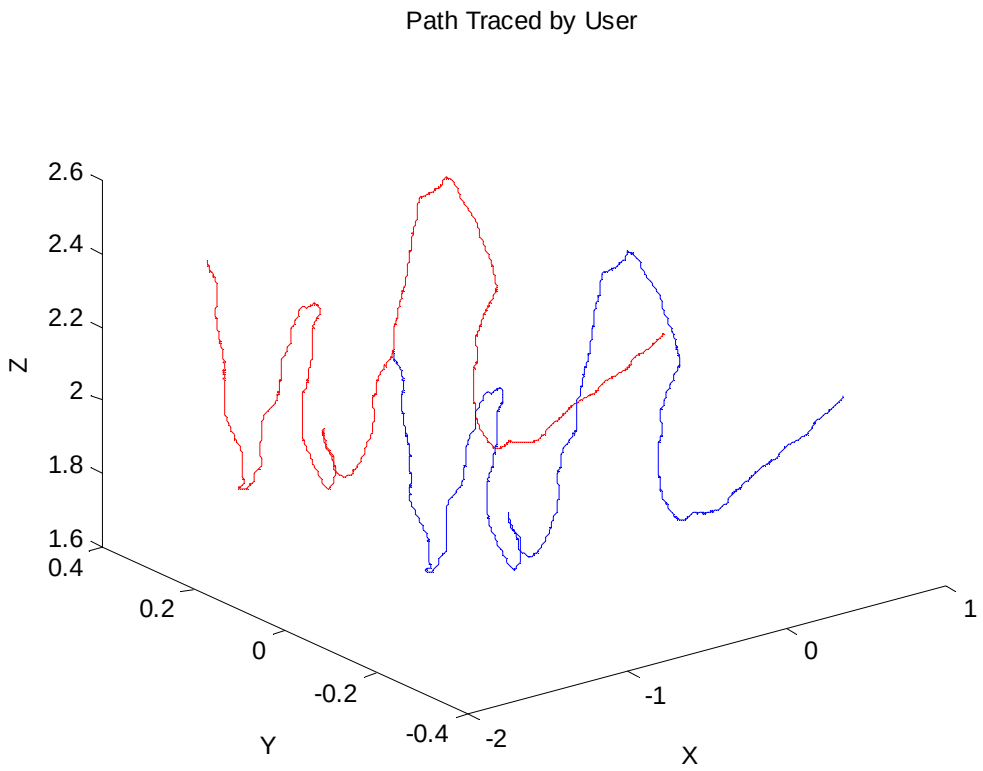

Figure 36: Sample Path traced by a user.

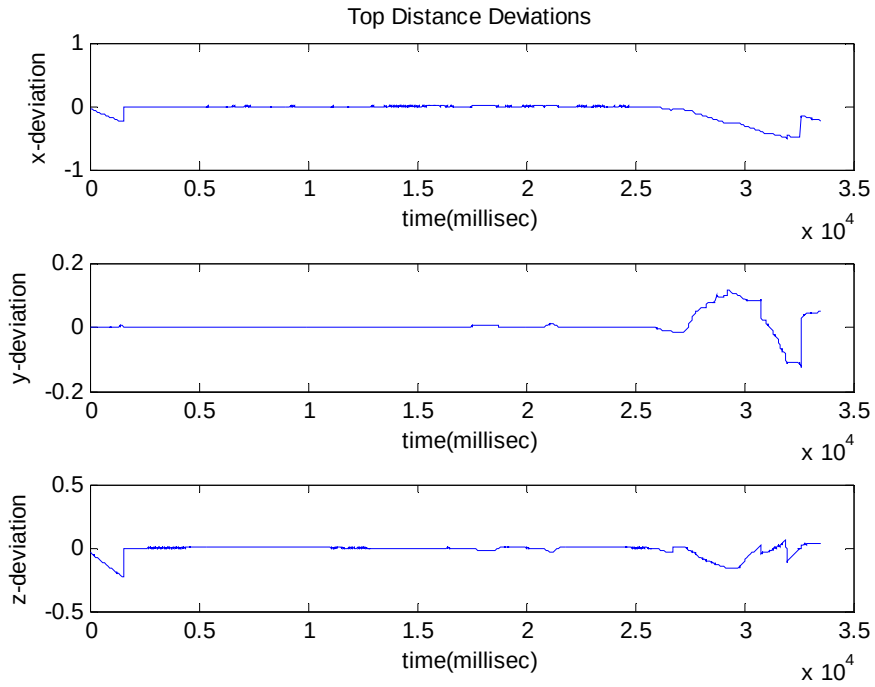

Figure 37: Top point distance deviations.

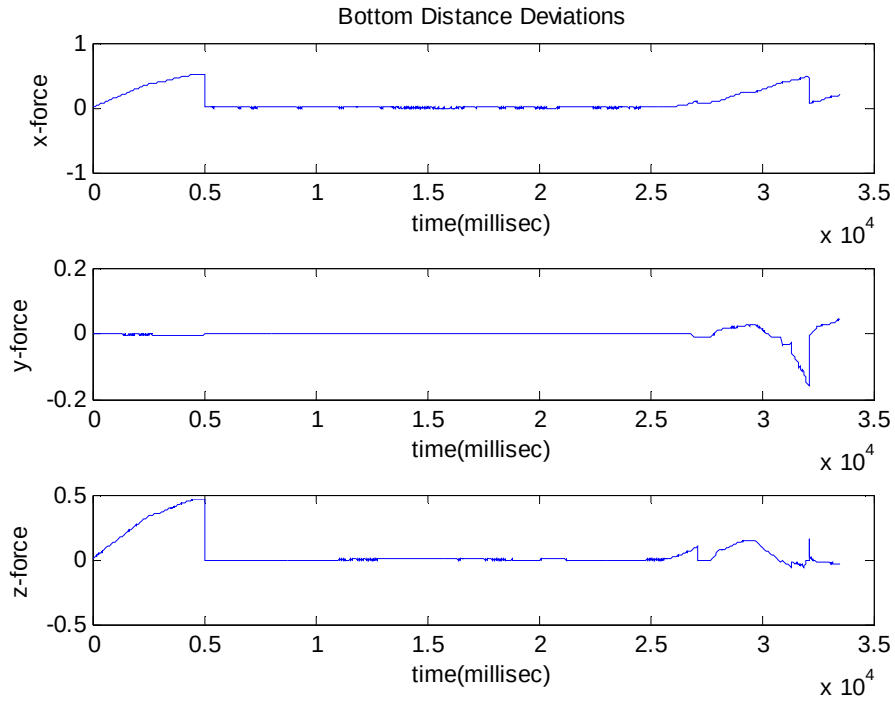

Figure 38: Bottom point distance measurements.

Data from all participants is summarized in Table 4. Data is gathered after offline analysis for all users. The two values calculated are average deviation and time taken for the task. The goal of the experiment is to understand the relationship between training method (visual or haptic) and performance on a kinesthetic navigation task through answering the following three inquisitions:

1. Is the task performance metric (*deviation*) related to the time taken (*time*) to perform the task?
2. Are the means of the output variable "*time*" significantly different for the two training groups (Haptic and Visual) at the p=0.05 level?
3. Are the means of the study variable "*deviation*" significantly different for the two training groups (Haptic and Visual) at the p=0.05 level? If so, which one is better?

The p value in the above questions represents the probability of incorrectly observing a difference as large as was observed from sampling identical populations. The α-value in statistics denoted acceptable level of error; = .05 level was used Due to the low number of data points recorded, the statistical points recorded and lack of other information regarding the distribution of data, the statistical analysis performed does not assume Gaussian distribution of data. Therefore, non-parametric tests are chosen to test the hypotheses.

To address the first question, the non-parametric Spearman Rank Correlation test is used with the following. Spearman correlation test is known to test the strength of the link between two sets of data by using ranks of the two sets of data (as opposed to numerical values). The analysis leads to calculation of Spearman rank correlation coefficient. If the value is 0, then there is no correlation between the data. On the contrary, positive or

negative values tending to 1 or -1, show that the sets of data are positively or negatively correlated

Table 4: Data Collected from 10 participants.

	Trial #	Deviation(mm)	Time (min)
User1	Trial 1	.20935	3.0323
visual	Trial 2	.221266	3.1638
User 2	Trial 1	1.718037	1.8198
visual	Trial 2	2.170987	1.1136
User 3	Trial 1	.892163	4.2739
visual	Trial 2	1.068168	4.6884
User 4	Trial 1	0.102428	3.443
haptic	Trial 2	0.025699	2.8556
User 5	Trial 1	0.052288	3.1354
haptic	Trial 2	0.16839	2.8868
User 6	Trial 1	0.116908	5.6858
haptic	Trial 2	0.046513	7.7259
User7	Trial 1	0.045892	1.784
haptic	Trial 2	1.031205	2.1267
User8	Trial 1	.283343	4.4071
visual	Trial 2	1.652648	4.0031
User9	Trial 1	0.331182	2.2161
visual	Trial 2	.991382	2.2161
User 10	Trial 1	1.07957	3.7418
haptic	Trial 2	0.183844	3.3487

In order to answer the first inquiry, a null hypothesis is set. The alternate hypothesis is also defined. The null hypothesis is defined as correlation constant = 0; that is, there is no rank correlation between the two sets of data (time and deviation are randomly mixed). The alternate hypothesis is that there is a correlation between time and deviation; this means a

correlation constant ≠ 0. The two data groups are tabulated and time and deviation are ranked as shown below.

Table 5: Data analysis for hypothesis 1.

Spearman Correlation Test : Hypothesis 1

Group 1 : Visual Training

Time	*Rank*	Deviation	*Rank*	d	d^2
3.0323	5	0.20935	1	4	16
3.1638	6	0.221266	2	4	16
1.8198	2	1.718037	9	7	49
1.1136	1	2.170987	10	9	81
4.2739	8	0.892163	5	3	9
4.6884	10	1.068168	7	3	9
4.4071	9	0.283343	3	6	36
4.0031	7	1.652648	8	1	1
2.2161	3.5	0.331182	4	0.5	0.25
2.2161	3.5	0.991382	6	2.5	6.25
					223.5

Spearman Correlation Test : Hypothesis 1

Group 2 : Haptic Training

Time	*Rank*	Deviation	*Rank*	d	d^2
3.443	13	0.102428	5	2	4
2.8556	7	0.025699	1	2	4
3.1354	10	0.052288	4	1	1
2.8868	8	0.16839	7	3	9
5.6858	19	0.116908	6	3	9
7.7259	20	0.046513	3	7	49
2.784	2	0.045892	2	0	0
2.1267	4	1.031205	9	8	64
3.7418	14	1.07957	10	2	4
3.3487	12	0.183844	8	2	4
					148

Spearman's rank test utilizes the following calculations. The value of d^2 is calculated and is used in the formula for calculating the correlation rank coefficient

$$r_s = 1 - \{(6 \sum d^2) / (n(n^2 - 1))\} ,$$

where d is the difference between ranks for each observation and n is the number of sets of observations. The values of r_s for the two groups are shown below:

$$Group1(Visual): r_s = -0.35$$
$$Group\,2\,(Haptic): r_s = 0.1030$$

The correlation coefficient can be determined from a table of critical values for the Spearman Rank correlation test. This proves that the rate of performance (time) and the quality of performance (deviation) are not related. Thus, the null hypothesis is accepted at the p=0.05 level. It can be inferred that for five samples, a value of 1.00 is required to accept the alternative hypothesis at the p=0.05 level [38].

The second inquiry deals with the means of the study variable "time" and whether the means of time are significantly different between both groups. Since the non-Gaussian is assumed for the data, a suitable non-parametric test is selected to perform the analysis. The Mann Whitney U-Test is chosen because of its applicability in testing the null hypothesis for two data samples that come from the same population. The null hypothesis for this case is chosen as the mean for both groups is the same ($\mu(visual)=\mu(haptic)$). The alternate hypothesis is the contrary: the means for both groups are not the same ($\mu(visual)\neq\mu(haptic)$). The data is tabulated for calculation of the sum of ranks, R, as shown in Table 6, where the time and deviation have been ranked for the two groups.

To utilize the Mann-Whitney test, the following values are calculated. The formula for calculating the U value is

$$U = n_1 n_2 + [n_1(n_1 + 1)]/2 - \sum R_i$$
$$n_1 = size\ of\ Group\ 1 - Visual$$
$$n_2 = size\ of\ Group\ 2 - Haptic$$
$$R = sum\ of\ ranks$$

The calculated value for U is 54. In order to prove the hypothesis at p=0.05 level, the z value is to be calculated ($z = (U - \mu_u)/\sigma_u$). μ_u is called the mean of the sampling distribution and is calculated by $\mu_u = n_1 n_2 / 2$. In this case, μ_u is 50. σ_u is the standard error for the U-statistic and is calculated by $\sigma_u = \sqrt{[n_1 n_2 (n_1 + n_2 + 1)]/12}$. For this data, σ_u is calculated as 13.23. From the values of U, μ_u and σ_u, z is calculated as -0.3023. The z-statistic table is utilized [39] to find the critical value for α=0.05 as 1.96. This value signifies that if z lies between -1.96 and +1.96, the null hypothesis is not rejected. Since this is true for the experiment data, the null hypothesis is retained and true. The mean values of Group 1 and Group 2 are not significantly different for the variable "time".

The third inquiry sought to understand if the means for the outcome variable deviation is significantly different for both groups. The analysis for this section is very similar to the previous one, the only change being the data in questions is deviation instead of time. Mann-Whitney-U test is performed on the data. The null hypothesis is chosen as the deviation mean for both groups is the same ($\mu(visual) = \mu(haptic)$). The alternate hypothesis is the contrary: the deviation means for both groups are not the same ($\mu(visual) \neq \mu(haptic)$). As shown in Table 6, deviations for each trial are ranked and the sum is calculated for the U value.

Table 6: Data analysis for hypothesis 2

Mann-Whitney U Test : Hypothesis 2

Group 1 : Visual

Time	*Rank*	Deviation	*Rank*
3.0323	9	0.20935	9
3.1638	11	0.221266	10
1.8198	3	1.718037	19
1.1136	1	2.170987	20
4.2739	16	0.892163	13
4.6884	18	1.068168	18
4.4071	17	0.283343	11
4.0031	15	1.652648	17
2.2161	5.5	0.331182	12
2.2161	5.5	0.991382	14

Group 2 : Haptic

Time	*Rank*	Deviation	*Rank*
3.443	7	0.102428	5
2.8556	3	0.025699	1
3.1354	5	0.052288	4
2.8868	4	0.16839	7
5.6858	9	0.116908	6
7.7259	10	0.046513	2
2.784	2	0.045892	3
2.1267	1	1.031205	15
3.7418	8	1.07957	16
3.3487	6	0.183844	8
	Sum = 101		Sum = 143

$$U = n_1 n_2 + [n_1(n_1+1)]/2 - \sum R_i = 12.$$

$$\mu_u = n_1 n_2 / 2 = 50$$

$$\sigma_u = \sqrt{[n_1 n_2 (n_1 + n_2 + 1)]/12} = 13.23.$$

From the above values, $z = (U - \mu_u)/\sigma_u$, is calculated as -2.87. It can be recalled that the critical value for z at the p=0.05 level is 1.96. Since the z value lies beyond the acceptable range, the null hypothesis is false.

Table 7: Data analysis for hypothesis 3

Mann-Whitney U Test : Hypothesis 3

Group 1 :

Time	Deviation	*Rank*	d
3.0323	0.20935	9	4
3.1638	0.221266	10	4
1.8198	1.718037	19	7
1.1136	2.170987	20	9
4.2739	0.892163	13	3
4.6884	1.068168	18	3
4.4071	0.2833425	11	6
4.0031	1.652648	17	1
2.2161	0.331182	12	0.5
2.2161	0.991382	14	2.5

Group 2 :

Time	Deviation	*Rank*	d
3.443	0.102428	5	2
2.8556	0.025699	1	2
3.1354	0.052288	4	1
2.8868	0.16839	7	3
5.6858	0.116908	6	3
7.7259	0.046513	2	7
2.784	0.045892	3	0
2.1267	1.031205	15	8
3.7418	1.07957	16	2
3.3487	0.183844	8	2

Sum = 143

Consequently, this means that one group's mean is significantly different that the other groups. It can also be noted in this case that Group 1-Visual Training mean is significantly higher than Group 2-Haptic Training mean for the deviation variable.

4.5 Conclusions and Future Work

Test data is analyzed for the experiment to yield the following results:

1. There is no correlation between time and deviation of the two types of training. The length of time to perform the kinesthetic navigation task was not related to the training method.

2. There is no significant difference in the means of the time variable between the two groups
3. There is a significant difference in the means of the outcome deviation variable between the two groups.
4. The deviation mean for visual training is significantly higher than the deviation mean for haptic training.

Overall, the initial hypothesis that haptic training is better than visual training for kinesthetic navigation tasks is proved by this experiment.

It is of interest, however, to devise methods to more accurately measure deviation. For example, two paths with completely different wand trajectories could still have the same average deviation. Also, the effect of haptic parameters on haptic feedback and visual feedback mechanisms can also be further probed.

APPENDICES

Appendix A

Building virtual haptic worlds with proSense™ toolbox (v 2.2)

In this appendix, a short tutorial on using proSense toolbox for MATLAB for building haptic virtual worlds is presented. It is assumed that the reader is familiar with the MATLAB/Simulink simulation environment. proSense installs as a toolbox for Simulink and has a library of blocks for haptic world building purposes. Simple 3D object building is shown.

To start the 3D modeling, use a VRML editor to construct a cube. In this work, V-realm Builder (v 2.0) is used. VRML has an option for standard 3D shapes. Select the box or cube option. A 3D box is placed at the origin (0, 0, 0). VRML code structure uses "nodes" for each object in the world. The node has several fields for adjustable properties such as color, translation, rotation etc. These parameters are adjustable. Sizing and scale for the cube can be set as [1, 1, 1]. Since an avatar needs to be designed in the virtual world for the wand, a thin cylinder is chosen for this purpose. Choose the option for cylinder in the VRML editor and adjust the radius to make it thin. This model is saved as a "*.wrl" file. This completes the VRML construction.

For the Simulink model, drag a "HVR World" block onto a new model (*.mdl file). Double click to open the block dialog. Choose the VRML file containing the cube for this block. Once this is done, the HVR block will show adjustable parameters for the model. Selecting the translation field will allow for real-time adjustment of that field. It is good practice to

select translation and scale of a 3D object as variable parameters. For setting the haptic parameters, choose the "Material" node and select the "hapticsEnabled" field. When this is set to "yes", adjustable haptic parameters are enabled. Stiffness, Damping and Coulomb friction are the most commonly used haptic parameters. When the parameters are set to the desired values, this block can be exited. Select translation and rotation fields for the cylinder as adjustable. For variable parameters (haptic parameters can also be made variable) a Constant block can be used in Simulink. This block is adjustable is real time.

The Simulink model needs to contain a block for the haptic interface. Since Quanser Inc.'s, Haptic Wand is used, the corresponding block is included in the model. The input to this block is a 5×1 vector of forces (3×1) and torques (2×1). Since the HVR block is not capable of rendering torques, only a 3×1 force vector is used as input. The output position is fed to the "Device Pos" input of the HVR block. In order to calibrate the haptic device on startup, check the "Calibration of startup" option for the Wand. Figure 20 also shown below illustrates the HVR world selection. Figure 40 gives the detailed Simulink diagram of the model.

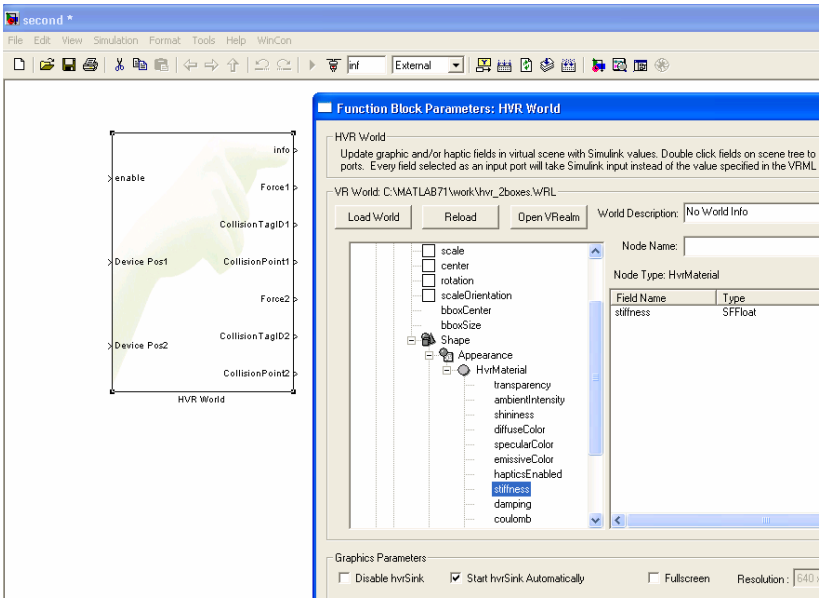

Figure 39: Simulink diagram of HVR World (haptic parameters)

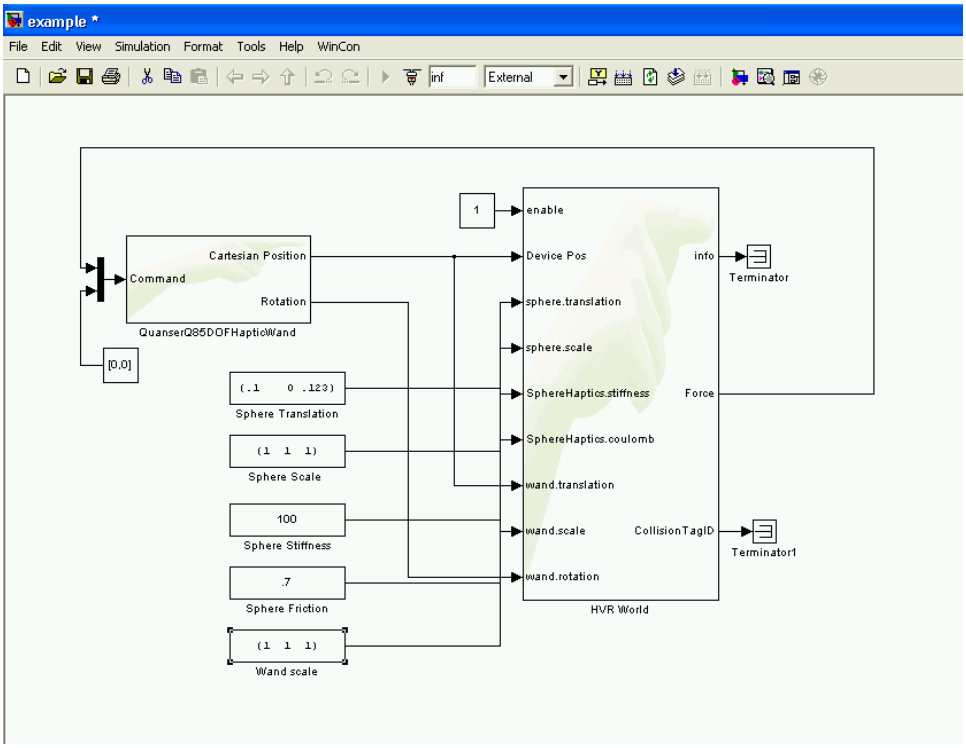

Figure 40: Model building using proSense™.

VRML also has other nodes including nodes for custom building shapes. “Extrusion” and “IndexFace Set” nodes are used for custom 3D shapes

Appendix B

Multi-point haptic rendering model: Code for subsystems

In this appendix, the code and Simulink diagram for the torque model are presented. For detailed presentation of this model, please refer chapter 3. Figure 41 presents the Simulink diagram for a 5 DOF haptic rendering implementation.

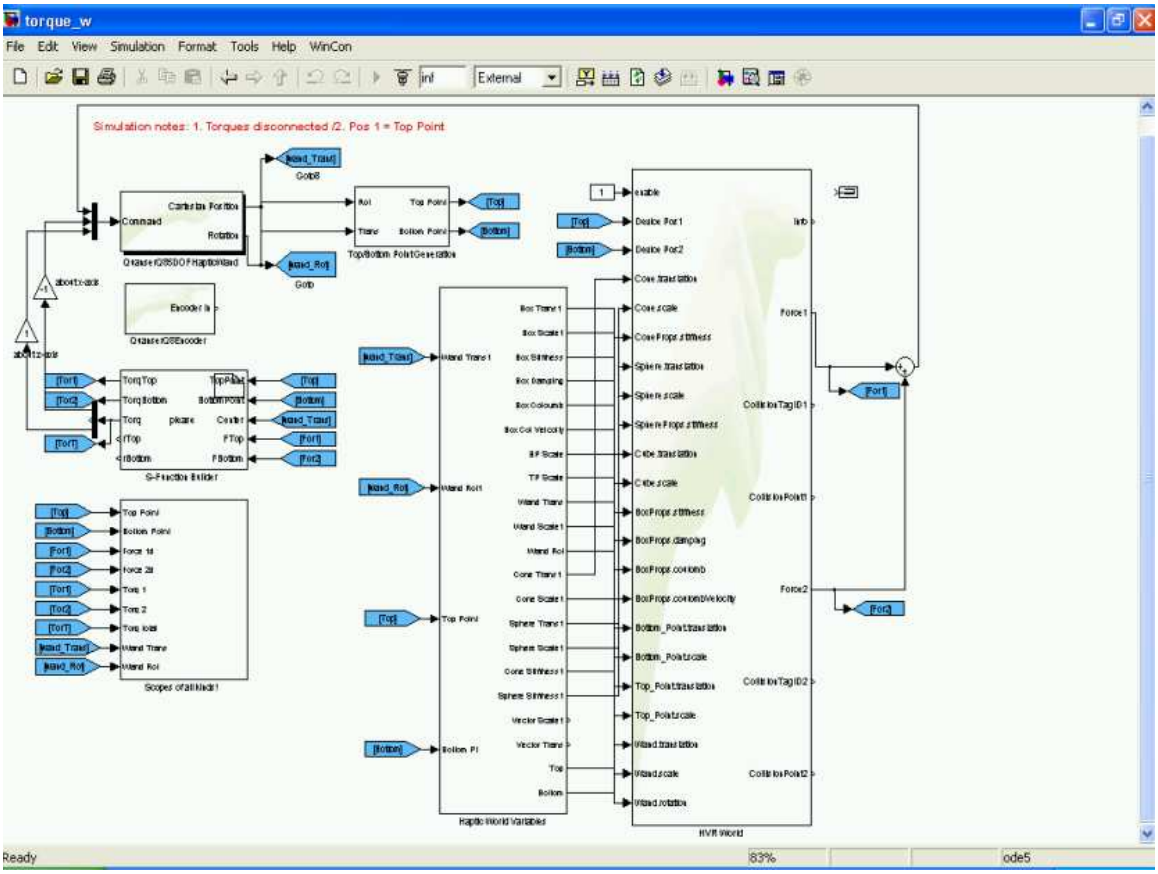

Figure 41: Simulink diagram of torque model.

Code for the Point generation subsystem is shown in

Figure 42. For a detailed derivation of the R matrix, please refer chapter 3.

```
// Initialization

    double kx,ky,kz,theta,x1,x2,x3,R1,R2,R3;
    double r_11,r_12,r_13,r_21,r_22,r_23,r_31,r_32,r_33;

    kx = WandRotVector[0];
    ky = WandRotVector[1];
    kz = WandRotVector[2];
    theta = WandRotVector[3];

    x1 = PointVector[0];
    x2 = PointVector[1];
    x3 = PointVector[2];


//    Expressions for the matrix

    r_11 = (kx*kx)*(1-cos(theta)) + cos(theta);
    r_12 = kx*ky*(1-cos(theta)) - kz*sin(theta);
    r_13 = kx*kz*(1-cos(theta)) + ky*sin(theta);

    r_21 = kx*ky*(1-cos(theta)) + kz*sin(theta);
    r_22 = ky*ky*(1-cos(theta)) + cos(theta);
    r_23 = ky*kz*(1-cos(theta)) - kx*sin(theta);

    r_31 = kx*kz*(1-cos(theta)) - ky*sin(theta);
    r_32 = ky*kz*(1-cos(theta)) + kx*sin(theta);
    r_33 = kz*kz*(1-cos(theta)) + cos(theta);

    //rotation_matrix  =  [ r_11 r_21 r_31;
    //                      r_12 r_22 r_32;
    //                      r_13 r_23 r_33 ];
//
// % Find the final rotation of the point
//      rot = (rotation_matrix) * x ;


    R1 = (r_11*0 + r_12*(.03) + r_13*0) + x1;
    R2 = (r_21*0 + r_22*(.03) + r_23*0) + x2;
    R3 = (r_31*0 + r_32*(.03) + r_33*0) + x3;

     //RotationVector = rotation_matrix*[0 .5 0] + [x1 x2 x3];

//    R = [R1;R2;R3];

 RotationVector[0] = R1;
 RotationVector[1] = R2;
 RotationVector[2] = R3;
```

Figure 42: Code for point generation subsystem

Appendix C

IRB Invitation letter for participation in haptic research experiment

The IRB approval document for the experiment conducted is listed below.

Consent Form for Participation in a Research Study
Clemson University

Testing the effectiveness of Haptic feedback in Kinesthetic Navigation tasks

Description of the research and your participation

You are invited to participate in an exciting research study conducted by Dr.Timothy Burg and Joseph Singapogu. The purpose of this research is to study the effectiveness of haptic feedback in kinesthetic navigation tasks. In other words, given a navigation path in a three dimensional virtual world, our study focuses on he advantages of haptic ("simulated touch") feedback versus traditional visual feedback methods

Your participation will involve using the 5-DOF Haptic Wand in a simulated virtual world. Users will be randomly assigned into two training groups at the outset, the Haptic and Visual groups. After this, depending on the users' group they will be trained accordingly, i.e, users in the Visual group will be trained using visual feedback and users in the haptic group will be trained using haptic feedback.

After this stage, the user will use his/her training to navigate through the 3-D "tube" without any feedback mechanisms. Data is primarily recorded in this stage. Depending on the performance of trained users and the feedback method used, data is analyzed to test the effectiveness of haptic feedback.

The total time for this experiment is estimated to be thirty minutes.

Risks and discomforts

There are no known risks associated with this research.

The Haptic Wand device used in this experiment is capable of producing a maximum force of 2N. This magnitude of force poses minimal if not zero physical hazard.

Potential benefits

There are no known benefits to you that would result from your participation in this research.

This research will help us understand the effectiveness and place of haptic feedback in navigation tasks. Haptic feedback is most widely used in the medical industry and study of these characteristics holds the promise of improving the quality of Minimally Invasive Surgery techniques.

Protection of confidentiality

Personal identification data will NOT be collected in this study. Identification of subjects is done based on a number ID assigned by the test administrator. All data collected for each participant will be marked against that assigned ID number.

We will do everything we can to protect your privacy. Your identity will not be revealed in any publication that might result from this study

In rare cases, a research study will be evaluated by an oversight agency, such as the Clemson University Institutional Review Board or the federal Office for Human Research Protections that would require that we share the information we collect from you. If this happens, the information would only be used to determine if we conducted this study properly and adequately protected your rights as a participant.

Voluntary participation

Your participation in this research study is voluntary. You may choose not to participate and you may withdraw your consent to participate at any time. You will not be penalized in any way should you decide not to participate or to withdraw from this study.

Contact information

If you have any questions or concerns about this study or if any problems arise, please contact Dr.Timothy Burg at Clemson University at (864) 656-1368. If you have any questions or concerns about your rights as a research participant, please contact the Clemson University Institutional Review Board at 864.656.6460.

Consent

I have read this consent form and have been given the opportunity to ask questions. I give my consent to participate in this study.

Participant's signature: Date:

A copy of this consent form should be given t

REFERENCES

[1] "Intuitive Surgical Inc.", 4 Jan 2007, <http://www.intuitivesurgical.com>.

[2] "Novint Technologies", 4 Jan 2007, <http://www.novint.com>.

[3] "Haptics.", *Wikipedia*., 6 Jan 2007, <http://www.wikipedia.com>.

[4] Salisbury. K, Conti. F, Barbagli. F, "Haptic Rendering: Introductory Concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24-32, Mar/Apr, 2004.

[5] Corliss, W.R., Johnson, E.G., "Teleoperator Controls"; *AEC-NASA Technology Survey;* NASA, Washington DC, Ref. NASA SP-5070; 1968.

[6] Mosher, R.S., "Industrial Manipulators"; *Scientific American*; 1964; 211(4); pp. 88-96.

[7] Robert J. Stone, "Haptic Feedback: A Potted History, From Telepresence to Virtual Reality", *Haptic Human-Computer Interaction, First International Workshop*, Glasgow, UK, Proceedings 1-16, August 31 - September 1, 2000.

[8] Burdea, G. C, *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, Inc. 1996

[9] "Proprioception.", *Wikipedia*. 3 Jan2007 <http://en.wikipedia.org/wiki/Proprioception>.

[10] Srinivasan, M.A. and Basdogan, C. "Haptics in virtual environments: Taxonomy, research status, and challenges", *IEEE Computer Graphics,* 21, 4, 1997, pp. 393-404.

[11] Srinivasan, M. A,. "Haptic interfaces: hardware, software, and human performance", *Proceedings of the workshop on human-computer interaction and virtual environments*, Hampton, VA, 1995.

[12] Basdogan. C, "Principles of Haptic Rendering for Virtual Environments", 3 Jan 2007, <http://network.ku.edu.tr/~cdbasdogan/Tutorials/haptic_tutorial.html>

[13] Gottschalk. S, Lin. M, and Manocha. D. "OBB-Tree: A hierarchical structure for rapid interference detection."*In Computer Grajphics (SIGGRAPH '96 Proceedings)*, pp. 171--180, 1996.

[14] Hubbard, P. M., "Collision detection for interactive graphics applications." *IEEE Transactions on Visualization and Computer Graphics*, 1(3):218–230, September 1995.

[15] Baraff. D and Mattikalli. R, "Impending Motion Direction of Contacting Rigid Bodies", Technical Report CMU-RI-TR-93-15, *Robotics Institute, Carnegie Mellon University*, 1993.

[16] Mirtich. B and Canny. J, "Impulse-based Simulation of Rigid Bodies," *Proceedings of 1995 Symposium on Interactive 3D Graphics*, April 1995.

[17] "The Merram-Webster Online Dictionary", 4 Jan 2007 <http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=interface>

[19] Brooks, F. P., Jr. "The computer "scientist" as toolsmith--studies in interactive computer graphics." *Proceedings of IFIP 1977*, 1977. pp. 625-634.

[20] Hayward, V., Astley, O. R., Cruz-Hernandez, M., Grant, D., and Robles-De-La-Torre, G. "Haptic Interfaces and Devices", *Sensor Review*, 24(1):16-29, 2004.

[21] *The Quanser 5 DOF Haptic Wand Reference Manual*, Quanser Inc. Revision 2.

[22] Stocco, L.J.; Salcudean, S.E.; Sassani, F., "Optimal kinematic design of a haptic pen", *IEEE/ASME Transactions on Mechatronics*, , vol.6, no.3 pp.210-220, Sep 2001.

[23] Salcudean, S.E.; Stocco, L., "Isotropy and actuator optimization in haptic interface design" *Robotics and Automation*, 2000. Proceedings. ICRA '00. IEEE International Conference on , vol.1, no.pp.763-769 vol.1, 2000.

[24] *5 DOF Wand Dynamic Equations – Maple Worksheet*; Quanser Inc.,

[25] Craig, J. J. *Introduction to Robotics: Mechanics and Control.* 2nd. Addison-Wesley Longman Publishing Co., Inc. 1989

[26] Spong, M.W and Vidyasagar, M, *Robot Dynamics and Control* New York: Wiley, 1989.

[27] "Axis Angle Representation" *Euclidean Space.* 4 Jan 2007 <http://www.euclideanspace.com/maths/geometry/rotations/conversions/angleToMatrix/index.htm>

[28] Basdogan, C., Ho, C., Srinivasan, M.A., 1997, "A Ray-Based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments" , *Winter Annual Meeting of ASME'97*, DSC-Vol. 61, pp. 77-84, Dallas, TX, Nov. 16-21.

[29] Mullins. J, Mawson. C, Nahavandi. S, "Haptic handwriting aid for training and rehabilitation," *2005 IEEE International Conference on Systems, Man and Cybernetics* , vol.3, no. pp. 2690- 2694 Vol. 3, 10-12 Oct. 2005

[30] Basdogan, C., Ho, C., Srinivasan, M.A., 2001, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration" *IEEE/ASME Transactions on Mechatronics* (special issue on Haptic Displays and Applications), Vol. 6, No.3, pp. 267-285.

[31] Delp S.L., Loan J.P., Basdogan C., Rosen J.M., 1997, "Surgical Simulation: An emerging technology for emergency medical training"; *Presence: Teleoperators and Virtual Environments (Special issue on Virtual Environments in Medicine)*, MIT Press, Vol. 6, No. 4, pp. 147-159.

[32] Basdogan, C., De, S., Kim, J., Muniyandi, M., Srinivasan, M.A., 2004, "Haptics in Minimally Invasive Surgical Simulation and Training", *IEEE Computer Graphics and Applications (special issue)*, Vol. 24, No.2, pp. 56-64.

[33] Marvik, R, Lango, T, Tangen, GA, Andersen, JO, Kaspersen, JH, Ystgaard, B, Sjolie, E, Fougner, R, Fjosne, HE, Nagelhus Hernes, TA, "Laparoscopic navigation pointer

for three-dimensional image-guided surgery" *Surgical Endoscopy* 2004 Aug;18(8):1242-8. Epub 2004 Jun 23.

[34] Feintuch, U., Rand, D., Kizony, R., & Weiss, P.L. "Promoting research and clinical use of haptic feedback in virtual environments." *Proceedings 5th International Conference on Disability, Virtual Reality and Associated Technology,* 2004, Oxford, UK, pp 141-148.

[35] Tendick F, Downes M, Cavusoglu CM, Gantert W, Way LW. "Development of virtual environments for training skills and reducing errors in laparoscopic surgery." *Proceedings of Surgical Assist Systems,* Vol 3262, Bellingham, Wash: SPIE Optical Engineering Press; 1998:36-44.

[36] M. C. Cavusoglu, F. Tendick, M. Cohn, and S. S. Sastry, "A laparoscopic telesurgical workstation", *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 728-739, 1999.

[37] "Immersion Inc.", 4 Jan 2007 <www.immersionmedical.com>.

[38] "Critical values of Spearman's rho (two-tailed)", 4 Jan 2007 <http://www.sussex.ac.uk/Users/grahamh/RM1web/Rhotable.htm>.

[39] "Critical values of the Mann Whitney U Test (two tailed method", 4 Jan 2007 <http://fsweb.berry.edu/academic/education/vbissonnette/tables/mwu.pdf >.