INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

DEVELOPMENT OF A HAPTIC FEEDBACK MODEL FOR COMPUTER SIMULATION OF THE EPIDURAL ANESTHESIA NEEDLE INSERTION PROCEDURE

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the Graduate

School of The Ohio State University

By

Leslie Lynne Hiemenz Holton, M.S.

* * * * *

The Ohio State University
2000

Dissertation Committee:

Professor Alan S. Litsky, M.D., Sc.D., Adviser

Professor Robert H. Small, M.D.

Professor Jan Weisenberger

Approved by

Adviser
Graduate Program in
Biomedical Engineering

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

Computer-based simulations can be a powerful tool for training medical professionals by allowing them to learn and practice skills on realistic virtual patients before working with actual patients. Creating virtual reality medical simulations is challenging due to the complexity and precision of the techniques being simulated. Every aspect of the physical world cannot be simulated exactly, but it is key that the important aspects of the techniques are realistic.

The epidural needle insertion procedure is a common anesthesiology procedure in which the anesthesiologist inserts a needle into the epidural region of the back near the spinal cord in order to inject anesthetic medications. During this procedure, the anesthesiologist relies heavily on perception of force from the needle in order to visualize the tissue layers being penetrated.

The motivation for, and initial development of, a virtual reality simulator for the epidural needle insertion procedure is presented. A user study in which anesthesiologists used the initial simulator and qualitatively scored it's performance is presented. The results of this study indicated that the force-feedback models in the initial simulator did not feel realistic.

The main focus of the research was to create realistic feeling force-feedback models for a virtual-reality computer simulation of the epidural needle insertion

procedure A force modeling methodology has been developed in which biomaterials testing was used to measure load displacement curves as needles were inserted into biological specimens. High resolution MRI scans were performed prior to the biomaterials testing and registered to the needle insertion positions in order to identify each layer the needle punctured.

Force models were developed for each type of tissue punctured during the epidural needle insertion procedure. These models were combined to create a generalized force model for the procedure. Force models were also developed for needle insertion into oranges.

Two virtual patients were developed from MRI data sets of two human subjects, an average female and a lean male. Four virtual oranges were also developed. A haptics-only computer simulator was developed which used a prototype force-feedback device on loan from Immersion Corporation.

The virtual oranges were used in a study to test the realism of force models developed through the force model development methodology. Two user studies were performed in which subjects directly compared needle puncture into physical oranges and virtual oranges in a blind randomized trial. The results of the studies indicate that the subjects could not differentiate between physical and virtual oranges.

A user study was planned to test for construct validity of the epidural simulator. Limitations of the hardware made this study infeasible, but planning the study did provide insight into future development issues.

In Memory of my mother,

Shirley Ann Hiemenz

ACKNOWLEDGMENTS

I would like to thank my husband Andrew Holton, my father Larry Hiemenz, and my sister Christie for their support and encouragement. Without their pushing, prodding, and teasing I would have been hard pressed to finish.

I thank my advisor Dr. Alan Litsky for his help, encouragement, and equipment.

I also wish to thank Dr. Robert H. Small for his advise on all things related to anesthesiology, Don Stredney and Dennis Sessana at the Ohio Supercomputer Center for their assistance and equipment, Dr. Petra Schmalbrock and James Monroe for the MRI imaging in the wee hours of the morning, and Michael Levin at Immersion Corporation for the use of the force-feedback device.

This research was supported in part through a fellowship from the Link Foundation for Simulation and Training.

VITA

July 15, 1968	Born - Portsmouth, VA
1994 - 1996	Graduate Research Assistant, The Ohio State University
1995	M.S. Computer Science, The Ohio State University
1996 - 1997	Research Fellowship, Link Foundation for Simulation and Training
1997 - 1998	Software engineer, Virtual Reality Development MusculoGraphics, Inc., Evanston, Illinois
1999 - Present	Software Development Engineer, Peak Performance Technologies, Inc., Englewood, CO
2000 – 2001	NLM Fellowship in Applied Informatics, University of Colorado, Aurora, CO

PUBLICATIONS

Research Publication

1. "A virtual reality module for intravenous catheter placement." *J.B. Prystowsky, G. Regehr, D.A.Rogers, J.P. Loan, L.L. Hiemenz, K.M. Smith,* American Journal of Surgery, vol. 177, no. 2, 171-5 (1999 Feb).

2. "Development of the Force-feedback Model for an Epidural Needle Insertion Simulator", *L. Hiemenz, D. Stredney, P. Schmalbrock*, In: Westwood JD, et al, eds. Medicine Meets Virtual Reality, IOS Press and Ohmsha, Amsterdam, 272, (1998).

3. "A Physiologically-based Virtual Reality Simulator for Training Residents to do an Epidural Block", *L Hiemenz, J McDonald, D Stredney, D Sessanna, M Fontana,* Proceedings from The Grace Hopper Celebration Of Women Conference, San Jose, California, September 19-21, 69, (1997).

4. "Puncture Mechanics for the Insertion of an Epidural Needle", *L. Hiemenz, A. Litsky, P Schmalbrock*, Proceedings of the 21st Annual American Society of Biomechanics Conference, Clemson, South Carolina, September 24-27, 37, (1997).

5. "A Virtual Simulation Environment for Learning Epidural Anesthesia", *D. Stredney, D. Sessanna, Dr. J.S. McDonald, L. Hiemenz, L. B. Rosenberg*, In: Weghorst et al, eds. MMVR 4: Healthcare in the Information Age, IOS Press, Amsterdam, 164, (1996).

6. "A Physiologically Valid Simulator for Training Residents to Perform an Epidural Block", *L Hiemenz, J McDonald, D Stredney, D Sessanna*, Proceedings of the 15th Southern Biomedical Engineering Conference, Dayton, Ohio, March 29-31, 170, (1996).

FIELDS OF STUDY

Major Field: Biomedical engineering

Other Fields: Computer Science

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF TERMINOLOGY

BMI	Body Mass Index
DOF	Degrees of freedom
Haptics	Relating to the sense of touch
ISL	Interspinous Ligament
JND	Just noticeable difference
LF	Ligamentum Flavum
MRI	Magnetic Resonance Imaging
MTS	Material Testing System
OSC	Ohio Supercomputer Center
PC	Personal Computer
SGI	Silicon Graphics, Inc.

CHAPTER 1

INTRODUCTION

Computer-based simulations can be a powerful tool for training medical professionals by allowing them to learn and practice skills on a realistic virtual patient before working with actual patients. Creating virtual reality medical simulations is challenging due to the complexity and precision of the techniques being simulated. Every aspect of the physical world cannot be simulated exactly, but it is key that the important aspects of the techniques are realistic.

The epidural needle insertion procedure is a common procedure in which an anesthesiologist inserts a needle into the epidural space near the spine. During this procedure the anesthesiologist relies heavily on perception of force from the needle in order to visualize the tissue layers being penetrated. This dissertation describes the development of a virtual reality simulator for the epidural needle insertion procedure. The primary focus of the research is to develop force-feedback models to drive the simulator's force-feedback device that will feel realistic to the user.

Chapter 2 provides an overview of virtual environments and the challenges involved in developing computer based medical training simulators. Then, the epidural procedure and the anatomy of the lumbar spinous region are described. Next, epidural needle insertion training is discussed including current simulators and

the task dependent cues that should be incorporated into a simulation of the technique. A description of the first version of the epidural simulator, developed at the Ohio Supercomputer Center, is then presented. A user survey was performed using this simulator to obtain evaluation of the simulation. The results of the survey indicated that the most unrealistic part of the simulation was the force-feedback. These results helped to define the direction of my research - to find a methodology that can be used to create realistic and accurate force-feedback models for needle insertion.

The focus of the research described in Chapters 3 and 4 was to create realistic feeling force-feedback models for a virtual-reality computer simulation of the epidural needle insertion procedure Chapter 3 begins with a literature review of previous work done in the area of force-feedback devices, haptic rendering methodologies, and needle puncture measurements. Then design specifications are presented for development of the force models. Next, an overview of the force-feedback methodology is presented. Details are then given for materials testing studies that were performed to collect data on needle puncture forces. Chapter 4 presents the force-feedback models developed for each tissue type, the creation of two virtual patients, and the incorporation of the force-feedback models into a simulator for the epidural procedure.

The anatomy of the spinal region is complex, with multiple layers of different tissues. The mechanics of puncturing through these tissues are very complicated. However, what I am developing are models which can run at real-time on a force-feedback device and which will feel realistic to the user. In creating these models

several simplifying assumptions are being made. I am relying on the non-perfect sensory system of the human and the process of distal attribution, defined in section 2.3.1, to "fill in the blanks" of the models. Chapter 5 of this dissertation describes the user testing done to validate the realism of the models created.

An important step in the development of a training simulator is to evaluate the ability of the simulation to model the physical task it is attempting to teach. Chapter 5 first describes user-testing studies performed in order to compare needle insertion into virtual oranges developed with the modeling methodology to real oranges in order to evaluate the realism of the force models. Next, a user study planned to test the epidural models is described. This study did not take place due to difficulties in creating a usable simulator with the given hardware. The problems with the simulator and potential future evaluation studies are presented. Finally, the simulator is evaluated with respect to the design specifications (which are presented in Section 3.3).

CHAPTER 2

SIMULATING THE EPIDURAL NEEDLE INSERTION PROCEDURE

2.1 Introduction

Computer-based simulations can be a powerful tool for training medical professionals by allowing them to learn and practice skills on a realistic virtual patient before working with actual patients. Creating virtual reality medical simulations is challenging due to the complexity and precision of the techniques being simulated. Every aspect of the physical world cannot be simulated exactly, but it is key that the important aspects of the simulated techniques are realistic.

I have worked with the Ohio Supercomputer Center and the Department of Anesthesiology at the OSU Medical Center to develop a prototype computer-based simulation system for use in training anesthesiology residents in the technique of placing a needle for an epidural block. This technique involves inserting an epidural needle into the epidural space that lies posterior to the spine. The anesthesiologist relies on perception of force stimuli at the fingertips to track the location of the tip of the needle. The needle insertion simulator combines real-time graphics, a voice interface, and a custom device [Impulse Engine 1000, Immersion Corporation, San Jose, CA], which provides force-feedback to a Tuohy needle to simulate a realistic epidural experience.

This chapter first provides an overview of virtual environments and the challenges involved in developing computer based medical training simulators. Then, the epidural procedure and the anatomy of the lumbar spinous region are described. Next, epidural needle insertion training is discussed including current simulators and the task dependent cues that should be incorporated into a simulation of the technique. A description of the original epidural simulator is then presented. A user survey was performed using the original simulator to obtain evaluation of the simulation. The results of the survey, presented in Section 2.4.3, indicated that the most unrealistic part of the simulation was the force-feedback. These results helped to define the direction of my research - to develop a methodology that can be used to create realistic and accurate force-feedback models for needle insertion.

2.2 Literature Review

2.2.1 Overview of virtual environments in medicine

Advances in computer interface technology can be incorporated into a realistic interface that allows for natural human-computer interaction. Some of these advanced interface modalities include real-time and static graphical displays, vocal/audio interfaces, olfactory feedback displays, and haptic feedback devices. *Haptic sensations* are "active force, tactile, and motion senses which change as a result of a persons movement through an environment. ... Two classes of haptic feedback devices exist; force-feedback devices using motors or other actuators to stimulate the kinesthetic and proprioceptive senses, and tactile feedback devices using small actuators to stimulate the cutaneous tactile sense." [Hasser, 1996]

Virtual environments are computer simulation systems that make use of multiple interface modalities to simulate a real environment. Virtual reality was described in Rosenberg's thesis, he said:

> The basic premise is that by simulating a physical reality and presenting it to a user through the same sensory channels with which the real physical world is perceived, the resulting *virtual reality* will seem as real to the user as does *the physical reality*. If the sensory information is appropriately represented, accurately mapped to the users sensory channels, and correctly modified as the user interacts with the computer model, the user will be given a sense of *presence* within the virtual world in the same way he or she feels present in the real world. [Rosenberg, 1994]

Virtual environments can be immersive, non-immersive, or augmentative. An immersive environment makes full use of all interface modalities in an attempt to replace the natural environment and make the user feel present within the virtual environment. Non-immersive environments attempt to model only a specific part of the physical world and require the user to suspend disbelief and in order to perceive the environment as realistic. Virtual environments can also be used to augment reality by adding new methods for users to interact with complex data sets.

The first use of virtual environments for training is in flight simulation. The first successful flight simulator was developed by Mr. Edwin A. Link in 1929. [Brown, 1994] Since then, these simulators have incorporated increasingly refined interfaces and have become accepted as an integral part of training pilots, both military and commercial. Recent advances in technology and a paradigm shift in medicine towards minimally invasive surgeries have led to crossover of aircraft simulation technology to medical simulations for education training, for preoperative planning, for surgical assistive tools, and for telemedicine.

Virtual environments work well as simulations for restricted environments, which have both small field of views and field of movements. As an example, minimally invasive surgical techniques such as laproscopic surgery, in which a surgeon operates looking through a laproscope and using tools inserted through small slits in the skin, is a good candidate for virtual simulation due to the limited field of view and limited field of movement. During laproscopic surgery the surgeon does not have his/her hands on the body while looking directly at the work being performed. Instead, the surgeon is operating through hand-eye coordination looking at a monitor in a different view-plane than the surgical site. Because the surgeon cannot manipulate the patient's body directly and cannot directly see the operation as it is being performed, the skills required for this type of surgery are non-intuitive and difficult to learn; therefore, these skills are good candidates for computer simulation training.

Many research groups and companies are developing virtual simulators to train residents on the skills required for minimally invasive surgeries. These simulators will allow residents to practice skills repeatedly before working with live patients, and also allow surgeons to be able to practice new techniques. To be effective the simulator must incorporate not just graphics, but also haptic-feedback so that the surgeons can learn the feel of using laproscopic tools. Descriptions of several of these systems can be found in Westwood, 1998, and Westwood, 1999.

Two companies have developed and are marketing simulators for IV needle insertion. The CathSim Intravenous Training System [HT Medical Systems, Gaithersburg, MD, www.ht.com] is an inexpensive IV needle insertion specific

simulator that runs on a PC. This 1-DOF system simulates the feel of the IV needle insertion, but does not allow the user to position the needle. Musculographics, Inc. [Evanston, IL, www.musculographics.com] markets an IV insertion simulator as part of their LTS Limb Trauma Simulator. This 3-DOF system simulates both the positioning and feel of the IV needle insertion with realistic graphics showing the needle movements, the virtual hand, and the blood flow into the catheter upon venous puncture. Audio feedback (the virtual patient says "Ouch") is provided when the needle is not positioned correctly or goes through the back of the vein.

Virtual environments have also been used to augment reality as a method to help users visualize complex tasks. One example is interactive image-directed neurosurgery, being done at the New York University Medical Center. Using this system the surgeon wears a partially immersive head mounted display that displays MRI or CT data, in effect overlaying the virtual image over the surgical field. As of 1996, 40 epilepsy surgeries had been performed with this device. The device provides for improved anatomical localization of brain structures allowing for smaller openings in the skull and localization of tumors, and should result in a safer procedure for the patient. [Doyle, 1996]

To make a perfectly realistic immersive simulation, the simulator must replicate every sensation felt during a specific task. This is not possible with current technology, and may never be. Instead, the task to be simulated must be carefully analyzed and task dependent cues must be selected which can best represent the task. Simulator developers must rely on the user to suspend disbelief and fill in the blanks between the simulation and reality. Developers must make tradeoffs to

balance speed and cost, to optimize use of available bandwidth across multiple interface modalities, and to maximize the quality of the environment while minimizing development time.

In his article describing the current status of virtual reality in training, Mark Hodges says that the largest technical challenge facing designers is "to make virtual reality a more truly multi-sensory experience. Virtual environments to date have provided increasingly better visual and auditory simulations, but if virtual reality is to reach it's potential for training, future simulators will need to offer users the ability to touch and feel objects." [Hodges, 1998]

At the Ohio Supercomputer Center we have developed a virtual reality epidural needle insertion simulator that focuses mainly on the feel of the needle interacting with the virtual patient's back. Needle insertion is a good candidate for virtual simulation as this task is haptically oriented with only limited reliance on visual cues. The epidural simulator utilizes real-time graphics, a voice interface, and force-feedback applied to a Tuohy needle [Becton Dickinson Company, Franklin Lakes, NJ] to simulate a realistic epidural experience. It is being developed for use as a training device to teach residents to perform the epidural needle insertion procedure. The epidural placement procedure will be briefly described next.

2.2.2 The epidural procedure

Epidural analgesia is one of the most common techniques in use for pain relief in obstetrics. It is also used for surgery and post-operative pain relief for abdominal, groin, thoracic, breast, and lower limb surgeries and for treatment of chronic pain syndromes. A heavy gage (17 or 18) special tipped Tuohy or Crawford

needle is used to place a catheter into the epidural region allowing for either a single bolus injection, bolus injections at regular intervals, or continuous infusion of anesthetics. Properly administered epidural analgesia is the most effective form of pain relief during labor and vaginal delivery, even though 15-25% of patients have a partial block rather than total pain relief during the entire labor. The success rate of epidural anesthesia is directly dependent on the experience of the administrator: for first year residents the failure rate is 12%, for second year residents 4%, for new attendings with 2-5 years experience 2.5%, and for senior attending staff 1%. [Bonica, 1995]

Figure 2.1 shows a line drawing of the vertebral column with key features pointed out. Figure 2.2 is a sagittal slice taken from the visible male data set showing the spinal column. Figure 2.3 shows a "pseudo-coronal" view following the spinal column.

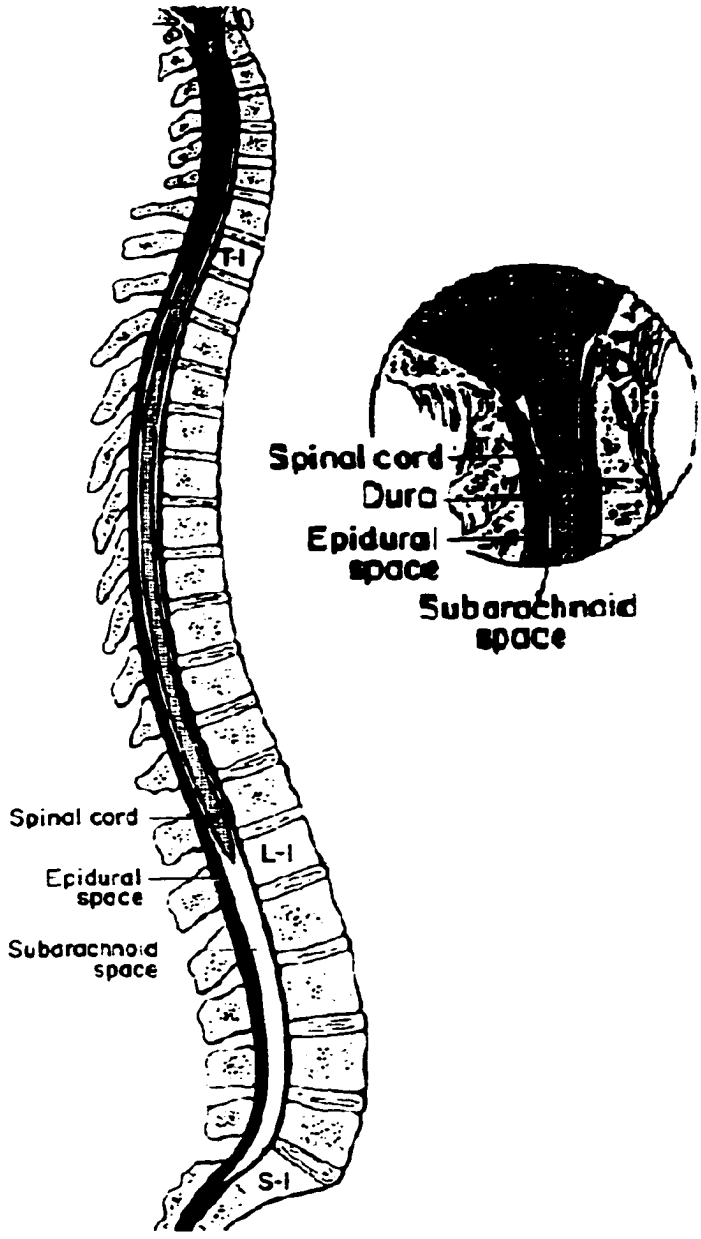

Figure 2.1: Sagittal view of the vertebral column with a close-up of key features around the epidural space. The lumbar vertebrae run from L1 down to L5 (above S1). [Bonica, 1995]

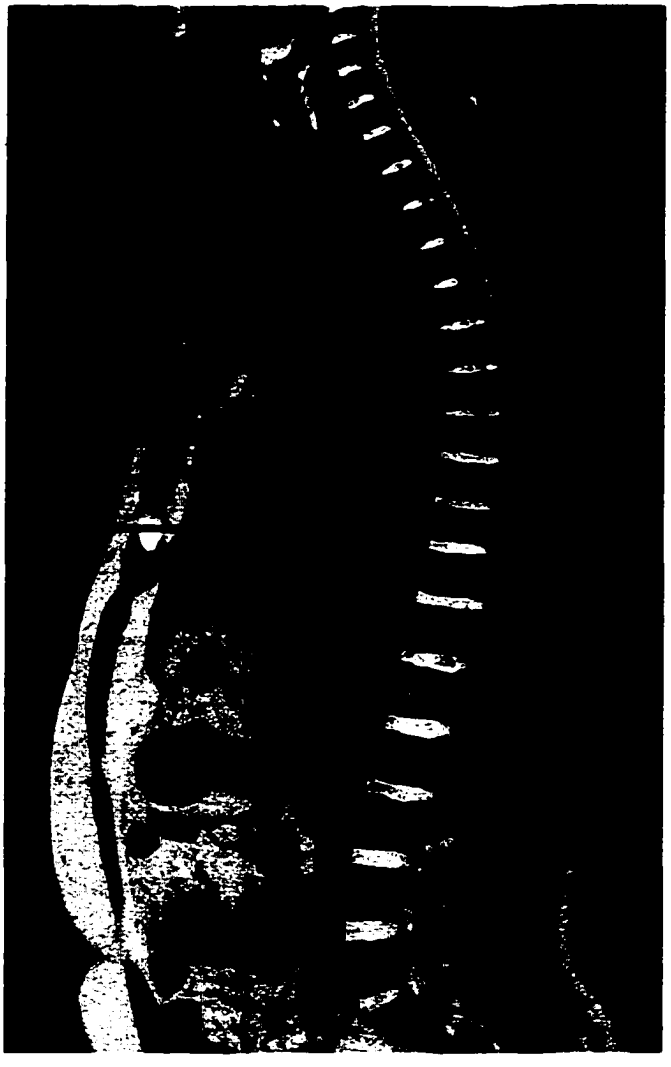

Figure 2.2: Sagittal view of the vertebral column taken from the visible male dataset. [Obtained and used with permission from Vic Spitzer, Center for Human Simulation, University of Colorado, Aurora, CO]

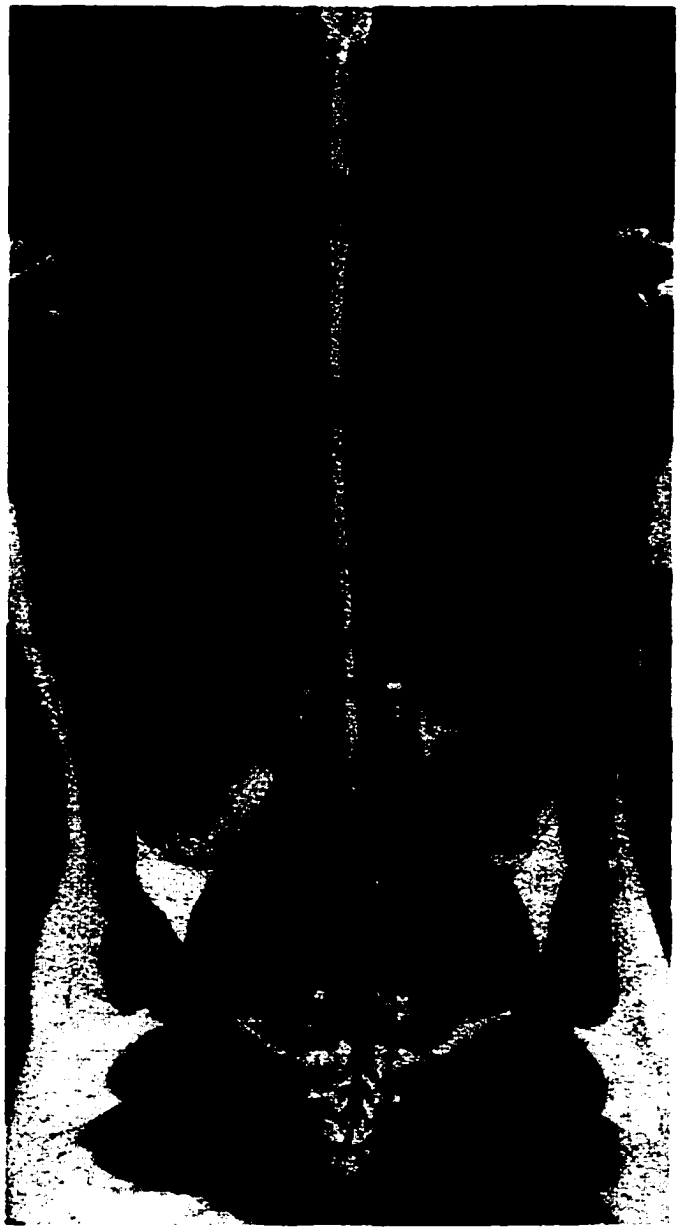

Figure 2.3: "Pseudo-coronal" view of the vertebral column taken from the visible male dataset. This view was made by defining a path on the sagittal view going through the center of the spinal column. Each pixel was then taken from a front coronal view marked by this path. The resulting image is a projection of the spine onto a coronal plane. [Obtained and used with permission from Vic Spitzer, Center for Human Simulation, University of Colorado, Aurora, CO]

The spinal column, illustrated in Figures 2.1, 2.2, and 2.3, normally consists of 30 vertebrae categorized as the cervical (8), thoracic (12), lumbar (5), and sacral (5) regions. The five lumbar spinous processes are thick, broad, 3-4 cm long, and are generally horizontal. This allows for epidural needle placement along the midsaggittal plane with a slight cephalad angulation along the vertical axis of the spine. Between adjacent spinous processes are the intraspinous spaces, and between the two pair of laminae of two adjacent vertebrae are interlaminar spaces. An epidural needle can be inserted into any of these spaces to gain access to the epidural space. These spaces can be widened by having the patient slump, flexing their spine, which makes insertion of the needle easier. [Bonica, 1995]

Three individual membranes known as the spinal meninges surround the spinal cord; the dura mater is the outermost of the meninges. The epidural space surrounds the dura and contains fat, alveolar tissue, nerve roots, and an extensive network of arteries and veins. The epidural space is widest (5-6 mm wide) adjacent to the second lumbar vertebrae (L2). In a study done on 163 obstetric patients, Mieklejohn found that the average distance from the skin to the lumbar epidural space was 4.78 cm. [Mieklejohn, 1987]

The epidural space is bounded anteriorly by the posterior longitudinal ligament, laterally by the intervertebral foramina, and posteriorly by the ligamentum flavum. [Christopherson, 1994] A diagram showing the epidural space and other key features of the vertebral canal is shown in Figure 2.4.

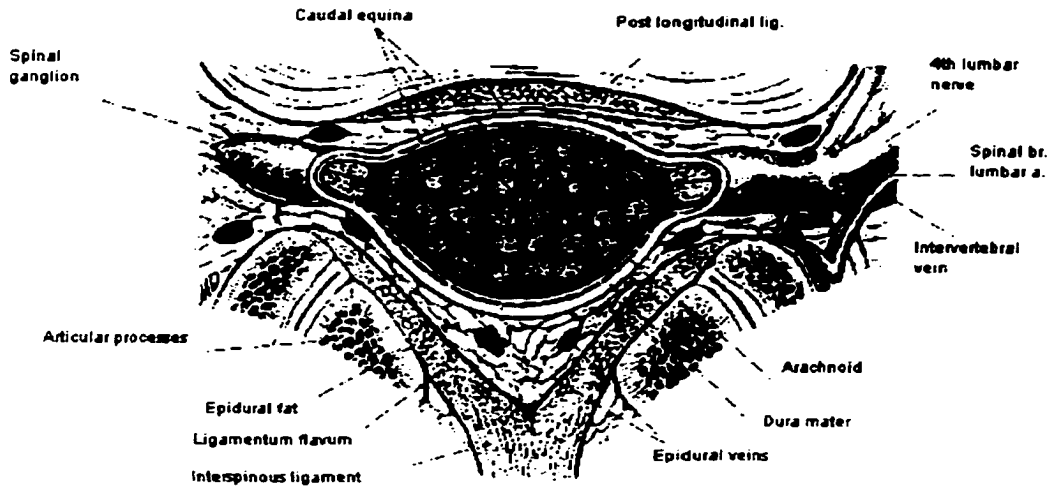

Figure 2.4: Diagram showing the epidural space and other key features in the vertebral canal. [Bonica, 1995]

The intraspinous ligaments fill the space between the ligamentum flavum and the supraspinous ligament in the lumbar region. The supraspinous ligament is a strong fibrous cord that connects the tips of the spinous processes of all the vertebrae from the lower portion of the spine to the tip of the seventh cervical vertebrae. The ligamentum flavum is a tough ligament that posteriorly bounds the epidural space at the midline. This ligament is 3-5mm thick and 12-22mm wide at the L2-3 level. [Zarzur, 1984]

Figure 2.5 is an illustration of the nerve supply to the lumbar vertebrae.

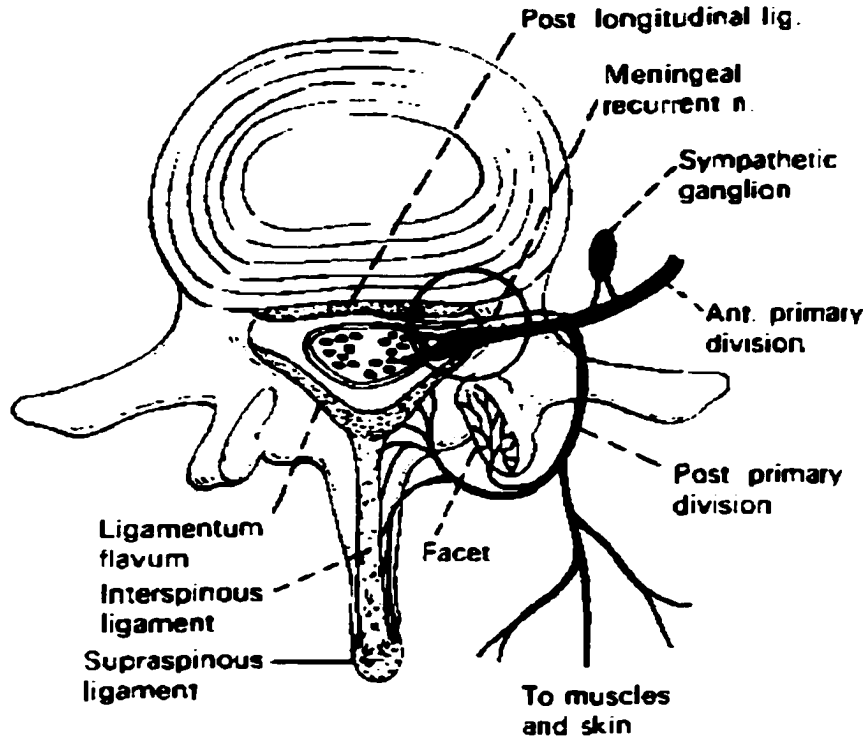

Figure 2.5: The nerve supply to the lumbar vertebrae. Note the posterior primary division that is located in the area the epidural needle is inserted. [Illustration from Bonica, 1995]

Figure 2.6 illustrates the blood supply to the lumbar vertebrae. Each lumbar vertebra is supplied by a pair of segmental lumbar arteries as shown in the top picture. The bottom picture illustrates the venous drainage of the lumbar vertebrae.

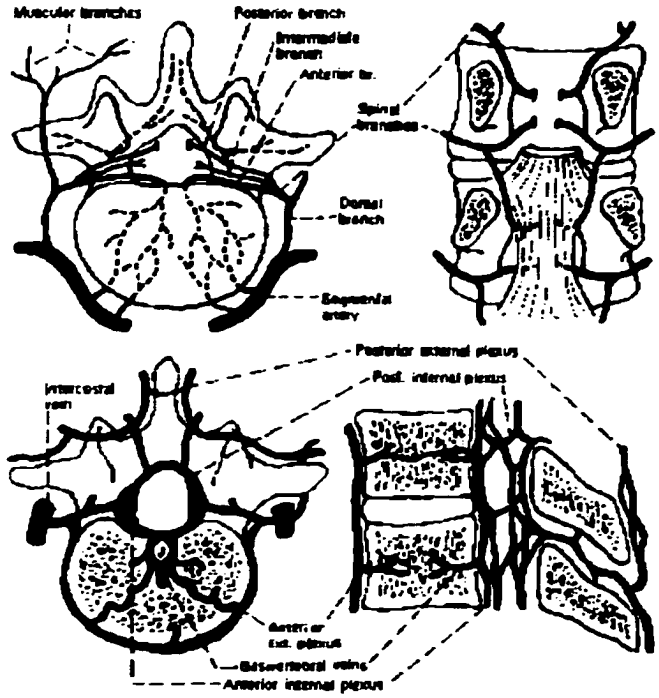

Figure 2.6: The blood supply of the lumbar vertebrae. Arterial supply is shown in the top picture. Venous drainage is shown in the bottom picture [illustration from Bonica, 1995].

A negative pressure is registered when a needle is inserted into the epidural space. Tenting of the dura mater by the point of the needle likely causes this negative pressure. [Bonica, 1995] The loss-of-resistance technique can be used to identify when the epidural space is reached. This technique is based on the difference in the tissue characteristics between the ligamentum flavum and the epidural space. Using this technique, a saline or air filled plunger is attached to the needle during insertion. The anesthesiologist keeps a constant pressure on the plunger, and when the epidural space is reached the saline can be injected. The

ligamentum flavum is a tough connective tissue and does not allow the injection of saline or air. The epidural space is filled with fat that is loosely connected and allows for easy injection.

The first step of the epidural procedure is preparation that includes palpating the back to identify the location for needle insertion, sterilizing the patient's back, applying a sterile drape, and preparing the tray of needles. Next a local anesthetic is administered using a small gage needle. The epidural needle is held using a precision grip by one hand, and the remaining hand is used to brace against the patient's back to control fine forward movement of the needle and isolate the needle's movement against patient movement.

The needle is inserted to the epidural space using either the midline or paramedian technique described below. Once the epidural space is reached, a catheter is inserted through the needle until it is 3-5 cm into the epidural space. Next, the needle is withdrawn over the catheter, leaving the catheter in place, and a test dose is administered to verify that the catheter has not been placed intrathecially or intravascularly. Finally the catheter is draped over the patient's back, and taped down to allow for administration of anesthesia over time.

According to Gaiser, the type of local anesthetic used in the epidural procedure is determined by the duration required, and the required intensity of motor blockade. These anesthetic agents can include (in order of duration) Lidocaine, Chloroprocaine, Mepivacaine, Bupivacaine, Ropivicaine, and Etidocaine. Epinephrine can be added to prolong the effect and reduce the peak blood concentration of the local anesthetic agent. [Gaiser, 1992]

Figure 2.7 illustrates the spread of the local anesthetic agent following epidural injection. The anesthetic spreads posteriorly to the region of the nerve root sleeve and then spreads to the CSF and spinal cord along with the anterior epidural space.

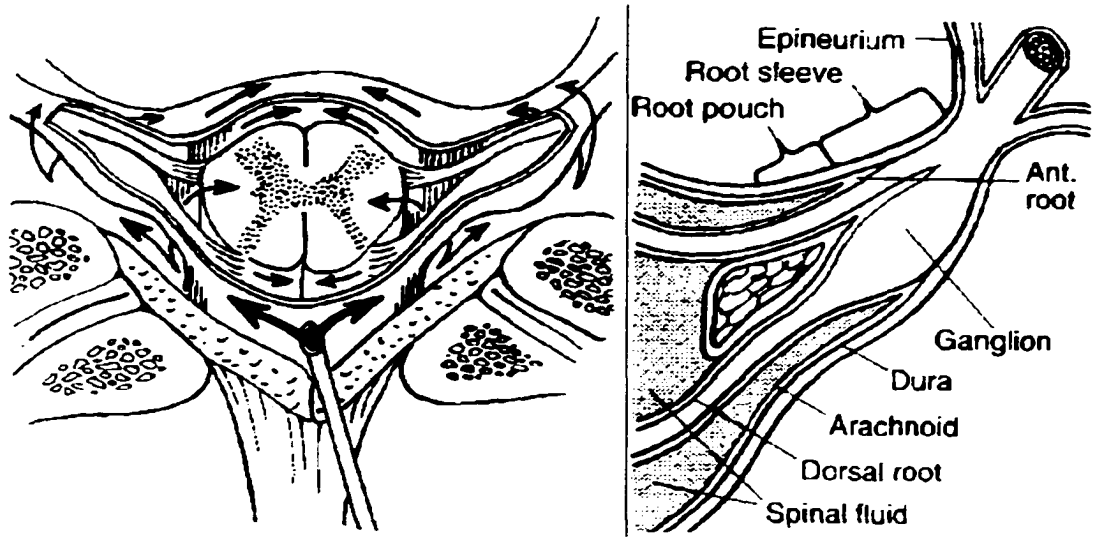

Figure 2.7: The spread of the local anesthetic following injection. The anesthetic spreads posteriorly to the region of the nerve root sleeve and then spreads to the CSF and spinal cord along with the anterior epidural space. [Bonica, 1995]

The midline approach is a commonly used approach that involves penetration of the skin, the supraspinous ligament, the interspinous ligament, and the ligamentum flavum. In this approach the needle is introduced in the exact midsagittal plane. The epidural needle is inserted into the back until the ligamentum flavum is reached as indicated by a strong resistance to needle advancement. The stylet is then removed, and a saline-filled (or air-filled) syringe is attached to the

needle. The needle is carefully advanced, while pressing on the back of the syringe until the ligamentum flavum is pierced and the epidural space is reached, indicated by a loss-of-resistance on the plunger. The entire contents of the plunger are administered in order to widen the epidural space, the syringe is removed, and the catheter is inserted through the needle. This approach and the anatomy of the spinal region are depicted in Figure 2.8.

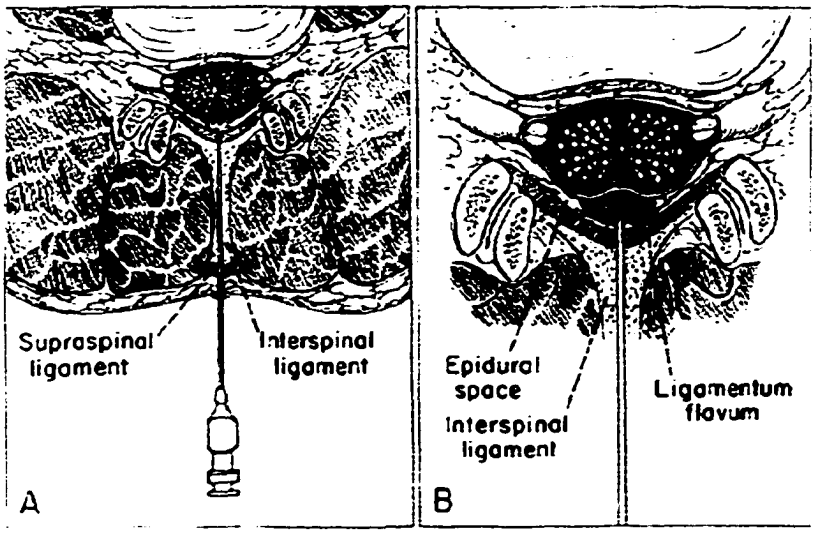

Figure 2.8: Cross sectional anatomy of the lumbar region depicting technique of the epidural puncture using the midline approach. Panel A illustrates needle insertion to the ligamentum flavum. Panel B illustrates the needle correctly positioned in the epidural space. [Bonica, 1995]

There are several problems associated with the midline technique. One is that the needle may deviate from its trajectory while it is puncturing the supraspinous or interspinous ligaments. This may cause the point of the needle to deviate from

midline and the epidural space may not be reached. Another difficulty with the midline approach is that puncture of these ligaments may lead to post-procedure backache due to trauma to nerve receptors. Incorrect catheter placement may occur because the needle placement requires the catheter to make a right angle upon leaving the needle bevel in order to go in the correct cephalid direction. [Bonica, 1995]

In the paramedian technique, needle placement is just lateral to the spinous process through the paraspinal muscles. The needle is introduced at a paramedian angle (15° with the midsagittal plane and 135° cephalid to the long axis of the spinal column) so that upon entering the epidural space, the bevel of the needle points cephalid to assist advancement of the catheter. The epidural needle is inserted until it contacts the surface of the lamina. It is then slowly advanced, "walked down" the laminar surface, until it engages and then transverses the ligamentum flavum. At this time a saline filled plunger is attached to the needle and compressed. The needle is advanced very slowly using the loss-of-resistance technique to identify the epidural space. [Bonica, 1995]

Figure 2.9 illustrates both the paramedian and median approaches to inserting the epidural needle.

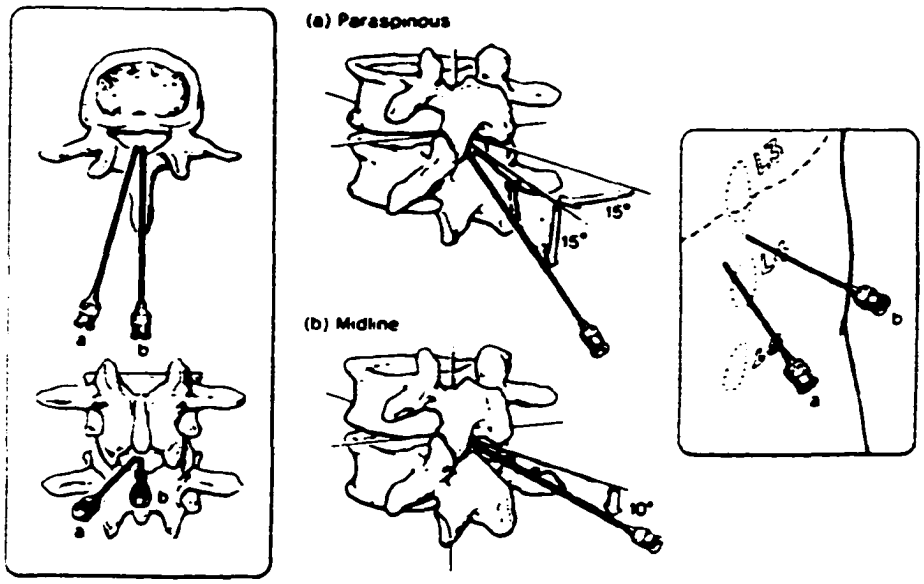

Figure 2.9: Diagram illustrating the paramedian and midline approaches to the epidural technique. A) is paramedian, B) is midline. [Gaiser, 1992]

There are many potential complications related to needle placement that can occur with epidurals. Some common complications are pain on insertion, inability to locate the epidural space, and parethesia due to nerve root stimulation.

Accidental dural puncture is a complication that can occur if the needle goes through the epidural space, through the dura, and into the spinal canal. Dural puncture can be recognized by the presence of cerebral spinal fluid (CSF) in either the needle or the catheter. Because of the large size of the epidural needle, dural puncture may result in a severe headache. Four courses of action are possible on accidental dural puncture: 1) a subarachnoid catheter can be placed to provide continuous spinal anesthesia, 2) a single dose of local anesthetic can be

administered to produce spinal anesthesia, 3) the needle can be removed an the epidural procedure can be attempted at another location, or 4) the needle can be withdrawn to the epidural space and an epidural catheter can be placed at the same sight. [Gaiser, 1992]

A test dose containing epinephrine and the local anesthetic agent can be used to recognize if the catheter has been placed intravascularly or intrathecally. If the catheter lies in the epidural space there will be no anesthetic reaction and no change in blood pressure or heart rate with the injection of the test dose. Intrathecal placement can be recognized by onset of anesthesia with the test dose. Intravascular placement can be recognized by a change in the patient's vital signs. [Gaiser, 1992]

The spinal anesthesia procedure is similar to the epidural procedure in terms of needle position and approach. In spinal anesthesia a smaller gauge needle is normally used, and the needle is inserted through the dura. Once the needle is positioned, the stylet is removed and CSF flows freely from the needle. Benefits of spinal anesthesia include speed in onset and the definite endpoint provided by the flow of CSF. The absence of dural puncture is a major advantage of epidural anesthesia over spinal anesthesia. [Gaiser, 1992]

2.2.3 Teaching epidural needle insertion

Learning to perform an epidural first involves building an internal representation of the three-dimensional configuration of structures that make up the regional anatomy. This construction takes place through reading texts, studying the anatomy, looking at physical models of the anatomy, and talking with

anesthesiologists and other residents. This mental model of the task will be used for spatial processing for the task, including orienting with the physical world, relating self-position to the physical model, reasoning relative positions of landmarks and movement trajectories for spatial navigation, and integrating movements. [Stredney, 1996]

Teaching epidural needle insertion procedure typically follows an observation then perform paradigm. After the resident has reviewed the procedure in depth, he/she will watch an attending anesthesiologist perform an epidural block. As the attending hits each perceivable layer of tissue he/she will have the resident push the needle in order to refine the resident's mental model of the haptic sensation. Next, the resident will perform a series of epidurals under the watchful eyes of the attending. Once the instructor feels that the resident has mastered the procedure, the resident will be allowed to perform epidurals autonomously. With each epidural procedure performed, the resident updates his/her mental model to integrate patient variance and the unique aspects of each procedure.

It is of interest to note that during the epidurals that the attending oversees there tends to be little vocal instruction during the procedure itself. The attending remains hushed and unobtrusive unless the resident needs help, at which time the attending may quietly take over the procedure or instruct the resident in a soft voice. This is mainly out of concern for the patient, who must feel that he/she is in safe hands.

2.2.4 Simulators for training epidural needle placement

A simulation of the patient can be used to train the resident how to insert an epidural needle before practicing the procedure on an actual patient. Adding this step to the training regimen offers several advantages. One advantage is that the instructor can speak in a normal tone to the resident and give detailed instructions during the procedure, as the simulated patient will not care. The simulated procedure can also be stopped or reversed at any time for detailed instruction. Another benefit of simulator learning is that it can be done at any time. While training on real patients the resident may have to wait days, or even weeks, between patients. This allows time for the resident to forget small details of the procedure. With the simulator the resident can do as many trials as he/she wishes on several different patients in a row. The simulator can also be used to brush up on skills, or to do a trial run, before doing the procedure on the real patient.

Another benefit of the simulator over learning on real patients is that with the simulator the resident can make mistakes without harm to a patient. For example, the resident could push the needle through the dura intentionally in order gain an understanding of the force required to push a needle through this layer. This type of learning through "intentional mistakes", or learning through trial and error, is not optimal on a real patient. The fear of making mistakes can cause the resident to lack confidence. However, due to this fear the resident may pay greater attention to details of the procedure on an actual patient than on a simulated patient.

Mannequin models have been developed for simulation of the epidural needle placement technique that are very realistic for the "ideal" patient (e.g.

Armstrong Medical Industries, Inc., Spinal Injection Simulator). The user can insert the needle at many locations and with any orientation. Materials can be used to build the mannequin that can closely approximate the tissue, giving very natural force-feedback. The models can often be taken apart in order to identify the anatomy and to help create a mental model of the anatomy of the back. However, there are limitations to this type of model. One is repeatability. Because inserting the needle requires actually making a hole in the mannequin, there is a limit to the number of times a particular mannequin can be used. Another limitation is that the mannequin model can only represent a single patient type; thus, in order to teach patient variance, many mannequins must be used.

A computer-based simulator has advantages over a mannequin model in that it can be programmed to show patient variance. With a simple change in the data set, the computer-based simulator can represent many different virtual patients including young children, obese patients, and elderly patients with calcification. The simulator can also be made to exaggerate certain aspects to aid in initial learning. Since the needle is being inserted into a virtual patient, a simulator also offers repeatability. Residents can perform exactly the same procedure as many times as they would like, or try different methods for doing an epidural on the same virtual patient for comparison between methods. Virtual patients with abnormal anatomy can also be programmed, to allow residents to fine-tune their skills.

As an additional benefit to the computer-based simulator, graphical extensions can be developed which could be used to aid in resident learning. These extensions could include graphics to identify the needle's location in the virtual

patient, to teach the anatomy, or to illustrate the ideal needle trajectory. The computer simulator could also be set up to collect statistics such as the time required for needle insertion, the location of final needle position, or the number of attempts required to identify the epidural space. These statistics could be used in order to track resident's learning progress.

While there are many advantages to a computer-based simulator, there are also drawbacks. Because virtual interfaces are leading edge technology, these simulators are still in the early stages of development. Development takes time, and money, which will likely make these specialized simulators expensive when they are ready for market. In addition, simulators must be shown to be effective in training residents in order for medical schools and hospitals to be willing to add simulator training to their curriculum.

Computer-based simulators are also limited by technology; for example, force-feedback technology. A force display will not present a valid illusion of feel if it is inherently unstable due to bad system dynamics related to the sampling period. Noise related to the display devices can cause the device to shake, or jitter, which can destroy the illusion of feel. Contact stability, when a virtual wall or hard surface is touched, is a difficult mechanical engineering problem to solve, and one which has not yet been successfully solved in inexpensive devices. Force-feedback devices can be made stable through increasing damping (which makes the user feel resistance and sluggishness even in free space), or by reducing stiffness (which makes it difficult to model hard surfaces). This problem is complicated by the fact

that the human arm is a non-linear system whose control is not fully understood. [Summarized from Minsky, 1990]

Another drawback of virtual environment training is that for residents training on these environments there is no ability to discriminate residual effects, caused by limitations of the simulation, from what actually occurs in reality. Thus, upon completion of training and beginning doing tasks in the real environment the resident will be unable to tell if discrepancies between the environments are due to limitations of the simulated environment or to problems with the task they are performing. Simulator designers must be careful to minimize the artifacts introduced from the virtual environment.

Even under the assumption that we have the ability to create perfect simulations that exactly duplicate reality, there are drawbacks to virtual environments. One is that a reliance on simulation training may decrease the time spent in learning with an instructor, with classmates, and in hands-on training. These types of learning have benefits that may not be available in virtual environments; instructors can share real-world experiences, exchanging information with peers is invaluable, and hands-on real world experience is the gold standard. Reliance on virtual environments for surgical preplanning could reduce the amount of discussion and coordination between staff physicians and the residents being trained. Virtual simulations can be useful tools in medical training; however, users of virtual environments must be careful to use them as aides, and not as replacements for real life.

2.3 Human factors analysis

A human factors methodology was used in the design of the epidural needle placement simulator. *Human factors*, as defined by Nemire, includes: "the application of knowledge about human physical, perceptual, and cognitive capabilities to the design of usable, "user-friendly," tools and systems that accommodate the capabilities and limits of the human user" [Nemire, 1994].

The procedure to be simulated must be carefully analyzed and task dependent cues must be selected which can best represent the procedure. Simulator designers must rely on the user to suspend disbelief and fill in the blanks between the simulation and reality. This suspension of disbelief is defined as *distal attribution*, "the innate human tendency to infer the mental models of our surroundings, which we construct from abstract perceptual cues, are direct reflections of the physical world rather than the stylized and simplified abstractions of reality which they truly are." [Rosenberg, 1994]

In analyzing the epidural procedure there are several cues that should be available in order to make the simulation feel realistic. Since this procedure is mainly haptically oriented it is important that there is accurate, realistic force-feedback on the needle. Each type of tissue that the needle comes into contact with should be modeled. These tissues include the skin, intraspinous ligament, ligamentum flavum, paraspinal muscle, epidural space, fat, and bone. Some of these tissue layers may be indiscriminable by the anesthesiologist. The simulator designer must have an understanding of the actual forces, and the limits of detectability by the user in order to know which forces need to be simulated. Some

of these tissues, such as bone, are impenetrable by the level of force that is normally applied and thus the interface should stop the needle movement when this cue is reached. In addition to haptic feedback to the needle, using the loss of resistance technique with a saline or air filled plunger commonly identifies the epidural space. Therefore, a realistic simulator should model the plunger release.

The resident also depends on visual feedback to identify the relationship between the orientation of the needle and the patient while inserting an epidural needle. The resident watches the distance the needle has been inserted into the back of the patient by seeing how much is left outside the patient. In the simulation, shadows and 3D imaging can be used to cue the user as to the angle and depth of the needle with respect to the back. Visual cues can also be used to represent size, age, and orientation of the patient. In actual practice, patient movement is sensed by a combination of visual and haptic feedback through movements of the needle. These cues must be integrated into the simulation.

Additional cues are available through audio feedback. For example, a patient may moan, say "ouch", cry, or yell when the needle is placed in an unanesthetised location or if too much pressure is placed on a spinal bone. There is also a cacophony of noise in the room including machines, other patients, and the hospital staff going about their normal routine.

If the needle placement part of the epidural procedure is to be simulated then the tactile feedback must also be taken into account. For needle insertion itself the tactile feedback is of little importance, as this task is motor-dominant. However, palpation of the back for needle placement is dominated by tactile feedback and

thus simulating this task would require a tactile feedback interface. Additional interfaces could be presented to the user for the purpose of knowledge acquisition, such as interactive MRI views and see-through three-dimensional back images.

Of all of the interface modalities, I believe that the haptic interface is the most important for simulating the epidural needle insertion technique. Vision, smell, and hearing are purely sensory; however, the haptic system can both sense and react to the environment. The addition of haptic feedback to a simulation can add enormously to the quality of the immersive experience for the user. As Durlach said, it is likely that much greater immersion in virtual environments can be achieved by a combination of visual and haptic interfaces than with even a very high-resolution visual interface alone. [Durlach, 1995]

2.4 The Epidural Simulator, circa 1995

2.4.1 Overview

Our team of scientists at the Ohio Supercomputer Center (OSC) and the Department of Anesthesiology at the OSU Medical Center has developed a prototype computer-based simulation system for use in training anesthesiology residents in the technique of placing a needle for an epidural block. [McDonald, 1995. Stredney, 1996] The needle insertion simulator combines real-time graphics, a voice interface, and an Impulse Engine 1000 [Immersion Corporation, San Jose, CA] which provides force-feedback to a Tuohy needle to simulate a realistic epidural experience.

Other groups have worked on the development of a simulator for the haptic lumbar puncture. The focus of Bostrom's group was to develop an appropriate

force-feedback device for needle insertion. Their hardware design was published in 1993 along with a description of their early software algorithms, but I have not seen any papers describing further developments. [Bostrom, 1993]

A group at the University of Colorado's Center for Human Simulation is also working on needle insertion simulation, [mentioned in Reinig, 1994] but their work is unpublished. Recent discussions with them indicated that they are interested in measuring needle insertion force to develop realistic force-feedback models but have not begun this research in earnest.

Gorman, et. al., presented a paper at the most recent MMVR conference describing a prototype haptic lumbar puncture simulator that they are developing. Their simulator uses a PHANToM force-feedback device hidden by a mannequin. They model six tissue layers as the needle is inserted into the virtual patient. The layers are modeled as thin haptic boxes with force parameters of surface friction, puncture force, internal friction, and damping. These parameters were changed interactively via slider bars until they found values that they felt were appropriate for each layer. [Gorman, 2000]

For the OSC epidural simulator, the user of the simulator sits in front of a workstation [Silicon Graphics, Inc. (SGI), Mountain View, California] and invokes the virtual-epidural program through a voice interface. A computer-generated image of the back, developed from Magnetic Resonance Imaging (MRI) data, appears on the screen along with an image of the needle. The user places his hands on the physical needle, which is held by the 1-Degree of Freedom (DOF) force-feedback device that only allows the needle to be moved along a single trajectory. As the

physical needle is pushed, the graphical representation of the needle moves in synchrony, and realistic forces are applied by the Impulse Engine to give the user the feel of inserting the needle into a patients back. The user can get feedback as to the position of the needle by calling up an interactive view of the needle in a image showing the axial (perpendicular to the long axis of the body), sagittal (parallel to the long axis of the body from anterior to posterior), or coronal view (parallel to the long axis of the body from side to side) of the MRI data of the patient's back.

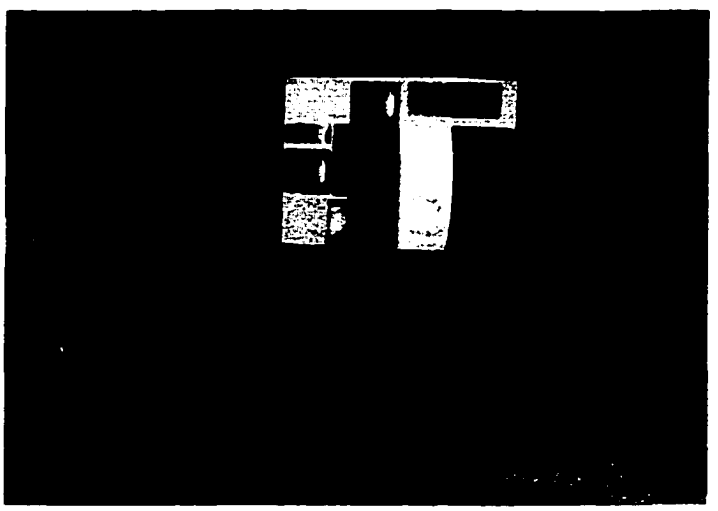

Figure 2.10: A user sitting in front of the epidural simulator.

The simulator, shown in Figure 2.10, consists of three user interface components: the graphical interface, the haptic interface, and the vocal/audio interface. These components will be described in detail next.

2.4.1.1 Graphical Interface

The main graphics window contains an image of the patient's back and the epidural needle. This window is used to simulate what the user would see while doing an epidural; the needle is inserted into the virtual patient's back in conjunction with movement of the physical needle. Additional graphical displays can be displayed through user commands in order to facilitate learning of the regional anatomy. These displays include:

- A transparent three-dimensional view of the layers beneath the skin.
- An interactive axial MRI image displaying the needle moving in the axial plane defined by the location of the needle.
- A set of three static MRI images, which shows the needle in either the axial, sagittal, or coronal plane with respect to the tip of the needle. While these images are displayed the movement of the simulation is paused and the physical needle is locked in place since the images are intended to be used only as static references. The simulation is resumed when the user gives the command to exit from this display.

The graphical displays are all based on a MRI data set that is read in as an input to the epidural simulator. This allows the virtual patient displayed to be changed simply by changing the MRI data set used as input.

2.4.1.2 Haptic Interface

The haptic interface is comprised of a physical epidural needle that is inserted into a 1 DOF Impulse Engine. The Impulse Engine is a device developed by Immersion Corporation to measure the depth of the needle penetration and provide

resistive feedback to the needle. As the needle is inserted into the Impulse Engine, a programmable motor drives back against the shaft holding the needle. The simulator sends motor commands to the device that correspond to the force curves defined for the epidural procedure.

The primary input-output variables of haptic-interfaces are displacements and forces. Because there is a direct relationship between these two variables, either can be used as the control variable and the other as a display variable depending on the force-feedback algorithms that are used. [Durlach, 1995] In this simulator, the current position of the needle (displacement) is used as the control variable for the simulation. The force output is calculated from a lookup table based on current needle position. The visual display is also updated based on the needle position.

The haptic interface is controlled by a personal computer that is in turn controlled through an Ethernet by the graphics workstation. The workstation sends commands to the PC to initialize the Impulse Engine, to return the position of the needle, and to terminate execution. After initialization, the PC program runs in a loop reading the encoders of the Impulse Engine to get the needle position, setting the magnitude of the force-feedback based on the needle position, and sending the position information to the workstation upon request. This loop is terminated only through a command from the workstation. A block diagram of the simulator is shown in Figure 2.11.

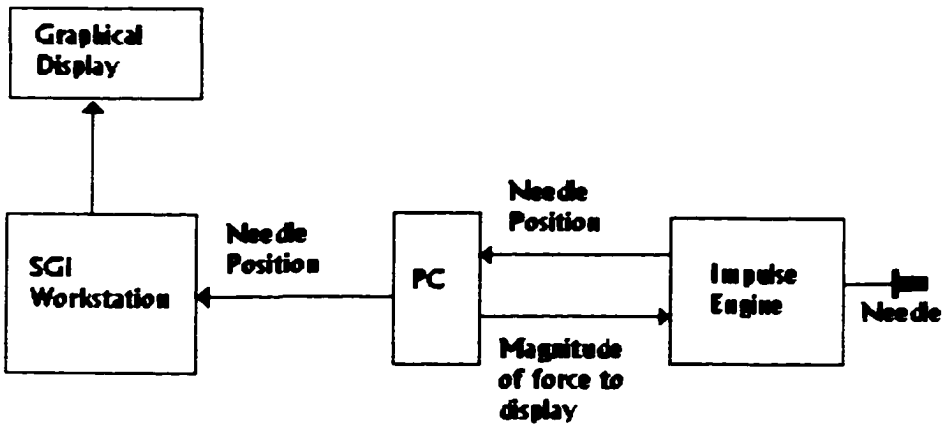

Figure 2.11: Block diagram of the epidural simulator.

The magnitude of the force-feedback in the epidural simulator was originally established through a lookup table based on the depth of the needle penetration. The magnitudes of the forces were defined by expert evaluation through a trial and error process. Several lookup tables of varying force magnitudes and widths were created and experts were asked if the forces were too hard, too soft, too short, or too long for a reasonable simulation of skin and ligamentum flavum puncture. This expert trials were performed under Protocol 94H0228 [Appendix A]. A formal study was not performed to identify the most acceptable force model. The force model for needle insertion was the refined lookup table that was generally accepted across the expert pool to feel like the "typical" patient. The force required to withdraw the needle was not simulated.

2.4.1.3 Vocal/Audio Interface

Communication between the user and the simulator can be established through either the vocal/audio interface, or through a standard keyboard ASCII display interface.

The user can control the simulation using vocal commands such as "Run epidural simulator", and "Display interactive MRI". The simulator communicates with the user through the speaker with questions such as "Which view would you like to see: axial, sagittal, or coronal?" Vocal commands are implemented using speech software [SGI]. The audio interface is implemented using system calls to the *playaiff* program [SGI], which plays audio files through the audio hardware.

The vocal/audio interface offers an advantage over the keyboard/ASCII interface because it allows the simulation to be controlled hands-free. Thus the hands of the user can remain on the physical needle and the haptic experience will be uninterrupted.

2.4.2 Needle insertion technique on the simulator

The simulator currently only implements the midline technique for epidural needle insertion. This technique was easier to implement on a simulator than the paramedian technique. In the midline technique needle angle only changes a small amount in one plane. Using this as a simplifying assumption, the midline technique can be modeled by setting the needle angle. The force-feedback device is a 1-DOF device, so the trajectory of the needle insertion is fixed to the forward direction, not allowing the user to set the needle angle.

In simulating the paramedian approach there are two major challenges for accurate simulation of the technique. The first is that the simulator must be able to model different angles of approach to the epidural space. This requires a device that is capable of sensing position in 3-DOF, though the force-feedback could possibly be limited to the axis of needle insertion. The second challenge is that the simulator must be able to model bone. On hitting a virtual bone surface, the needle's advancement must be able to be halted as if the needle has hit a hard surface. Neither of these challenges is easily met, but both are attainable within technological limitations.

In order for the simulator to handle general cases (calcified ligaments, misplacement of the needle) and not be just a simulation of a correct midline approach it must be able to model bone and multiple angles of approach. The normal-patient midline approach is a good way to start development of an epidural simulator but the finished version should handle these difficulties, making modeling the paramedian approach a natural extension. The simulator must be able to model forces very precisely, including the force required to penetrate skin, intraspinous ligament, ligamentum flavum, paraspinal muscle, fat, blood vessels, and bone. Both techniques also require the ability to attach a plunger to the needle and simulate loss of resistance and plunger pressure as applied to the various tissue types.

2.4.3 Results of user feedback study in Fall, 1995

A user survey was done at a hands-on display in a booth at the national meeting of the American Society of Anesthesiologists (Atlanta, GA October 23, 1995) and in the Anesthesiology resident's lounge at The Ohio State University

Hospital on December 8, 1995. This study was performed under Protocol 94H0228 [Appendix A]. The simulation did not include the vocal/audio display, as both environments were very noisy. The survey was developed to obtain formative evaluation of the simulator. The questions required the respondents to rate the simulator's visual, haptic, and overall effectiveness on a 10 to 1 scale (10 = Very Good, 1 = Very Bad). The overall ratings for the simulator were good, with an average score for all questions of 7.12. Summary statistics from the survey are presented in Table 2.1. [Hiemenz, 1996]

Question	N	Mean	StDev	Q1	Median	Q3
Rate the visuals for adequately performing the task.	90	7.64	1.95	6.73	8	9
Rate the usefulness of the MRI section images in locating the tip of the needle.	90	7.51	1.96	7	8	9
Rate the comfort of the manual interface.	88	6.78	2.18	5	7	8
Rate the fidelity of the haptic sensations in providing spatial information.	90	6.77	2.29	5	7	8.25
Rate the accuracy of the haptic sensations as compared to those experienced in actual practice.	87	6.10	2.32	5	6	8
Rate how straightforward the system was to use.	92	8.07	1.80	7	8	10
Rate this system as compared to a mannequin or model.	82	7.33	2.36	5.75	8	9

Table 2.1: Summary statistics from user trials of the epidural simulator in Fall, 1995. Q1 = First quartile. Q3 = Third quartile.

2.5 Discussion

Although there is definitely work still to be done in the development of the epidural needle insertion simulator, the positive results from the user trials on the 1995 version of the system indicate that it could become a valuable tool for training anesthesiology residents.

The weakest scores received during the user survey were related to the fidelity and accuracy of the haptic feedback. At the time, the haptic forces were being modeled from a combination of the MRI intensity values and expert opinion. Although creating force-feedback models through expert opinion is quick and straightforward there are severe limitations with the resultant model. Since the model is based on personal opinion there is a wide variance across experts and even within the same expert on different trials. The result is a model that is generally accepted across the expert pool to feel like the "typical" patient. The model created is for the "perfect force trajectory" which is only valid for a single needle insertion trajectory and a single class of patients.

My hypothesis is that more realistic and flexible force models can be created through biomaterials testing, by measuring actual needle penetration forces and then creating force models from those measurements. The next phase of this research focused on the development of force models based on data collected through biomaterials testing and the integration of these models into the next version of the virtual reality epidural needle placement simulator. Force models were developed for each layer traversed by the epidural needle. The force-feedback on the simulator is driven by this set of equations and a segmented MRI data set.

These models offer several advantages over models derived from expert opinion. Since they are derived from measured data they are more accurate. The output of the force-feedback device can be compared against the measured data, thus these models are verifiable. Patient variance can be incorporated into the simulation by developing multiple virtual patients from MRI's of different people. The

development of these force models is described in detail in Chapters 3 and 4 of this report.

The field of virtual reality is a new, rapidly growing area of simulation technology. The next decade should be very exciting as interface technology continues to improve, the process of developing haptic models becomes better understood, and medical simulations become a standard in the field of medicine. The research performed for this dissertation will help to create realistic models for needle insertion. As the technology continues to develop, training residents to perform the epidural needle insertion technique using a computer-based simulator will become an accepted step in the educational process.

CHAPTER 3

FORCE-FEEDBACK MODELING I: OVERVIEW & BIOMATERIALS TESTING

3.1 Introduction

The goal of this research is to create realistic feeling force-feedback models for the epidural needle insertion procedure that can be used to drive the force-feedback device in a virtual reality epidural simulator. A force-feedback modeling methodology has been defined based on MRI data and needle puncture force measurements from biomaterial specimens.

This chapter begins with a literature review of previous work done in the area of force-feedback devices, haptic rendering methodologies, and needle puncture measurements. Next, design specifications are presented for the force models that are being developed. An overview of the force-feedback methodology is then presented. Next, details are given for the material testing studies performed to obtain information on needle puncture forces. Chapter 4 presents the force-feedback models for each tissue type, the creation of two virtual patients, and incorporating the models into a simulator for the epidural procedure.

The key aspects of this work are:

- The force-feedback models are based on force vs. displacement data collected from biomaterials testing of analogues for the human back.

- A voxel-based representation of the virtual patient was created from MRI data of the spinal region for two human subjects.
- Forces are represented through curves associated with each tissue type and defined for each voxel with linear interpolation between voxels.

The spinal region is a complex layering of tissues, and the mechanism for puncturing through these tissues is very complicated. However, the models being created need to be able to run at real-time on a force-feedback device and which will feel realistic to the user. In creating these models, several simplifying assumptions are being made. The simulator relies on the non-perfect sensory system of the user and the process of distal attribution to fill in the blanks in the models. Chapter 5 of this dissertation describes user testing done which validates the realism of the created models.

3.2 Literature review

3.2.1 Force-feedback devices

The most prevalent form for the force-feedback devices being used today is a desk-grounded master. These devices sit on a desk, which supports it's weight and keeps the device from slipping or toppling [Burdea, 1996]. Two common configurations are joysticks, with force-feedback applied to the joystick handle (such as the laproscopic Impulse Engine, Immersion Corp., San Jose, CA, Figure 3.1, right), and pen-based interfaces (such as the PHANToM, SensAble Technologies, Cambridge, Massachusetts, Figure 3.1, left). These devices have motors that can be programmed to produce different levels of force output on the joystick or pen. A

good description of these device configurations and others can be found in Burdea. [Burdea, 1996]

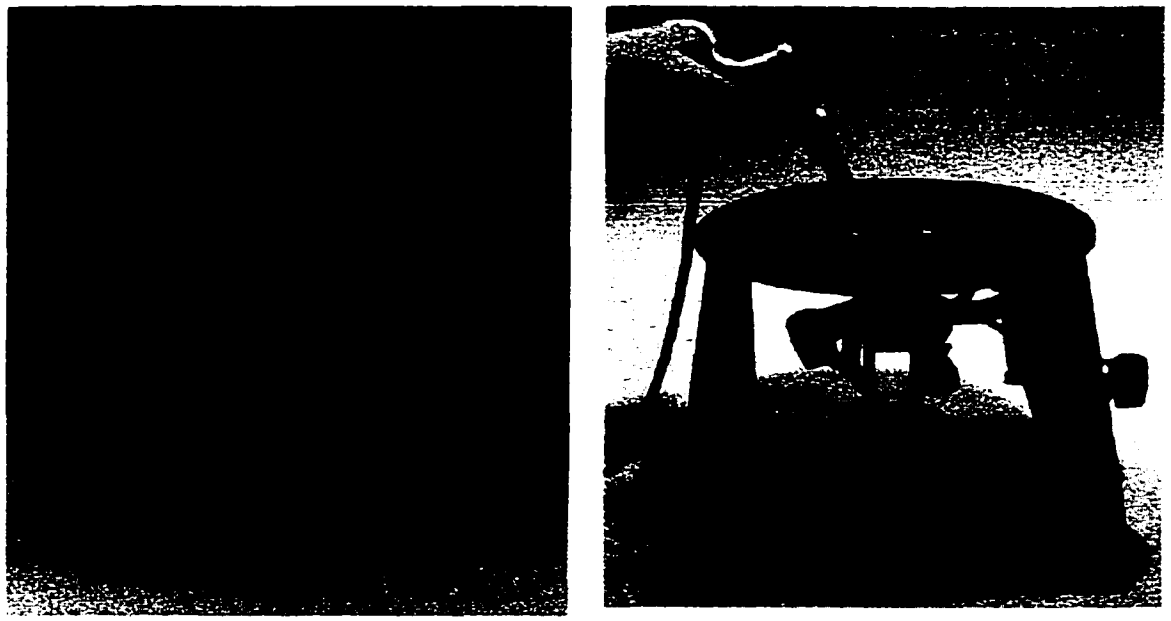

Figure 3.1: Examples of two force-feedback devices. The SensAble PHANToM on the left [SensAble, 1999] is an example of a pen based tabletop device. The Laproscopic Impulse Engine on the right [Immersion, 1999] is an example of a joystick based tabletop device.

Massie defined three criteria that are necessary for a force-feedback device to be able to give realistic haptic sensations. The first is that *free space must feel free*. In other words, the device should move freely when it is not interacting with virtual objects. The second criterion is that *solid objects must feel stiff*. This is related to the stiffness of stable control that the device can achieve, and no current devices are able to reproduce a virtual wall accurately. The third criterion is *that the*

virtual constraints must not be easily saturated. In other words, the user should not be able to overpower the device and push through a solid object. [Massie, 1994] None of these criteria are fully achievable with current hardware technology, thus simulator designers must understand the limitations of the hardware and design systems with these limitations in mind.

Minsky studied control issues in force display. She found that the minimum sampling rate needed for good perception of force is related to "human response time, human arm dynamics, system performance, and the object or substance being displayed." She states, "If the system is inherently unstable, the illusion of a real object is destroyed immediately." These instabilities can show up as jitters in the force display and can be caused by noise in the display device or bad system dynamics related to the sampling period. This is the root cause in the difficulties with modeling contact with a hard surface. [Minsky, 1990]

3.2.2 Force-feedback modeling

Force-feedback models are commonly created by setting spring constants for virtual surfaces. The value of the spring constants are set based on input from experts as to what the force should feel like. This type of force-feedback model is simple to create and normally results in a pretty good approximation of physical reality. However, there are drawbacks because the models are based on the expert's opinion of how the sensation should feel, which can be heavily biased between experts and even within the same expert on different trials. There is also difficulty in creating polygonal surface models. For medical data, creating these

models can require a significant amount of processing due to the resolution of the images and the complexity of the data sets.

Finite element models have been widely used in engineering to analyze soft-tissue deformations under mechanical constraints. These models tend to require large amounts of computation time, although much work has been done recently to develop real-time finite element modeling techniques. A good description of soft-tissue modeling techniques for medical simulations can be found in [Delingette, 1998].

Blezek and Robb have developed a method to create force-feedback models directly from medical image data. Their method uses a ray from the surface contact point of the volume to the current position to get the exact intersection point. They then set the force-feedback based on the brightness of the voxel at this point. Smoothing can be done by averaging the current voxel's value with that of its neighbors. They found that the algorithm worked well for quadratic implicit surfaces and bone surfaces from CT data, but it does not do well with soft tissues and more complex surfaces. They are working on improved smoothing algorithms and surface interpolation algorithms to better extract the surface from the volumetric data set. [Blezek and Robb, 1999]

Researchers are beginning to identify the need to take force measurements for improved force-feedback modeling in surgical simulations. Rosen is using an instrumented endoscopic tool to measure force/torque signatures during endoscopic procedures. [Rosen, 1998] Bicci's group is collecting data on tissue elastic

properties to improve force-feedback modeling for laproscopic surgical simulators. [Bicchi, 1996]

3.2.3 Resolution required for the force-feedback models.

The movement of the epidural needle is a motor-dominant precision manipulation task. The perception of force-feedback relies mainly on kinesthetic information. [Durlach, 1995] *Kinesthesia* is defined as "a sense mediated by end organs located in muscles, tendons, and joints and stimulated by bodily movements and tensions." [Merriam Webster's, 1994]. Because the user is both sensing and controlling the force-feedback, the specifications for the force-feedback interface are directly dependent on the abilities of the user to perceive and control the position and force from the force-feedback device.

Tan, Pang, and Durlach out of MIT have published several studies on the manual resolution of length, force, and compliance for the pinching motion of the forefinger and thumb. These studies were conducted using single interval forced-choice discrimination. The subject was asked to answer whether the stimulus they felt was the reference signal, S_1, or the reference signal plus an increment, S_2.

Results of the user trials were placed in a 2x2 response matrix, which was then processed to get estimates for the sensitivity index and the response bias. When the bias is negligible compared to the sensitivity index the just noticeable difference (JND) results can completely characterize the data. They found that this was the case for length discrimination and force discrimination with a fixed displacement.

For length discrimination, they found that the JND for length is approximately 1 mm for a reference length of 10 mm and increases to 2.4 mm when the reference length increases to 80 mm. This is inconsistent with Weber's Law that states that the JND is linearly proportional to the reference length.

For force discrimination Tan, et. al., found that the JND is 5-10% of the reference force, follows Weber's Law (the JND is linearly proportional to the reference force), and is independent of the initial finger span. These results are for a fixed distance where the force always occurs at the same location. It was found that if the distance changed in each trial the force JND was significantly increased. [Tan, 1992]

In Burdea's book he defines sensing bandwidth as the frequency that tactile stimuli are sensed and control bandwidth as the rapidity with which humans can respond. He summarizes studies in which Shimoga measured human sensing and control bandwidth. Results from these studies are summarized in Table 3.1. [Shimoga, 1992] Burdea also mentions that the force resolution should be greater than 0.1N to keep users from feeling jumps in the forces. [Burdea, 1996] Thus, the haptic feedback device must be able to display forces with both a fine resolution and a reasonable update rate in order for the forces to be perceived correctly by the user.

Frequency (Hz)	Bandwidth for human finger
1-2	Maximum for response to unexpected force/position signals.
5-10	Maximum for comfortable application of force and motion commands.
8-12	Maximum for correction of positional disturbances.
12-16	Maximum for correction of grasping forces if the grasped object slips.
20-30	Minimum for meaningful perception of force input signals.
320	Maximum bandwidth for discrimination between two consecutive force signals.
5000-10,000	Minimum bandwidth for sensing vibration during skillful manipulative tasks.

Table 3.1: Summary of human sensing and control bandwidth as measured by Shimoga [Shimoga, 1992].

3.2.4 Force measurements during needle insertion

Although needle insertion is an extremely common technique for medicine, a thorough literature search turned up very few instances of research on the mechanics of needle puncture into tissue. For needle insertion, it has not been important in the past to know the magnitude of the force required to puncture tissues. It is only now, when realistic feeling force-feedback models for needle insertion simulation are becoming necessary, that collecting needle insertion force data is becoming useful.

Eriksson measured the penetration forces needed for peripheral venous cannulation to compare polyurethane (Insyte®) and polytetrafluoroethylene (Venflon®) cannulae. He developed a device using a force transducer, a power

supply, and a recorder that printed the force readings on scaled paper at a chart speed of 20 mm/s. He measured forces as two cannulae (one of each type) were inserted into 37 volunteers. He found the skin penetration force to be 3.9±1.6 N for the Insyte® cannulae, and 2.7± 0.8N for the Venflon® cannulae. He found the vein wall puncture to be 3.5 ±1.2N for the Insyte® cannulae, and 2.4± 0.9N for the Venflon® cannulae. [Eriksson, 1991]

The needle insertion studies I have found using epidural needles relate to the path of the needle, the relative puncture force for different types of needles, and the size of the hole left in the tissue. The facets of these studies that are relevant to my research are briefly described next.

Baumgarten performed a study in which he inserted Tuohy and Quincke needles of various gauges into Styrofoam models to look at needle deflection. He drew a line on the top of the Styrofoam and looked at how much deflection there was from this line when the needle tip came out the end of the Styrofoam. He found that when a needle is inserted with the bevel down, there was virtually no deflection. (25-gauge Quinke needle, Mean deflection +/- SE: 0.4 +/- 0.5mm). If the needle was inserted with the bevel lateral there is deflection of 6.7+/- 0.5mm for a 17 Gauge Tuohy needle. He stated that in studies on chuck steak using a 25-gauge Quinke needle deflection was about 25% less. [Baumgarten, 1995]

Sitzman, et. al. performed a study that looked at the path the needle followed as it was inserted into porcine paraspinous muscle that was held in a Plexiglas frame. Introducers were used to provide accurate alignment of the needles. Quinke

needles were inserted with their bevels facing laterally, perpendicular to the muscle fiber. In this study they found that none of five 18-gauge Quinke needles deflected from the initial insertion trajectory as it was inserted up to 60 mm into the tissue. [Sitzman, 1996]

A study by Mieklejohn showed that 1.774N of perpendicular force was required to puncture the dura in vitro for an 18-gauge Tuohy needle without rotating the needle. In this study, he used three dural sacs taken from adult human cadavers. The dura was sectioned into strips approximately 1.5 cm long. The strip was clamped so as to be flat, but not stretched, and placed over a vessel containing Ringer lactate solution. A weighted Tuohy needle was used to puncture the specimen and the weight needed for puncture was recorded. He found that rotating the needle while it was weighted and pressed against the dura caused a decrease in the puncture force. His conclusion was that epidural needles should not be rotated after insertion of the needle tip into the epidural space. [Mieklejohn, 1987]

Towler modified an Instron Universal Testing machine to measure the maximum vertical force necessary to puncture a Medpar 1200 synthetic membrane. In the study, he was comparing two different curved surgical needles. He decided to use an artificial membrane instead of human or pig skins because the physical properties of these skins vary by anatomical site, specimen age, and storage technique. He was interested in studying the relative difference between puncture force of the two needles, and not in the force required to puncture the specimen. [Towler, 1988] In my research, I used a similar test setup to measure needle insertion puncture force.

Westbrook performed a study to compare the force required for different types of spinal needles through *in vitro* bovine dura. He found the puncture force was 2.25 ± 0.29 N averaged across 10 trials for a 16-gauge Tuohy needle. The dura was excised and mounted in a fixture with a watertight seal and a column of cerebrospinal fluid behind the dura. [Westbrook, 1994]

3.2.5 Related research

Despite an extensive literature search, I have only been able to find one study that closely relates to the research I have done. Brett, from the University of Bristol, has also been studying the simulation of resistance force acting on surgical needles. His group has developed a 1-DOF force-feedback device for epidural needle insertion simulation. Tuohy needles were inserted into porcine samples using a tensile machine to obtain force curves for the procedure. A hand-held device was used to measure forces as Tuohy needles were inserted into recently deceased human cadavers. For cadaver measurements, the needle was advanced into the body by pushing on the end of the syringe. Syringe fluid pressure was measured, and the puncture force was derived from this pressure.

Brett created force models for skin, fat, and ligamentum flavum (LF) along with a step force he called skin friction from the measured data. He assembled a force model for the epidural procedure using these models as building blocks. Figure 3.2 shows Brett's resultant force behavior of skin, fatty/loose muscle tissue, and ligamentum flavum, along with the resultant force model as a function of needle displacement.

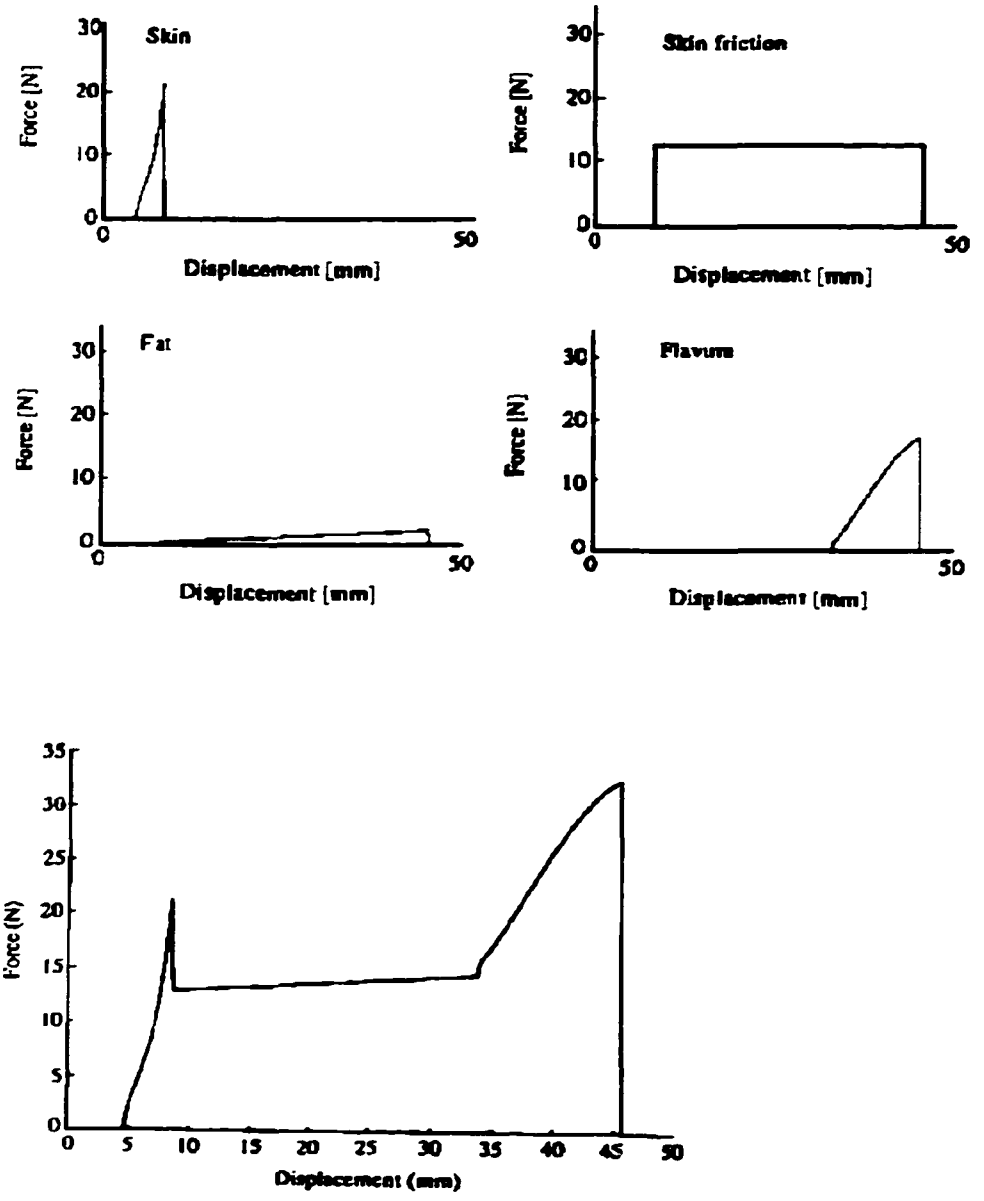

Figure 3.2: The top four figures illustrate the force behavior of skin, fatty/loose muscle tissue, and ligamentum flavum as derived by Brett. The bottom graph shows his resultant force model as a function of needle displacement. [Brett, 1997]

Brett assumed that the puncture force was dependent on needle velocity. He measured force response at different constant needle velocities into the porcine cadavers and found a relationship between the change in the peak force values and the change in velocity. No details of these experiments or the relationship are presented in his paper. He concludes that anesthesiologists that have used his simulator state that it provides a representative sensation of epidural puncture, but that a more extensive study is needed. [Brett, 1997]

3.3 Design Specifications

A. A force-feedback model design methodology will be constructed to create force-feedback models for needle insertion based on force measurements from biomaterials testing studies.
B. The modeling methodology will be able to incorporate differences in patients due to age (thickness of skin, calcification of ligaments, other biomechanical changes in tissue structure), body mass index, and anatomical differences.
C. The force-feedback device will be tested to identify the minimum perceivable force.
D. Force models will be created for each tissue type encountered in the epidural needle insertion procedure. These tissues are: skin, fat, muscle, interspinous ligament (ISL), ligamentum flavum (LF), epidural space, and bone.
E. Assumptions required to create the models will be identified and tested for validity.
F. High resolution MRI data will be used for tissue identification.

G. Material testing will be done to identify the puncture force curves for each tissue. Load vs. displacement data will be collected every 0.1mm.

H. Two virtual patients will be created, an average female and a lean male.

I. Virtual oranges will be created in order to test the realism of the simulation.

J. A haptics-only simulation will be developed for the epidural needle insertion procedure. The simulation will meet Massie's criteria for realistic feeling haptic sensations: free space will feel free, solid objects will feel stiff, and the virtual constraints will not be easily saturated. The system will feel stable while inserting the needle at any speed.

3.4 Developing the force-feedback models.

3.4.1 Force thresholds using the force-feedback device

3.4.1.1 Overview

In order to understand the precision needed for the force models a study was done in December 1996 to identify the smallest perceivable force using an Immersion Corporation Impulse Engine force-feedback device. This device had only 1-DOF of needle positioning and force-feedback. It was the predecessor to the device used for the simulator development described in Chapter 4.

3.4.1.2 Methods

Data was collected for 20 subjects with no prior needle insertion experience completing 60 simulator trials per subject. This study was performed under Protocol 94H0228. [Appendix A] Each trial consisted of one of ten forces being exerted by the force-feedback device with the test subjects responding as to whether or not they could feel the force. The ten forces ranged from no actual force to a readily

perceivable force with the remaining eight forces being equally spaced between the endpoints. The forces were presented in random order, according the method *of constant stimuli.* [Levine, 1981]

The data was analyzed using classical psychophysical methods in order to determine the minimal perceivable force threshold. A good description of these methods can be found in Levine, 1981. The data was first plotted as a psychometric function (shown as Figure 3.3), which is a curve of the percentage detection versus the stimulus force intensity. Next, the data was plotted on a normal probability axes from which it is easier to read the minimum perceivable force. This plot is shown as Figure 3.4.

3.4.1.3 Results

A summary of the data is shown in the Table 3.2. The threshold of detection for the Impulse Engine force feedback device was found to be 0.25N, which is the force at 50% probability.

Stimulus	Force (N)	Yes Response	% Detected
1	0	13	10.8
2	0.079	14	11.7
3	0.158	32	26.7
4	0.238	57	47.5
5	0.317	71	59.2
6	0.36	88	73.3
7	0.439	88	73.3
8	0.518	99	82.5
9	0.6	104	86.7
10	0.677	99	82.5

Table 3.2: Data from minimal threshold of perception study.
[n = 120 for each stimulus]

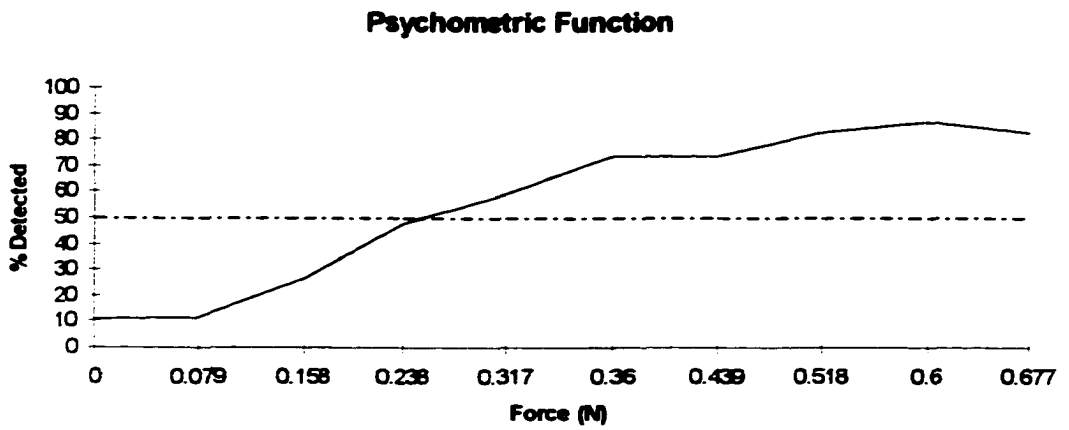

Figure 3.3: Psychometric function for the minimal threshold of perception study plotting the percentage detection vs. the magnitude of the force stimulus. The threshold of detection is at the 50% detected line.

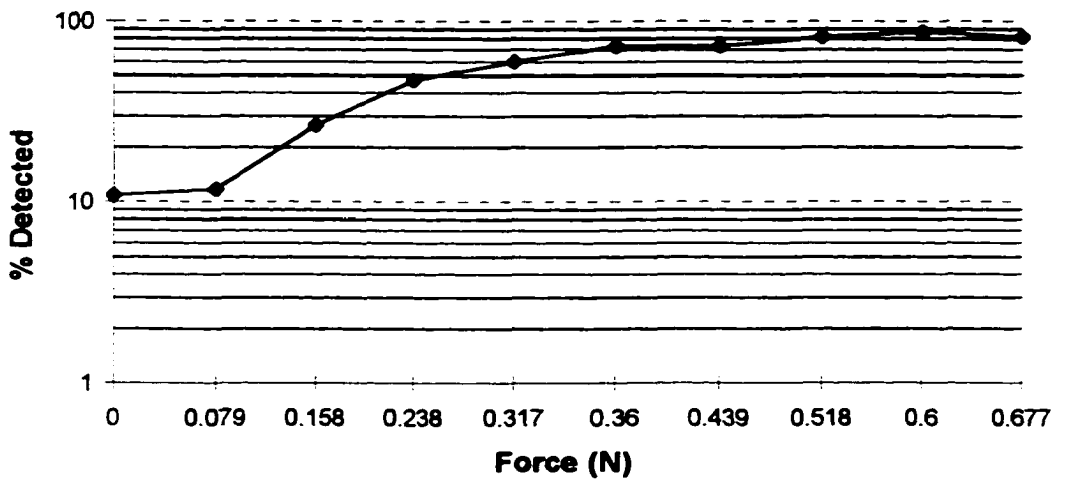

Figure 3.4: Data from the minimal threshold of perception study plotted on normal probability paper. The threshold of detection is the force at 50% probability, 0.25N.

3.4.1.4 Discussion

It is understood that there are limitations with using classical psychophysical methods rather than modern psychophysical methods, mainly that there is no absolute threshold of perception and classical methods attempt to quantify such a threshold. However, an approximation of threshold is all that is needed for development of this simulator because of accuracy limitations from the hardware, and limitations in achievable resolution in the force-feedback models.

3.4.2 Overview of the force-feedback modeling methodology

The force-feedback models for the needle insertion simulator are comprised of equations of penetration force for the individual tissue types overlaid upon a 3D computer model developed from magnetic resonance imaging (MRI) data. In this section I will give an overview of the methodology I developed. [Also published under Hiemenz, 1998] In subsequent sections, each step will be given in detail for each of the subjects used to create the force-feedback model.

For a first approximation for measuring the penetration forces involved in epidural needle insertion, a variety of analogues for the anatomical structures of the human back were tested (gelatin models, oranges and tomatoes, and an animal cadaveric specimen). Several assumptions were made in order to develop the models; these are detailed in section 3.4.3.

The first step to creating the force-feedback models was to obtain high-resolution MRI images of the test objects using a General Electric 1.5 Tesla MRI system. The MRI images were registered with an oil soaked yarn placed in a grid on

the surface of the object (e.g.. orange) in order to identify the internal structure at each location where a needle would subsequently be inserted.

Next, a materials testing system (MTS machine) [Bionix 858, MTS Corporation, Eden Prairie, MN] was used to measure force vs. displacement for the tip of an 18 gauge Tuohy needle as it was inserted into the samples under displacement control.

An example force vs. displacement curve measured using the MTS system on a canine cadaveric specimen is shown in Figure 3.5.

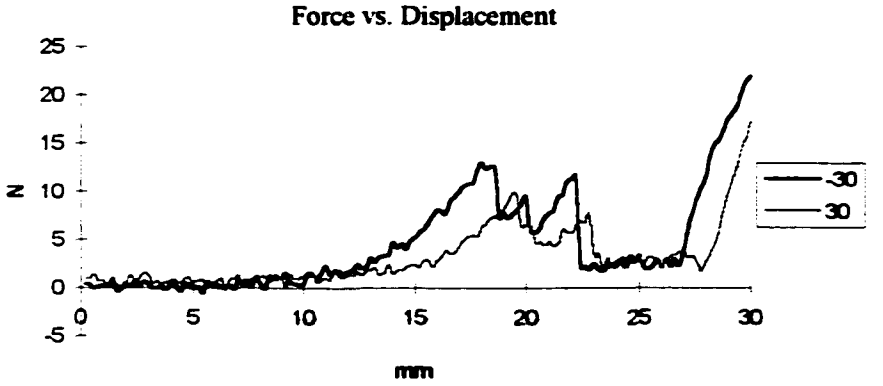

Figure 3.5: Force vs. displacement curve, Canine specimen, 0.2 mm/s insertion speed, 30° and -30° angles of insertion.

Programs to analyze the MTS force vs. displacement curves were developed using Matlab. [The Mathworks, Inc.] The data were filtered using a 7-point moving average filter to remove machine noise from the signal. Section 3.4.4.3.3.2 discusses why this filter method was chosen. Through correlation of the penetration

depth and the MRI image, the curves were then broken down into smaller curves representing the different tissue layers. These normalized curves were then averaged across the trials to create a general force profile for each tissue type. Finally, curve fitting was performed to create the force equation for the given tissue type.

The MRI data set underwent post processing using LEGION, an image segmentation program developed by Shareef. [Shareef, 1996] This program was used to tag each voxel of the data set according to the tissue layer it belonged to. The resultant tagged MRI data creates a "scaffolding" (a 3D computer model) for the force-feedback equations. Each tagged region represents a type of tissue, and thus a force region in which the tissue's force equation will be active. An example of an original MRI image slice and the corresponding segmented image is shown in Figure 3.6.

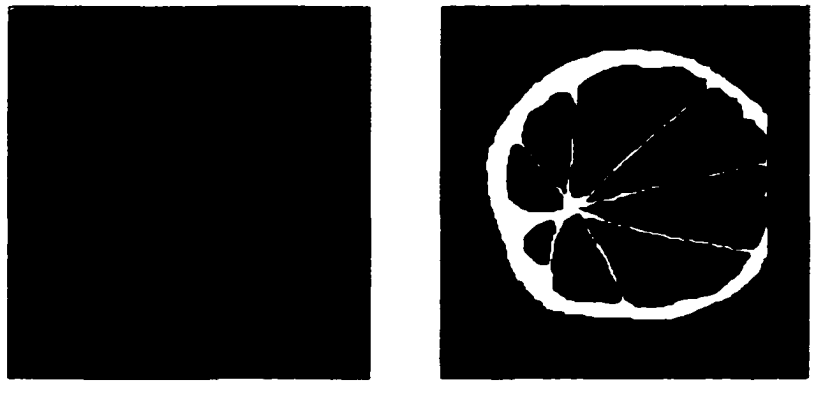

Figure 3.6: Original MRI orange slice (left). Segmented MRI image (right) with three tagged regions corresponding to orange pulp, seed, and rind.

Next, the force-feedback models were incorporated into the needle insertion simulator. A function for the force equation for each tissue type was coded into the simulation program. As the user inserts the needle into the force-feedback device the type of tissue at the tip of the needle is identified from the segmented MRI data set. The needle is modeled as a point, specifically the point of the tip of the needle. The magnitude of the force-feedback at any point along the needle trajectory was then set based on the force equation for the type of tissue at the tip of the needle, and the depth of the needle tip into the particular tissue layer.

3.4.3 Assumptions

In order to create a model of the forces felt at the tip of the needle during the epidural needle insertion procedure several assumptions were made regarding the mechanics of the needle puncture. The assumptions were:

1. *Puncture force is independent of insertion speed.* The forces felt at the needle tip are independent of the insertion speed over the range of speeds typical of inserting an epidural needle.
2. *Skin puncture force is independent of the initial insertion angle.* The difference in forces obtained during initial skin puncture due to angle of insertion is immeasurably small.
3. *Muscle puncture force is independent of the initial insertion angle.* The difference in forces obtained due to angle of insertion with respect to the muscle fiber is immeasurably small.

4. *Puncture forces are axial.* The puncture force is purely axial along the trajectory of the needle.
5. *The trajectory of the needle can be identified prior to needle insertion.* There is no deflection in the trajectory as the needle is inserted.

Assumptions 1, 2, and 3 were tested during the pilot studies described in section 3.4.4, and found to be valid. Assumption 4 was not tested because it is not possible to change the angle of insertion after the needle is inserted a short distance into the body; however, I believe it is valid to assume that the needle movement is constrained to a line along the trajectory of insertion and therefore puncture forces are axial along this trajectory. I believe that there is a drag force associated with friction from the tissues that the needle has passed through. Because I am measuring and modeling the forces at the tip of the needle, these forces are intrinsically a part of the model.

Two studies are described in Section 3.2.4, by Baumgarten and Sitzman, which look at whether there is any deflection in the trajectory of the needle as it is inserted (Assumption 5). Sitzman's results show that there is no deflection on an 18-gauge Quinke needle as it is inserted up to 60mm into porcine muscle. This implies that in my study I can assume that the needle will stay on the initial trajectory after it has been inserted. Baumgarten's results show some evidence that assumption number 5 may not be a good assumption; however, he did not present results for insertion of a 17-gauge Tuohy needle into chuck steak. This needle is significantly bigger and stiffer than the 25-gauge Quinke needle he tested and may have significantly less deflection in chuck steak than in Styrofoam. In my research, I

disregarded any change in the needle trajectory due to needle bevel placement. A future topic of research would be to perform a study similar to these two studies but focusing on epidural needles, and using a realistic analogue for human back tissues.

3.4.4 Pilot Studies

The first step of this project was to perform initial biomaterials testing, test model assumptions, and solidify the force modeling methodology. Biomaterial testing was performed on four material specimens: gelatin models, oranges, tomatoes, and a canine cadaveric back section. Each study is described in detail next.

3.4.4.1 Identifying the needle insertion speed

As part of the human factors analysis, twenty epidural procedures were videotaped and analyzed in 1997. This study was performed under Protocol 94H0333. [Appendix A] Experts as well as new residents learning the epidural procedure at the OSU Hospital Department of Anesthesiology performed these procedures. By using a stopwatch and watching the 1cm markings on the needle as it was inserted, the range of needle insertion speeds were found to range from 0.4 to 10 mm/s (millimeters/second) for the epidural needle insertion procedure. The slower speeds are associated with the final penetration of the LF. The faster speeds are associated with the initial skin penetration.

3.4.4.2 Gelatin study: MRI correlation to force

3.4.4.2.1 Overview

This study was used to investigate the relationship between MRI parameters and material properties. [This section is summarized from Hiemenz, 1997] The goal

was to identify whether force models could be directly created from the MRI data itself. There is limited research investigating the relationship between MRI imaging parameters and material properties. Chu and Rutt show that by changing the mechanical properties of a polyvinyl alcohol cryogel the T1 and T2 relaxation times are changed. [Chu, 1997] A proportional relationship has also been found between viscosity and T1 relaxation times for a variety of liquids. [Abragam, 1966]

Two hypotheses were tested in this study:

Hypothesis 1: Over the range of insertion velocities typical for epidural needle placement the force at the tip of the needle is independent of the insertion speed.

Hypothesis 2: For each type of test sample, the puncture resistance is related to MRI imaging parameters, T1 and T2 relaxation times. (These parameters describe proton longitudinal relaxation time and transverse relaxation time respectively in response to a magnetic field).

3.4.4.2.2 Methods

Five solid samples were made using Knox™ unflavored gelatin mixed with distilled water in the following concentrations: 70.9%, 47.3%, 36% 15.8% and 5.3% gelatin by weight. As a comparison, the recipe for gelatin of edible consistency calls for 3.5% gelatin by weight. Two gelatin analogues of the tissue layers in the lumbar region of the back were also created. These analogues were made by layering [47.3% 5.3% 70.9% 5.3%] and [36% 5.3% 70.9% 15.8%] gelatin concentrations.

High-resolution MRI images were obtained for each of the samples tested using a General Electric 1.5 Tesla MRI system. An inversion recovery sequence was used to measure the T1 relaxation times. T2 relaxation times were determined using a spin echo sequence. GE publishes a general reference for MRI scanning which describes these concepts. [Keller, 1988]

The MTS machine was used to measure force vs. displacement curves for the tip of an 18 gauge Tuohy needle as it was inserted into the samples under displacement control with continuous recording of load (on a 200 N scale) and displacement (on a 100 mm scale). Forces were recorded for each sample at three penetration rates: 0.1 mm/s, 1 mm/s, and 10 mm/s. The modulus of compression was also obtained using a 1-inch diameter solid steel compression fixture. A 7-point equally weighted moving average filter was used to remove noise from the data. Section 3.4.4.3.3.2 discusses why this filter method was chosen. Linear regression was performed using Minitab. [Minitab, Inc., State College, PA]

3.4.4.2.3 Results

Visual inspection of the data curves showed no differences in the force displacement curves for the three insertion speeds. Results are summarized in the following table:

%Gel (g/g)	PR (N/mm)	CM (N/mm^2)	T1 (msec)	T2 (msec)
70.9	0.167	9.306	266.8	35.7
47.3	0.067	5.308	434.2	115.5
15.8	0.021	1.136	751.1	240.9
5.3	0.003	0.192	1153.4	737.2

Table 3.3: Data summary for the gelatin study.

The puncture resistance (PR) and the modulus of compression (CM) were found to be linearly related, as shown in Figure 3.7.

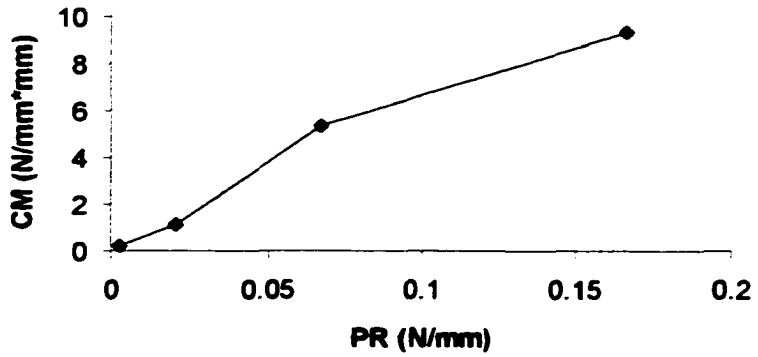

Figure 3.7: Graph showing the linear relationship between the puncture resistance and the modulus of compression for the gelatin study.

The difference in needle penetration force required to puncture each layer of the layered samples was clearly identifiable. The penetration force was found to be a linear function of the water content following the equation:

$$PR = -0.018 + 0.00237 * (\%\ gel) \qquad R^2 = 0.932$$

An exponential relationship was found between the puncture resistance and both the T1 and T2 relaxation times from the MRI study. Stiffer gelatin mixtures have larger resistance to puncture and shorter T1 and T2 times than more compliant mixtures. This relationship is illustrated in the Figure 3.8.

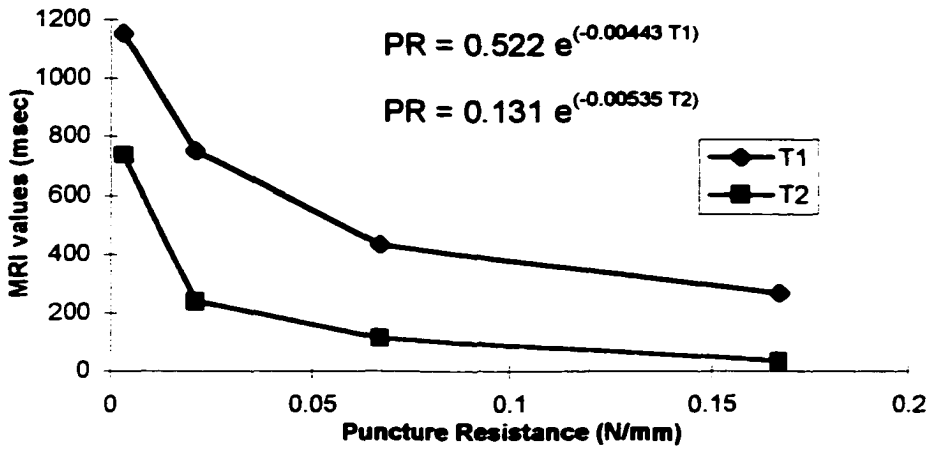

Figure 3.8: Relationship between puncture resistance and T1, T2 relaxation times.

3.4.4.2.4 Discussion

These experiments with gelatin analogues for tissue were the first step in creating the force-feedback methodology. Gelatin was chosen because the samples

were homogeneous and layered analogues were easy to create. The study indicated that it is possible to use the MTS machine to measure small changes in the needle penetration resistance due to puncturing different materials. It also gave evidence that there is no difference in the force curves due to changes in insertion speed in the given range of speeds.

The exponential relationship found between MRI values and both the puncture resistance and the compressive modulus allows the force models for the gelatin samples to be based on their MRI values. Further experimentation (the canine study described in section 3.4.4.3) showed that this relationship did not follow for tissue samples because the puncture force is dominated by puncturing the fascial layers between tissue layers, not the tissues themselves. However, further study could be warranted in looking at tissue compression and MRI values.

3.4.4.3 Oranges/Tomatoes

3.4.4.3.1 Overview

The next pilot study, performed on April 10-11, 1997, used oranges and tomatoes as the test specimens. The hypotheses under test were:

Hypothesis 1: Forces are measurable using the MTS machine.

Hypothesis 2: Force differences due to change in angle of needle penetration are negligible.

Hypothesis 3: There are no significant differences in needle penetration forces between 0.1mm/s, 1mm/s, and 10 mm/s rates of needle insertion.

In addition, this data was used to identify the best method of filtering the data in order to remove noise added in during the data collection process from the MTS system.

3.4.4.3.2 Methods

Four of each fruit were used in the study. Acrylic boxes were built to hold the specimens, two oranges and two tomatoes per box. The top of each box had a 3 x 5 grid of holes drilled into it above each specimen at 1cm intervals to use as a guide for the needle insertion. When the fruit was placed in the MRI machine the boxes were filled with ice water to minimize temperature effects and so that the holes were visible for registration of where the needles would be inserted. The water was removed from the boxes before they were placed on the MTS machine.

During the materials testing, the needle was inserted into the specimen at three different speeds: 0.1mm/s, 1mm/s, and 10mm/s. The initial needle insertion angle varied due to the shape of the fruit. The needle insertion direction was always straight down (perpendicular to the floor, 90°), so the initial insertion angle varied from 50° to 130°. This angle was identified during post processing by looking at the MRI images.

3.4.4.3.3 Results

3.4.4.3.2.1 Hypothesis 1

Hypothesis 1 was validated through visual inspection of the data. The load displacement curves from the MTS machine clearly showed the differences in puncturing the rind, the pulp, and hitting seeds inside the orange.

3.4.4.3.3.2 Identifying the filter method

The initial filter method used was a 7-point moving average filter based on the equation:

$$smoothedData(n) = (oData(n-3) + oData(n-2) + oData(n-1) + oData(n) + oData(n+1) + oData(n+2) + oData(n+3)) / 7$$

where oData = original Data, and n = 0 ... (number of data points).

Figure 3.9 illustrates the effect of the 7-point moving average filter. The noise is smoothed from the trial data, but most of the key features are still present in the filtered data.

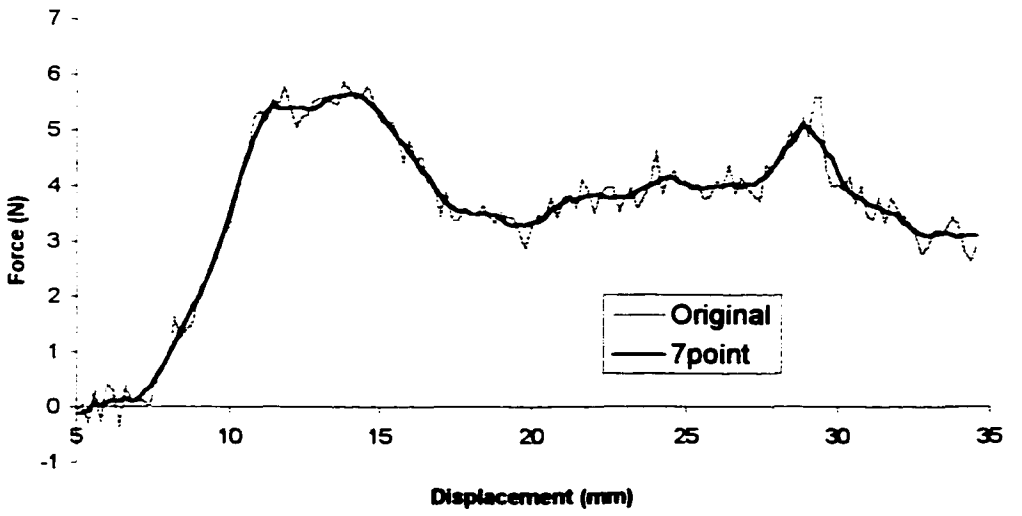

Figure 3.9: Graph of original orange trial data and the corresponding 7-point moving average filter. The noise is smoothed from the trial data, but most of the key features are still present in the filtered data with this filter.

After initial filtering of the data, I re-evaluated the use of 7-point moving average filter method. The filter tool of Peak Motus [Peak Performance, Inc., Englewood, CO] was used to look at three other well-known filter techniques: Butterworth, Cubic Spline, and Fast Fourier Transform. The filter parameters and representative graphs for each filtering method showing the original data and the filtered data for a medium speed orange data trial are presented in Figures 3.10, 3.11, and 3.12. None of these filters were significantly better than the original filtering method. Because the 7-point moving average filter was easy to understand and implement, I decided to remain with this filtering method.

The Butterworth filter (4^{th} order, zero lag, recursive) had a cutoff frequency of 2.0, with a prescribed limit of 0.1000 for optimization by the Jackson Knee method. As shown in Figure 3.10, the Butterworth filter does not smooth much noise from the original trial data.

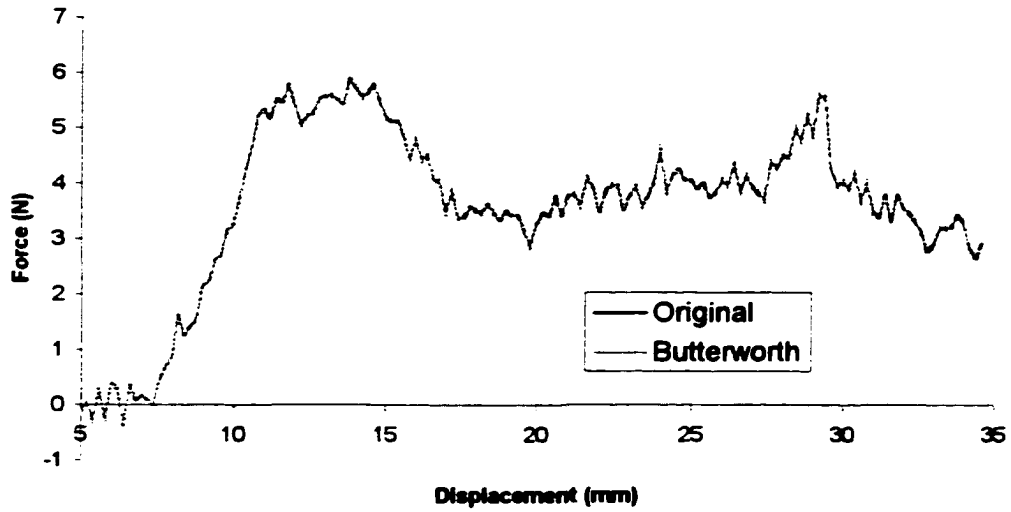

Figure 3.10: Graph of original orange trial data and the corresponding Butterworth filtered data. Not much noise was smoothed from the original trial data

The cubic spline filter used 4 passes, with a prescribed limit of 0.2000 for optimization by the Jackson Knee method. As shown in Figure 3.11, the cubic spline filter smoothed much of the detail from the original trial data.

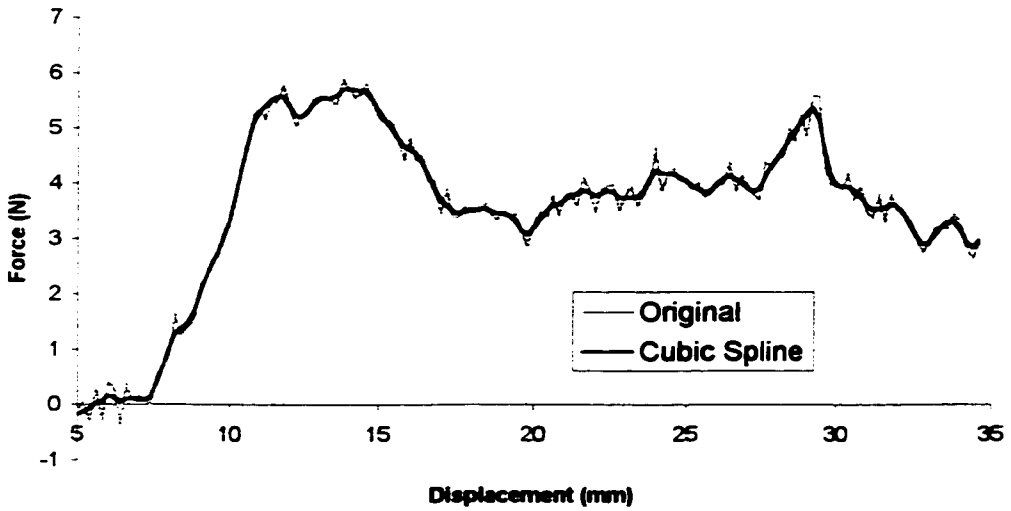

Figure 3.11: Graph of original orange trial data and the corresponding cubic spline filtered data. With this filter, much of the detail is smoothed from the original trial data.

The Fast Fourier Transform filter had the window size was set to 5, and a prescribed limit of 0.0500 for optimization by the Jackson Knee method. As shown in Figure 3.12, the Fast Fourier Transform filter smoothed much of the detail from the original trial data.

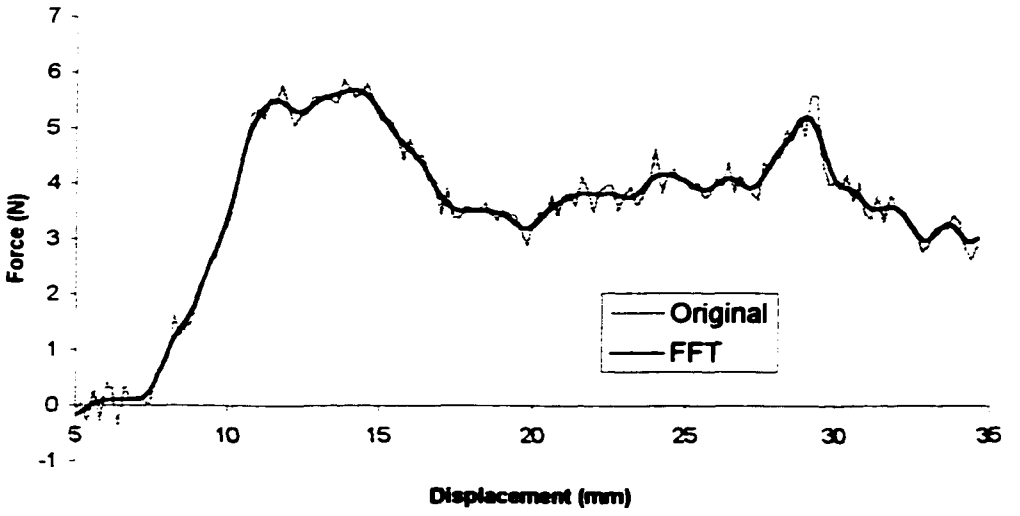

Figure 3.12: Graph of original orange trial data and the corresponding Fast Fourier Transform filtered data. With this filter, much of the detail is smoothed from the original trial data.

3.4.4.3.3.3 Hypothesis 2

To test Hypothesis 2, data was analyzed to look at the relationship between the angle and the initial puncture force. The slope of the initial puncture (for the rind) and the thickness of the rind were used to describe the initial puncture force. For this analysis, I used all of the trials from the four oranges at the medium insertion speed (1mm/s). The regression equation was:

$$punctureForce = -1.00 + (1.14)*distance + (0.326)*thickness + (0.00122)*angle$$

The p-values indicate that the angle is not significant, $p_{angle} = 0.321$, but the needle penetration distance and the thickness of the rind are significant ($p_{distance}$ =

0.00 $p_{thickness}$ = 0.00). This is illustrated in figures Figure 3.13 and Figure 3.14 showing the slope of the force vs. the angle and the slope vs. the thickness respectively.

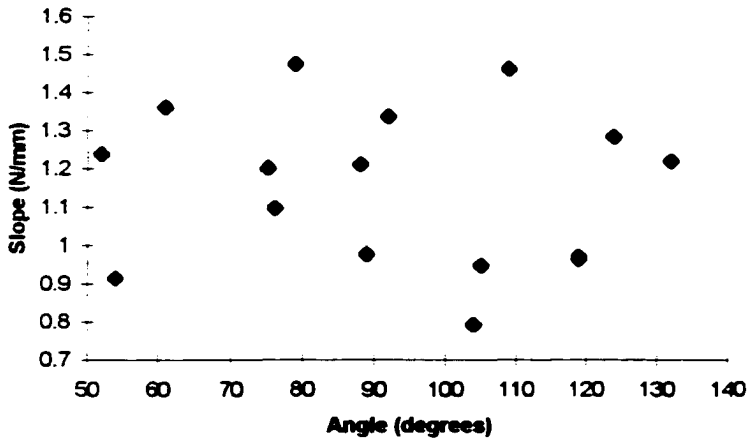

Figure 3.13: The slope of the rind puncture (puncture force / distance) vs. needle insertion angle. The scattering of the points indicates no relationship between the two variables.

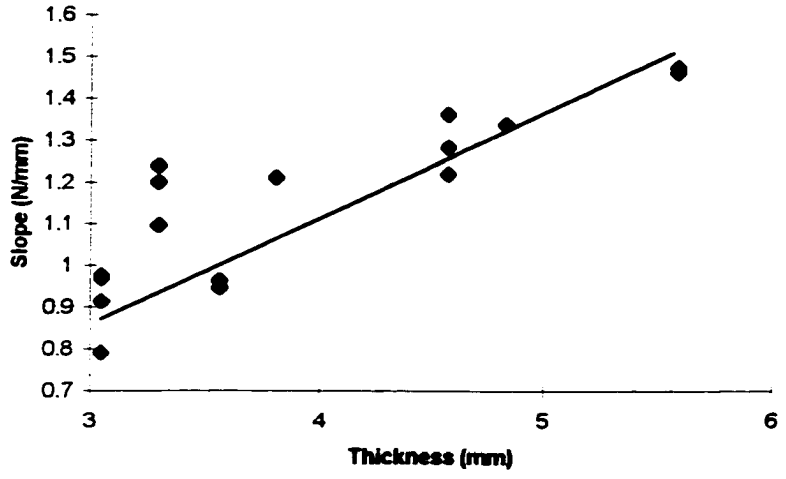

Figure 3.14: The slope of the puncture force (puncture force / distance) vs. orange rind thickness. The organization of the points is indicative of a linear relationship between the two variables.

3.4.4.3.3.4 Hypothesis 3

The data were first visually inspected to see if there was a difference in the force curves for the three different insertion speeds. The three force curves (representing the three speeds) at each insertion location were plotted out, and appeared to be similar for the three speeds. The curve for the highest speed of insertion is noticeable more noisy than the other two speeds, this is due to noise from the MTS machine moving quickly with little force resistance. Figure 3.15 illustrates the force curves for the three speeds for orange 1 at location A.

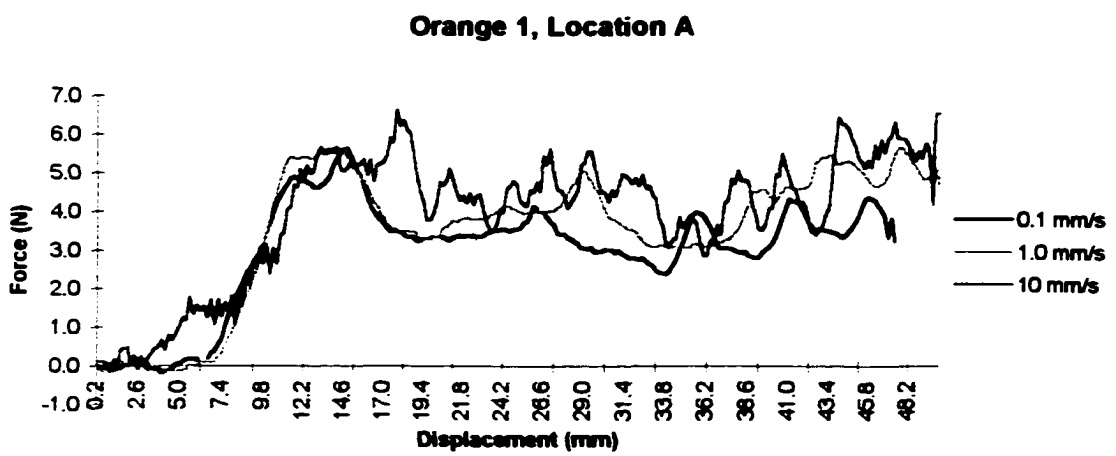

Figure 3.15: Force vs. displacement curve for Orange 1 at the three insertion speeds. Notice the similarity between the force curves. Each curve shows an increase as the needle is inserted to the rind, followed by a drop-off to the pulp puncture force after the rind puncture. Subsequent increases are due to hitting seeds, or the segmental membranes of the orange.

Next, the data was analyzed and three key features from each force curve (the peak rind puncture, the slope of the rind puncture curve, and the average pulp puncture force) were identified. The feature data were plotted as scattergrams versus the insertion speed and the Pearson's r coefficient of correlation was calculated for each feature. The scattergrams are presented in figures 3.16, 3.17, and 3.18. The scattergrams clearly show an increase in variability due the MTS machine at the highest speed.

The r-value for the puncture slope was found to be 0.054 indicating for this data there is no correlation between needle insertion speed and puncture slope. For the average pulp force the r-value was found to be 0.296, and for the peak puncture

force the r-value was found to be 0.288. This indicates a very weak correlation (they are both barely significant at $\alpha = 0.05$ but are not significant at $\alpha = 0.01$).

From these results, I believe the assumption that there is no difference in the force curves for the three different insertion speeds is a good assumption to make in developing the force models. This assumption was tested again during the canine cadaver specimen testing described in Section 3.4.4.

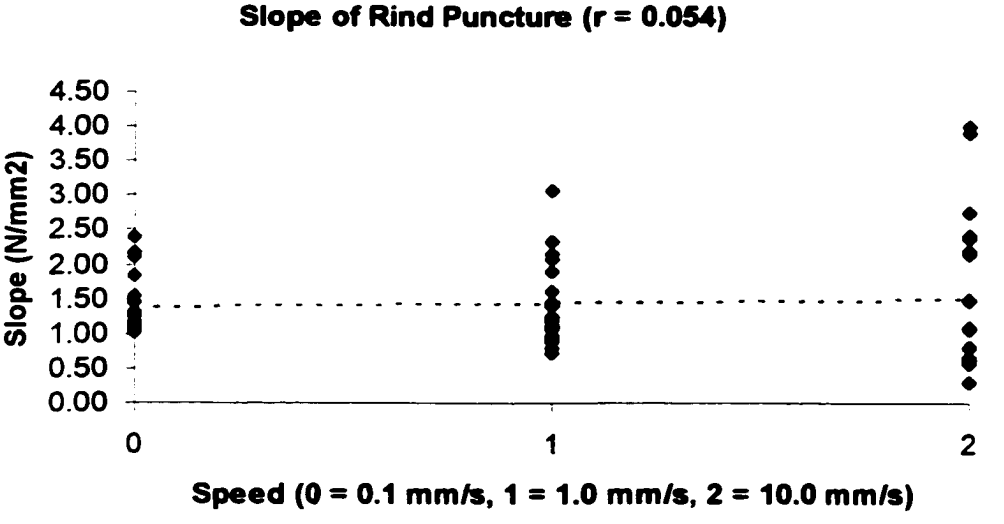

Figure 3.16: Scattergram for the slope of the rind puncture curve versus the speed. There is no correlation between the variables. The variability in the highest speed is due to limitations of the MTS system.

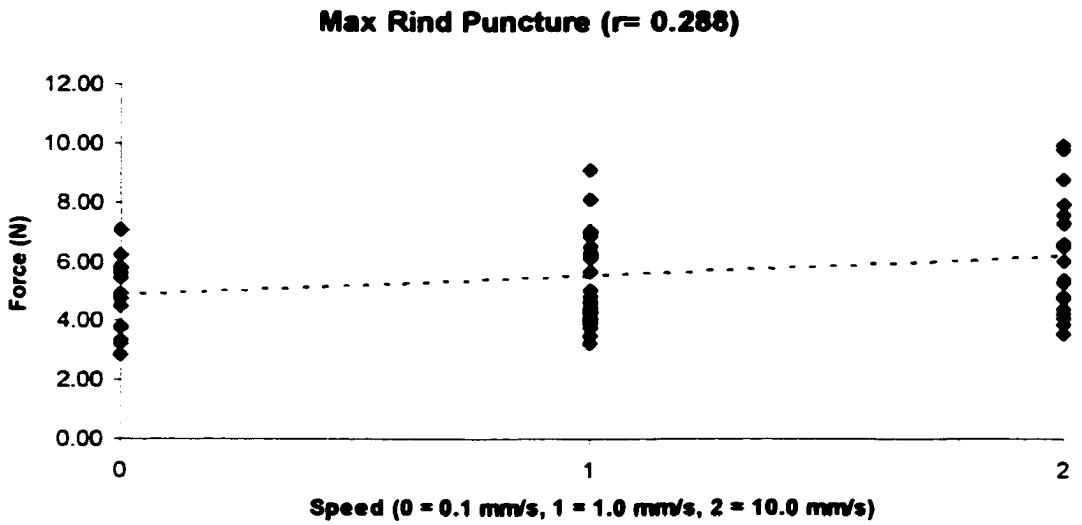

Figure 3.17: Scattergram for the maximum rind puncture force versus the speed. There is a weak correlation between the variables.

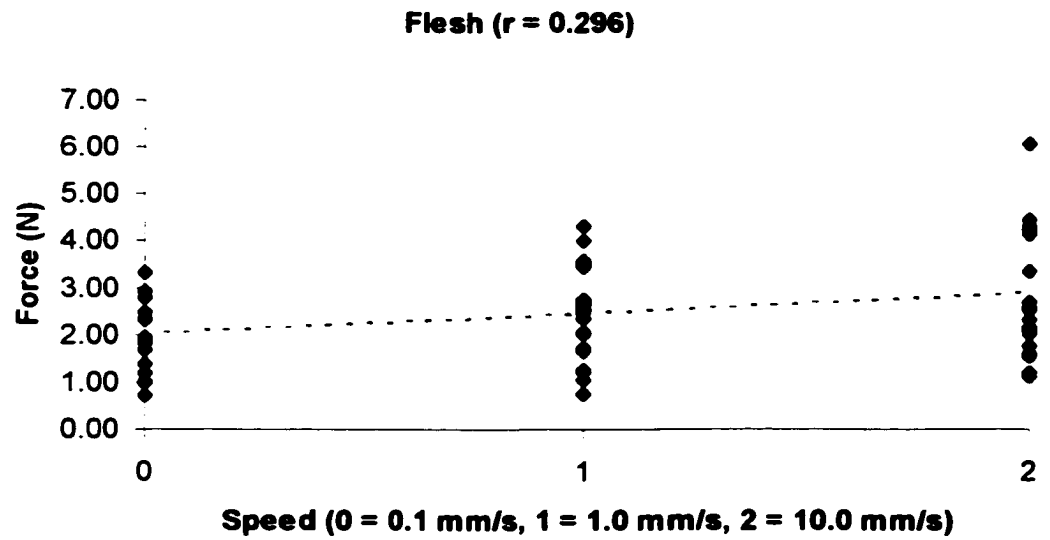

Figure 3.18: Scattergram for the average pulp puncture force versus the speed. There is a weak correlation between the variables.

3.4.4.3.4 Development of Virtual Oranges

Force models were developed for the various parts of the orange (the rind, the pulp, and the seeds). The curves were parsed into individual curves for each tissue type. The generalized rind curve is a line created from the regression equation using the distance the needle has penetrated into the orange and thickness of the rind as independent variables created in Minitab. The pulp puncture force is a constant, the average of the pulp force measurements. The generalized tissue force equations are:

*Rind: Force = 0.37*Distance -0.14(Thickness) + 0.20(Distance)(Thickness)*

Pulp: Force = 2.54 N

Seed: Force = 2.54 N + 0.3(Distance into the seed)*

*Noise: Force = Force + 0.2 * (Random number [0 to 1] - 0.1)*

Note that as this was an initial attempt at creating forces curves, the fall off from the peak rind puncture force back to the pulp force was not modeled. Instead, a line with the slope -3.5 N/mm was used for this equation. The noise term was added in because the initial force curves appeared to be too smooth as compared to the real orange force curves.

A masked MRI data set for the real oranges was created as described in section 3.4.2. The first step in creating a virtual orange was to create a tissue trajectory from the masked MRI data. This was done by shooting a ray through the masked orange and outputting a tagged MRI trajectory with a constant representing the tissue type at each voxel (ex. [0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1] where 0 is air, 2 is rind, 1 is pulp, 3 is seed). Next, a force array was

created by using the force equations to identify the puncture force at each voxel. Figure 3.19 shows force curves from two representative real oranges, and two virtual oranges.

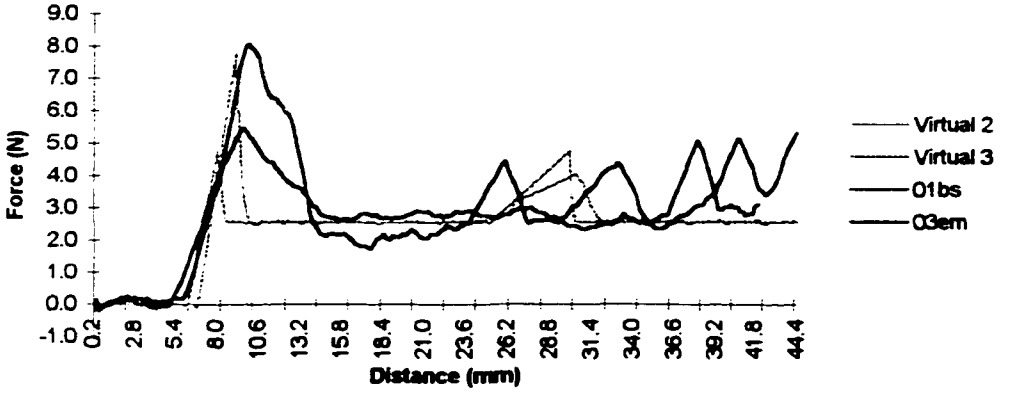

Figure 3.19: Force curves for two virtual oranges and two real oranges.

Four virtual orange force trajectories were created and incorporated into a simple simulator for user testing (described in Chapter 5). The simulator queries the user for which orange to display, and then reads the input position from an Immersion Impulse Engine and outputs the force from a force array for the virtual orange. The code for this simulator is in Appendix B.

3.4.4.3.5 Discussion

Oranges and tomatoes were used as test objects because they have a complex layered anatomy that somewhat resembles the human back (in that it is complex, and layered, and that similar modeling techniques could be used for both).

When a needle was inserted into the orange, the magnitude of the force peaks then drops off as the rind is punctured. Next, the pulp was punctured. Small juice oranges that had seeds were used, so occasionally there would be a spike in the force curve as a seed was hit. It was possible to distinguish changes in the force curves as the needle punctured the substructures of the orange. The force curves for the tomatoes turned out to be less interesting because after the initial skin puncture there were no structures to affect the force magnitude.

One difficulty caused by the setup for the MTS testing was that the needle rubbed against the hole in the acrylic box in some trials. This increased the magnitude of the measured force. It was decided that a new method of registration must be found for future tests. Another difficulty was that when the angle between the needle and the surface of the orange was too acute, the needle bent and slid down the side of the orange before puncture.

3.4.4.4 Canine back

3.4.4.4.1 Overview

The next pilot study was performed using a specimen taken from a canine back. This study was performed under Protocol 94H0415. [Appendix A] The goal of this study was to collect data from a close analogue to the human epidural region, and investigate any difficulties that could occur in the experimental design. The hypotheses under test were:

Hypothesis 1: Forces are measurable using the MTS machine.

Hypothesis 2: Forces due to change in angle of needle penetration are negligible.

Hypothesis 3: There are no significant differences in needle penetration between 0.2 mm/s and 2.0 mm/s rates of needle insertion.

3.4.4.4.2 Methods

A tissue specimen taken from the lumbar region of a canine back was obtained through the Animal Resources Laboratory at Ohio State. All biomaterials testing was performed within 24 hours of sacrifice. The specimen was shaved and mounted in a wooden box with a grid of oil soaked yarn placed over it for registration. A MRI scan was performed to visualize the internal anatomy. The following morning MTS testing was performed.

The needle was inserted under load displacement control at 2.0 mm/s and 0.2 mm/s in each of five intervertebral spaces. These speeds were chosen because they were representative of typical epidural needle insertion speeds. It was decided not to use a higher rate of insertion because the MTS system added a lot of noise at 10mm/s speed during the orange trials. A wooden stand was used to mount the specimen at different fixed angles. Data were taken at five angles of insertion (-60, -30, 0, 30, and 60 degrees), for each speed, at each location.

3.4.4.4.3 Results

By matching up the force curves with the MRI images it was apparent that the MTS machine was capable of measuring the changes in force as individual tissue layers were punctured (Hypothesis 1).

The data were first visually inspected to see if there was a difference in the force curves for the two different insertion speeds or for the different angles of

insertion. The force curves at each insertion location were plotted out, and appeared to be similar for the two speeds and across the angles.

A scattergram was plotted (shown as Figure 3.21) for the peak puncture force at the five angles of insertion. The Pearson's r was found to be –0.524, indicating there is a negative correlation between insertion speed and puncture force (at $\alpha = 0.01$). This is likely due to the differences in the tissue layers punctured as the angle changed. For the epidural modeling, I am assuming there is no difference in force due to change in needle insertion angle. These results may indicate that the force models cannot be generalized across different insertion angles. Further study is needed before such generalization is warranted.

A scattergram was plotted (shown as Figure 3.22) for the peak puncture force at the two different rates of insertion. The Pearson's r was found to be 0.126, indicating no significant correlation between speed and force (Hypothesis 3). Instead, differences in the force curves were dependent on the types of tissue that the needle punctured. A sample data set (from the 5th vertebrae tested) is shown as Figure 3.20.

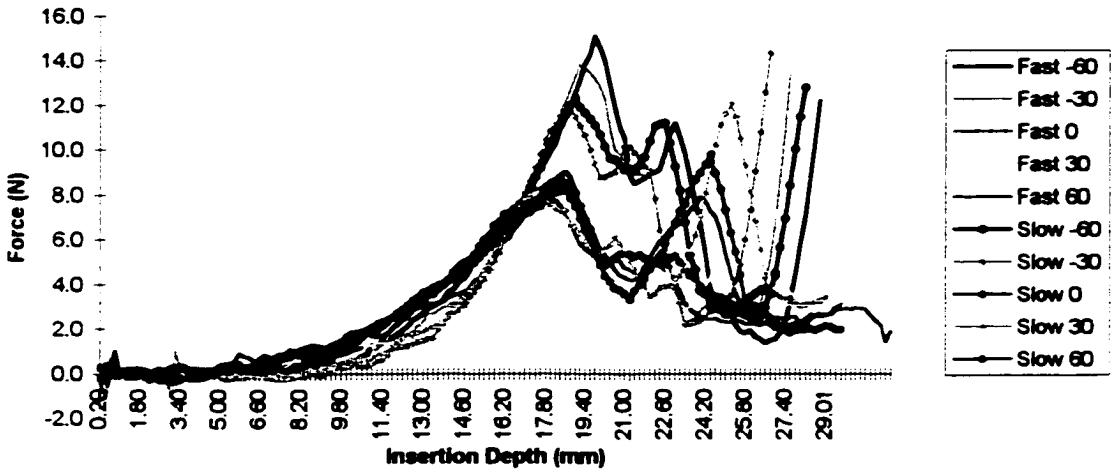

Figure 3.20: An example data set from the canine back specimen testing. The data were collected at two different insertion speeds (slow = 0.2 mm/s, fast = 2 mm/s), and 5 different angles from perpendicular for each speed (-60, -30, 0, 30, and 60).

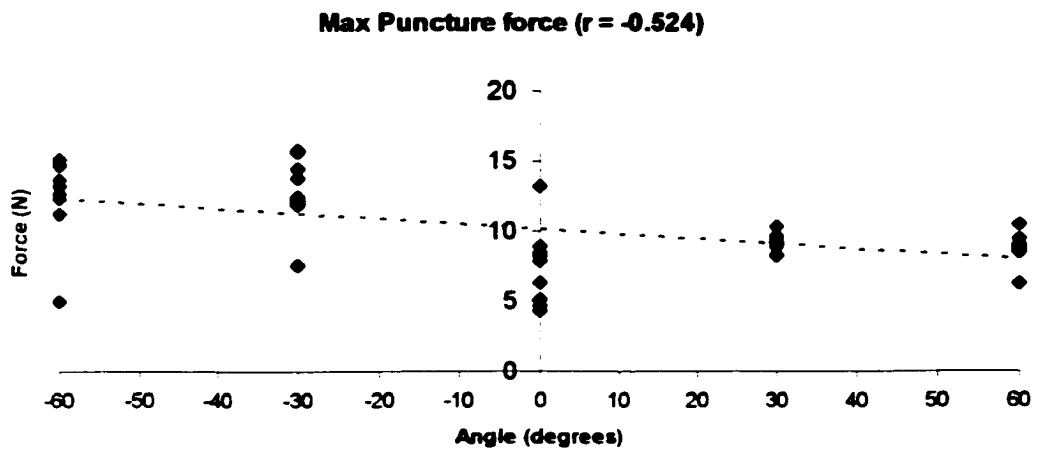

Figure 3.21: A scattergram for the canine peak puncture force for each vertebra tested at angles –60, -30, 0, 30, and 60. The Pearson's r was found to be –0.524, indicating a significant correlation between angle and force.

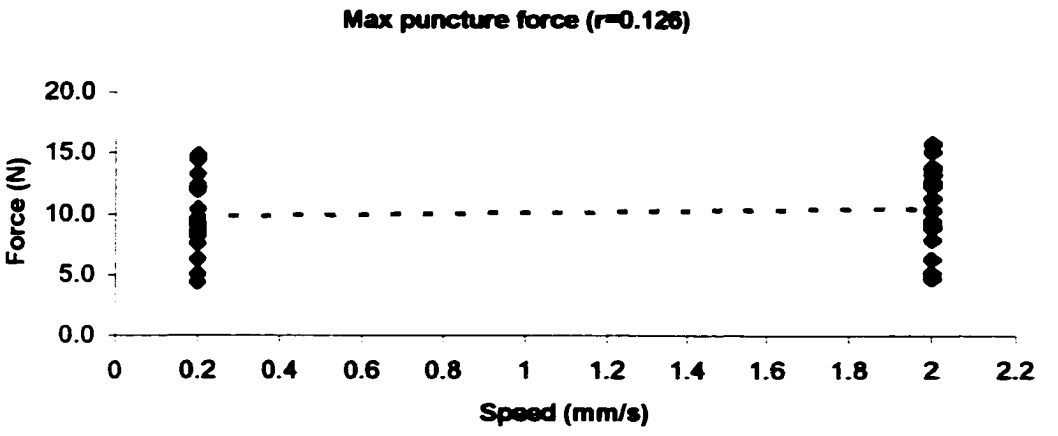

Figure 3.22: A scattergram for the canine peak puncture force for each vertebra tested at 0.2 and 2.0 mm/s. The Pearson's r was found to be 0.126, indicating no significant correlation between speed and force.

After the above tests were completed the tissues were dissected layer by layer and tests were performed inserting the needle into each current top layer. It was found that the puncture force is not dominated by the type of tissue being punctured, but instead by the fascial layers in between the tissue layers. For example, when the specimen was dissected to the facial layer encapsulating the muscle and the needle was inserted a measurable puncture curve was observed. However, when the layer of fascia was subsequently removed and the needle was inserted into a new spot no measurable puncture curve was obtained.

3.4.4.4.4 Discussion

In performing this study, one difficulty occurred because the specimen was *ex vivo*, an approximately 8cm x 12cm x 6cm section of the back containing the

lumbar spinal region. The specimen was held in a wooden box and affixed with plastic tie wraps. The skin was not held under normal tension, so as the needle compressed into the tissue, the skin slid across the fat layer and might have affected the puncture forces. Because of this, it was decided that for future animal model studies the entire cadaver would be used on the MTS machine. This would allow each tissue layer to be held in place by the natural tension that occurs *in vivo*.

Another result found in this study was that the fascial layers dominate the puncture force. Because of this, it was decided to forego further attempts to create a virtual patient directly from the MRI data (intensity, tissue T1 and T2 relaxation times, etc.) but instead to create a masked data set from the MRI data. This process is discussed in detail in section 4.3. This also implies that the force models developed in Section 4.2 are dependent on the known layering of the tissues as the needle is inserted.

3.4.5 Biomaterials testing

The previous section described several pilot studies that were performed to evaluate the feasibility of the force modeling methodology and to test the validity of the model assumptions. This section describes the three data collection studies performed to obtain the data needed to develop the force models. Data were collected on human tissue samples and on swine cadavers. At the end of this section, limitations of using the data collected for modeling the human epidural needle puncture will be discussed.

3.4.5.1 Human tissue (skin/fat)

The goal of this study was to collect needle puncture force data for human skin and human fat. This study was performed under Protocol 94H0415. [Appendix A] Human tissue samples were obtained through the Human Tissue Procurement Services at the OSU Hospitals on August 21, 1997. The samples were excised pieces from the underarm region and were comprised of a layer of skin with a thick layer of fat. The samples were mounted in round acrylic containers on top of a piece of Styrofoam. No MRI images were obtained for the samples. We were able to obtain fourteen force curves for needle insertion into the samples using the MTS machine. The needle insertion speed was 0.2 mm/s, and load was measured as a Tuohy needle was inserted under displacement control. From this data, force curves for skin and fat were developed, as explained in Section 4.2.6 along with a figure showing the data from this study.

3.4.5.2 First swine cadaver study

The goal of this study was to collect needle puncture force data for a close analogue to the human back. This study was performed under Protocol 94H0415. [Appendix A] Three young swine cadavers were obtained through the Animal Resources Laboratory at Ohio State on March 30, 1999. The lumbar region in the swine is comprised of six vertebrae. Figures 3.23 and 3.24 illustrate the skeletal anatomy of the swine.

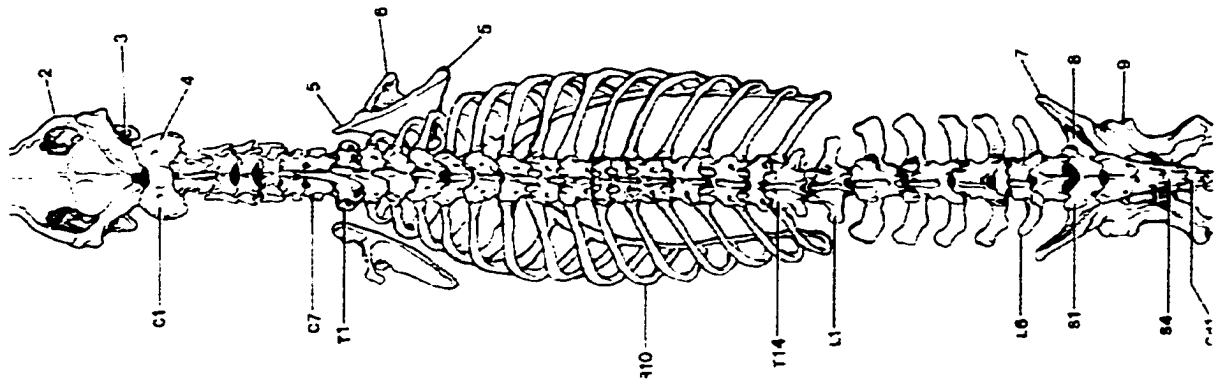

Figure 3.23: Skeletal anatomy of the swine viewed from above. Note the lumbar region, L1 through L6 [reproduced with permission from Sack, 1982].

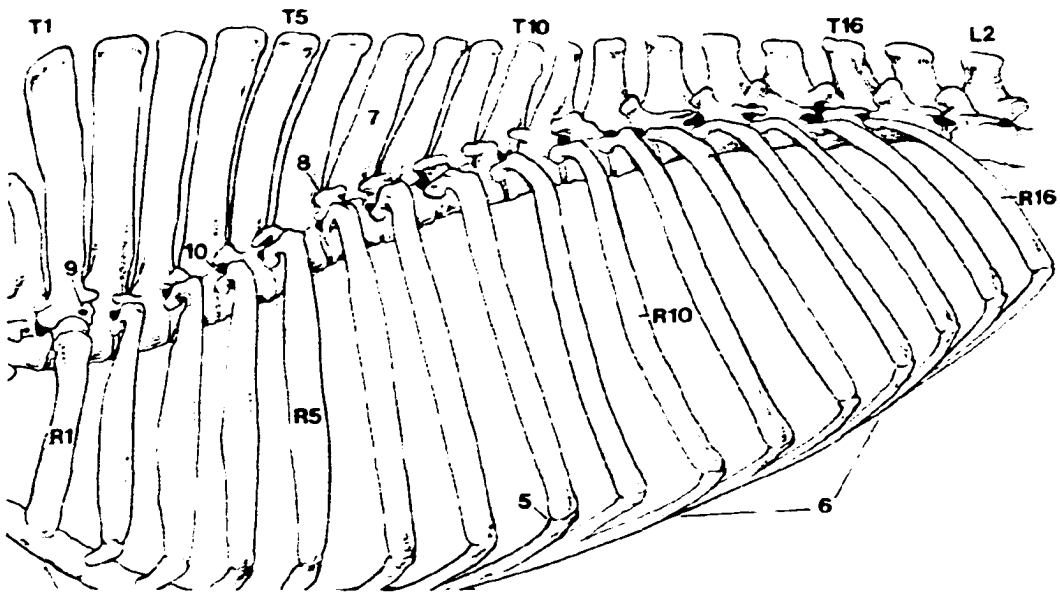

Figure 3.24: Skeletal anatomy of the swine from the sagittal view. Note L1 and L2 at the top right side of the figure [reproduced with permission from Sack, 1982].

The swine were freshly sacrificed at noon. The cadavers were then mounted on wooden boards and firmly secured with tie wraps. The backs were palpated and the lumbar spinal region was approximately identified. A centerline was drawn along the spine. Marks were made at each centimeter mark along this line with a permanent marker. Next, strips of oil soaked yarn were placed over the marks for registration of the MRI images with the locations for needle insertion during MTS testing. The MRI imaging was performed in the evening of the day the swine cadavers were obtained. Figure 3.25 shows one of the cadavers before it was placed into the MRI machine. Figure 3.26 shows a MRI image from the second swine, 7 cm up from the start of the grid marks.

Figure 3.25: Swine cadaver prepared for the MRI imaging. The cadaver is firmly affixed to a wooden board with tie-wraps. Notice the grid of yarn lying on the skin that is used for registration between the MRI imaging and the MTS testing. The rump of the swine is at the bottom of the picture.

Figure 3.26: MRI image of Pig 2, 7 cm up from the start of the grid marks. Note the white dot at the center of the bottom of the image and the bright line going across it. This is the oil-soaked yarn.

The following morning the swine cadavers were tested on the MTS machine. The Tuohy needle was inserted into the previously marked locations at a displacement rate of 0.1 mm/s and the puncture force was measured at every 0.1 mm. This rate was chosen because it was representative of the typical speed of epidural needle insertion, was slow enough to minimize noise from the MTS testing system, and was fast enough to allow for completion of the testing within 26 hours of sacrifice. Figures 3.27 and 3.28 show the swine cadaver on the MTS machine.

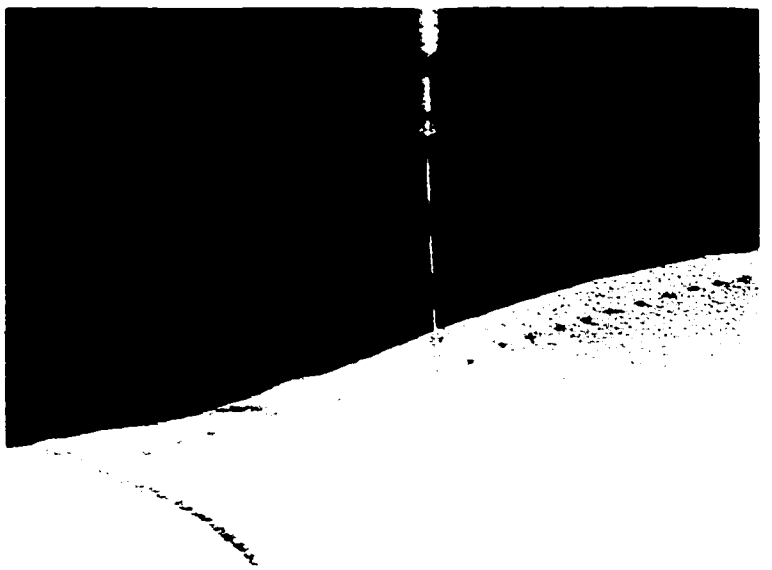

Figure 3.27: Photograph of a swine cadaver undergoing MTS testing. Note the marked spots, axial slice views were obtained showing the internal anatomy at each of these locations from the MRI scans. Also note the dimpling of the skin as this picture was taken just prior to skin puncture.

Figure 3.28: Photograph of a swine cadaver on the MTS machine. Note the needle attached to the end of the insertion rod (indicated by the arrow).

Once all the data was collected it was analyzed to create force models for muscle, interspinous ligament, and bone. The first step was to use the MRI images to identify the types of tissues the needle punctured as it was inserted into each location. Next, the load vs. displacement curves from the MTS testing were filtered using the 7-point moving average filter to remove noise from the test system, and then parsed to separate out the curves for each tissue. The generalized force curves for each tissue type are presented in section 4.2.6. Figure 3.29 below shows the force curves for swine 2.

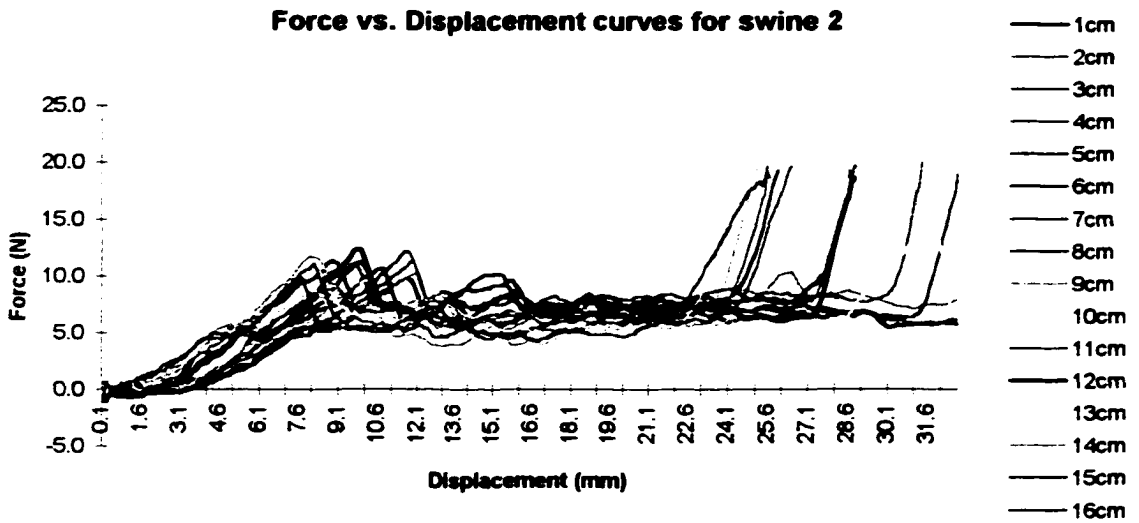

Figure 3.29: Force vs. displacement curves from the MTS testing for Swine 2. The force increases sharply at the end of each curve when the needle hits bone.

3.4.5.3 Second swine cadaver study

One limitation with the first swine study was that none of the trials resulted in placing the needle in the epidural space because of the small opening between the vertebrae in the swine. The goal of this second study was to collect needle puncture force data for the ligamentum flavum and the epidural space. This study was performed under Protocol 94H0415. [Appendix A]

A single swine cadaver was obtained through the Animal Resources Laboratory at Ohio State on June 4, 1999 and taken to the Orthopaedic Biomaterials Laboratory. No MRI imaging was performed on the subject. A dissection was performed excising the skin, fat, and muscle layers to expose the spinal column. A block was placed under the pig to widen the spaces, and the tissues between the laminae were removed to allow access of the needle from perpendicular to the body.

The Tuohy needle was inserted into the 5 exposed vertebral spaces at a displacement rate of 0.1 mm/s while the puncture force was measured at every 0.1 mm. The MTS testing was completed within 4 hours of sacrifice. The following photograph shows the specimen immediately before MTS testing.

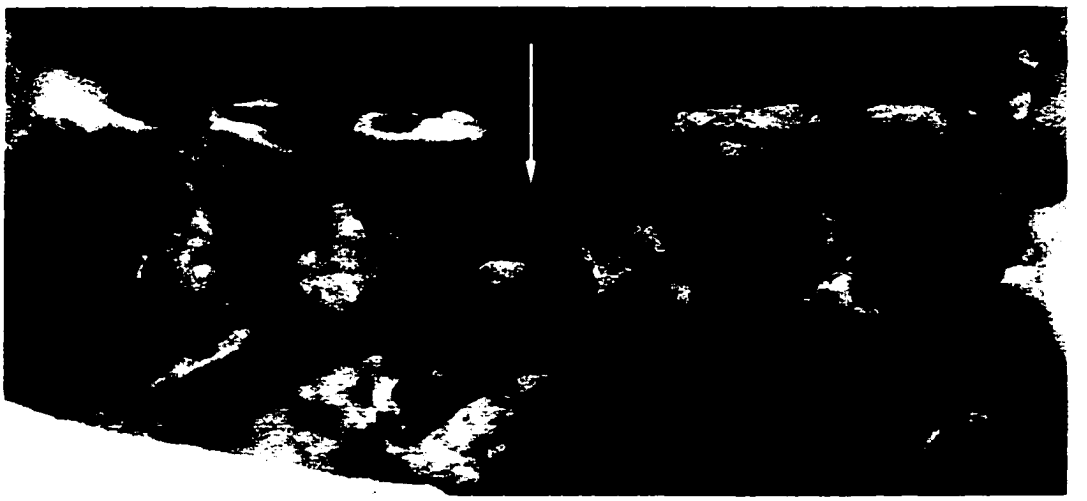

Figure 3.30: Dissected swine cadaver. Force measurements were taken of the ligamentum flavum, perpendicular to the specimen between the spinous processes. In the picture the middle spinous process has been removed. The arrow indicates where the needle will be inserted.

The collected data was analyzed to create force models for the ligamentum flavum and the epidural space. As with the load vs. displacement curves from the other studies, the curves were filtered using a 7-point moving average filter to remove noise introduced into the data by the MTS system. The curves were then parsed to separate out the curves for each type of tissue. The resultant force curves are presented in section 4.2.6.

3.4.5.4 Limitations of using swine cadaveric specimens

It is a common practice to use swine models for research as analogues to humans. [Caputy, 1995] However, dead pigs are not equivalent to live humans so there are limitations to the resultant force models due to this. One such limitation is

that the needle puncture force for human tissue is not equivalent to that for swine tissue. As an example, the average puncture force for human skin was found to be 6.0 ± 0.7 N in the human tissue trials described in section 3.4.5.1, whereas for the swine trials in section 3.4.5.2 the average puncture force was 12.9 ± 2.6 N. This is because swine skin is tougher than human skin.

Although all biomaterials testing were performed within 26 hours of sacrifice on unembalmed animal cadavers, there are physical changes that occur with death. Two such changes are rigor mortis and the pooling of liquids at the base of the cadaver due to gravity. No attempt was made to quantify what effects these changes may have on the puncture force.

CHAPTER 4

FORCE-FEEDBACK MODELING II: EPIDURAL FORCE MODEL & SIMULATOR DEVELOPMENT

4.1 Introduction

Chapter 3 described the data collection studies in which force vs. displacement curves were collected for human tissue samples and swine cadavers. In this chapter, the process to turn these data into generalized force curves for the individual tissue types will be described, and the force models will be presented.

4.2 General force-feedback model for the epidural needle insertion procedure

The first step in developing the force-feedback model curve was to remove noise introduced by the MTS machine from each force curve. A program was written in Matlab to read in the MTS data, filter it, invert it (in the original data the forces and displacements are negative), and write it out to a text file. As described in Section 3.4.4.3.3.2, the chosen filter was a 7-point moving average filter.

For the data from the first swine study the next step was to use the MRI images to measure the width of each tissue punctured during each needle insertion trial. This step was not necessary for the human tissue samples, because it was easy to measure the tissues directly; or for the second swine study, since the cadaver was dissected down to the layer of interest. The width of each individual

tissue layer was correlated with the force curves to break them into curves for each individual tissue. Parsing the curves was performed by hand using an Excel [Microsoft Corporation, Santa Rosa, CA] spreadsheet.

The next step in the process was to perform a curve fit to the data for each tissue type, resulting in the generalized force models for the specific tissue. The curve fitting process is described in the discussion for each tissue type.

The force modeling relies on the known layering of the tissue types. Because of this, forces such as the friction force of previously punctured layers are inherently incorporated as I have measured the force needed to puncture the tissue at a given depth. This factor will also be discussed as it affects the different force models for each tissue.

The curve fitting process results in a smooth curve that bests fits the data. In simulator trials, user's comments during the orange study (section 4.5) implied that these curves are too smooth. Therefore, a noise term is introduced to add low frequency noise to the resultant force model. This noise term is added the modeled force at any location, x, and is:

Noise term: $Noise = 0.2 * (Random\ number\ [0\ to\ 1] - 0.1)$

A possible future area of research would be to study this noise in order to identify how much of the noise in the measured data was from the MTS machine. Another avenue of study would be to identify the best way to reintroduce noise into the force models. A third area of study would be to measure and accommodate noise introduced by the force display device.

The force curve equations for each tissue type are presented in detail next. All force units are Newton's.

4.2.1 Skin

The force-feedback model for the skin was created from data collected from human tissue specimens described in section 3.4.6.1. In Excel, the curves from each trial were aligned at the point of the start of puncture. Then each curve was broken into three pieces: pre-puncture (increase in force to the point of puncture), post-puncture (decrease in force until the steady-state is reached), and fat (the steady-state until the next layer is reached). Next, the partial curves were averaged on a point-by-point basis. This average curve was read into Matlab and curve fitting was performed to find the coefficients of a polynomial p(x) of degree n that fits the data, $p(x(i)) \approx y(i)$, in a least squares sense. N was varied from 1 to 4 and the output curves were plotted against the input curve to judge which order polynomial produced the best fit.

The skin puncture models were calculated to be:

Pre-puncture: $Force(X) = 0.0235 + 0.0116 * (X - X_s) - 0.0046 * (X - X_s)^2 + 0.0025 * (X - X_s)^3$; *for* $Force(X) <= Puncture\text{-}Force$, $X < X_p$

Puncture-force: $Force(X) = 6.0372\ N$; $X_p = X$

Post-puncture: $Force(X) = 6.0372 + 0.4516 * (X - X_p) - 0.5287 * (X - X_p)^2$ *for* $X > X_p$ *and* $Force(X) >= 1.974$.

X = *Needle penetration Depth (mm)*

X_s = *Skin surface depth (location of the surface of the skin relative to the start of the needle movement) (mm)*

X_p = *Skin puncture depth (mm)*

The graph in figure 4.1 shows all of the load-displacement curves for the human tissue specimen trials along with an example force curve from the force model for skin puncture.

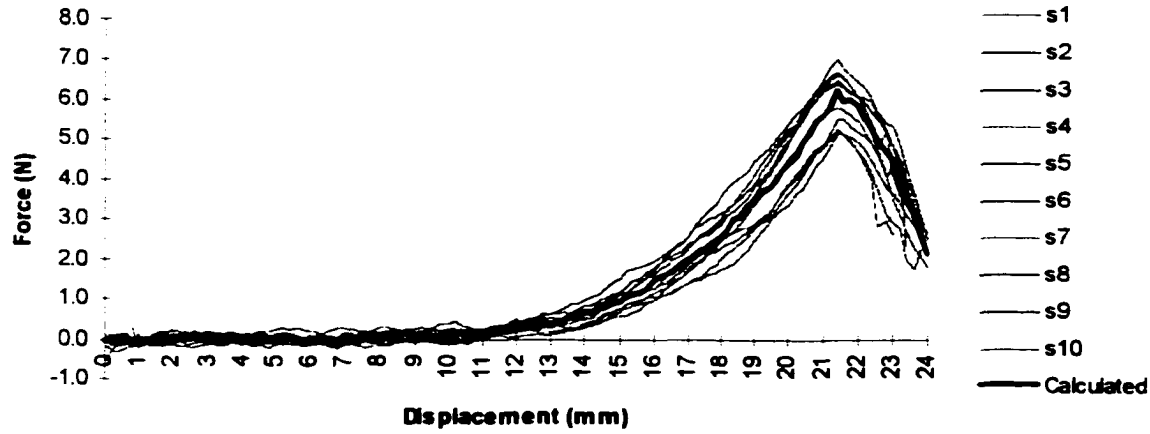

Figure 4.1: Force/displacement curve for skin puncture. S1 to s10 are the data curves collected from human tissue samples. "Calculated" is the force curve calculated from the equations.

4.2.2 Fat

The force-feedback model for the subcutaneous fat was also created from data collected from human tissue specimens described in section 3.4.6.1. These

curves were broken into three pieces, and the fat layer corresponds to the steady-state level reached before the trial ended or the Styrofoam layer was reached. The fat constant was calculated to be:

Fat: *Force = 1.974 N*

Because of the nature of fat, it is a loosely connected gelatin-like substance, it seems that this force may be too high. I believe that the force needed to puncture fat is fairly low, and is dominated by the frictional force from the skin layer and the hardness of the tissue below the fat (in this case, Styrofoam). I believe that a more accurate puncture force for fat could be ascertained from in-vivo human measurements that were not possible in this series of studies, or from data using samples with thicker layers of fat.

4.2.3 Muscle

The force-feedback model for the muscle was created from data collected in the first swine study, described in section 3.4.5.2. The muscle curve was a distinct curve found in most of the data sets from the MTS trials, and appears as a second puncture curve after the steady state is reached from the skin puncture. In Excel, the muscle curve was parsed out of the data from each trial and broken into three pieces: pre-puncture (increase in force to the point of puncture), post-puncture (decrease in force until the steady-state is reached), and steady state.

The start and end of these trial curves were not as apparent as for the skin curves; in some cases the pre-puncture curve started before the fat steady-state was reached, in others the post-puncture curve did not reach a steady-state before the next tissue layer was reached. The best curve-fitting strategy was to read each

curve into Matlab and perform polynomial curve fitting (again varying N was from 1 to 4 to find the order of polynomial which produced the best fit). The polynomial constants were stored, and then averaged across all of the trials.

The models were created using data from swine, but I am trying to develop models which approximate puncture of human tissue. The force model for puncturing human fat was 1.974 N; however, for the pig fat the average puncture force was 6.027 N. I believe that this discrepancy is caused by the increased frictional force from the tougher and thicker pig skin. Therefore, a pig correction (PC) constant of 4.053 N will be used to adapt from the calculated pig muscle puncture models to models for human muscle.

The muscle puncture models were calculated to be:

Pre-puncture: $Force(X) = (6.0266 + 0.8287 * (X - X_m) + 0.1078 * (X - X_m)^2) - PC$

for $Force(X) <= Puncture\text{-}Force$, $X < X_p$

Puncture-force: $Force(X) = 8.407 - PC$; $X_p = X$

Post-puncture: $Force(X) = (8.407 - 2.2543 * (X - X_p) + 0.2902 * (X - X_p)^2) - PC$

for $X > X_p$ *and* $Force(X) >= 7.728 - PC$.

Steady state: $Force(X) = 7.728 - PC$.

X = *Needle penetration Depth (mm)*

X_m = *Muscle surface depth (mm)*

X_p = *Muscle puncture depth (mm)*

$PC = 4.053$ *N.*

4.2.4 Interspinous Ligament (ISL)

The force-feedback model for the ISL was created from data collected in the first swine study, described in section 3.4.5.2, in a similar fashion to the muscle model. The ISL was not punctured in every data trial, but with the MRI data set as a reference, several trials were found in which ISL was punctured. In Excel, the ISL curve was parsed out of the data from these trials and broken into two pieces: pre-puncture (increase in force to the point of puncture) and post-puncture (a steady-state value). There was not a distinct drop in force after puncture because of the thickness and density of the tissue. The curve-fitting strategy was the same as used for muscle, and again the models incorporated the pig correction factor.

The ISL puncture models were calculated to be:

Pre-puncture: $Force(X) = (8.4592 + 0.9598 * (X - X_{ISL})) - PC$

for $Force(X) <= Puncture\text{-}force$, $X < X_p$

Puncture-force: $Force(X) = 11.52 - PC$; $X_p = X$

Steady state: $Force(X) = 11.52 - PC$.

X = *Needle penetration Depth (mm)*

X_{isl} = *ISL surface depth (mm)*

X_p = *ISL puncture depth (mm)*

$PC = 4.053\ N$

4.2.5 Ligamentum Flavum (LF)

The force-feedback model for the LF was created from data collected from the second swine study. In this study the cadaver was dissected down to the LF. Five trials were collected, but only three of these trials were used to calculate the

force models. Trial 1 was discarded because the needle went through cartilage, not LF. Trial 3 was discarded because the forces were significantly lower than for the other 3 trials, indicative that again LF was not punctured. The force curves were broken into three pieces: pre-puncture (increase in force to the point of puncture), post-puncture (decrease in force until the steady-state is reached), and steady state.

The collected force curves represent puncture directly into the LF, and do not contain the drag force caused by friction on the needle from previous layers. A drag constant (D) of 6.0 N is added to correct for this effect. This constant is based on the qualitative knowledge that the LF is significantly tougher than the skin, and the measured puncture force for the two was roughly equivalent (skin = 6.0372, LF = 6.1330). Further biomaterials testing, such as in-vivo needle puncture measurements during the procedure, is necessary to create accurate LF models.

These LF models were calculated to be:

Pre-puncture: $Force = (-0.0085 + 1.5029 (X - X_{LF}) - 0.0583 (X - X_{LF})^2) + D$

for $Force(X) <= Puncture\text{-}Force$, $X < X_p$

Puncture-force: $Force(X) = 6.1330 + D$; $X_p = X$

Post-puncture: $Force = (6.1330 - 3.4693 (X - X_p) - 0.7177 (X - X_p)^2) + D$

for $X > X_p$

X = *Needle penetration Depth (mm)*

X_{LF} *LF surface location (mm)*

X_p = *LF puncture depth (mm)*

D = *6.0 N.*

Figure 4.2 show the force/displacement curves for LF puncture, both measured from the swine cadaver and calculated from the force model.

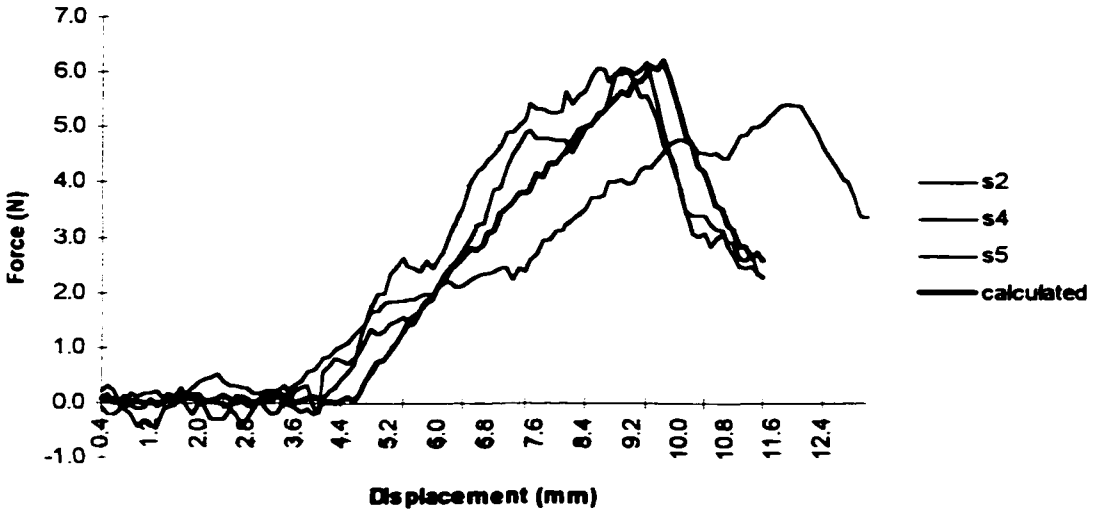

Figure 4.2: Force/displacement curve for LF puncture. S2, S4 and S5 are the data curves collected from the swine cadaver. "Calculated" is the force curve calculated from the equations. Note that I have not included the drag force in the graph.

4.3.6 Epidural Space and Subdural Tissue

The force-feedback model for the epidural space and subdural tissue was created from data collected from the second swine study. These curves were broken into three pieces, and the layer of interest corresponds to the steady-state level reached before the trial ended or bone was encountered. I grouped these two tissues together because I was unable to see any change in the force curve when the dura was punctured. This indicated the dural puncture was too small of a force change to be picked up by the machine. Further biomaterials testing, such as in-vivo

needle puncture measurements during the procedure, is necessary to create more accurate models for these tissue layers.

The epidural space and subdural tissue constant was calculated to be:

Epidural Space and Subdural Tissue: $Force = 2.437\ N$

4.2.7 Bone

The force-feedback model for the bone was created from data collected in the first swine study, described in section 3.4.5.2. The bone curve was a distinct curve found in most of the data sets from the MTS trials, and appears as a large increase in force followed by halting the data collection trial since the MTS machine was set to stop if the puncture force exceeded 20 N. In Excel, the bone curve was parsed out of the data from each trial as one piece, bone-strike (increase in force to the maximum force).

The curve-fitting strategy was the same as used for muscle, and again the models incorporated the pig correction factor. The bone models were calculated to be:

Bone-strike: $Force(X) = (8.0265 + 2.210 * (X - X_b) + 1.4814 * (X - X_b)^2) - PC$

$for,\ X < X_{max}$

X = *Needle penetration Depth (mm)*

X_b = *Bone surface depth (mm)*

$PC = 4.053\ N.$

4.2.8 Example force trajectory for the epidural needle insertion procedure

The graph in Figure 4.3 shows an example force trajectory created from the force-feedback models that have just been described. The trajectory is

perpendicular to the back at L2 for the average female virtual patient described in the next section. The graph illustrates the skin curve (13.5mm to 27mm), a small amount of fat (27mm to 27.1mm) note that this does not indicate the fat layer is only 0.1 mm thick because the needle is puncturing fat on the downslope of the skin curve. A slight muscle puncture curve is shown (27.1 mm to 29 mm) but the ISL is hit before the muscle reaches its steady state value (29 mm to 47mm). Next there is the LF puncture (47 mm to 54.4 mm) and a drop to the epidural space and dura puncture before hitting bone at 63mm.

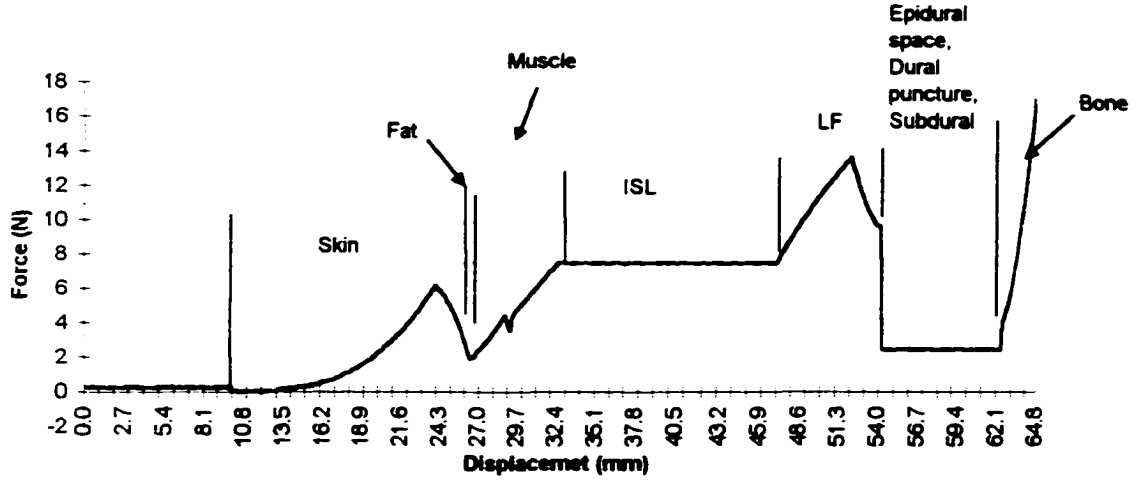

Figure 4.3: Force curve developed from the force models for the virtual female patient. Each layer of the model is identified.

4.3 Development of two virtual patients

Two virtual patients have been developed. The first is based upon an average female subject, the second on a lean male. The average female was 28 years old, 64 inches, 135 pounds, body mass index (BMI) of 23.2. Her MRI scan was taken in June 1995. The lean male was a professional athlete; 31 years old, 72.5 inches, 156 pounds, with a BMI of 19.8. His scan was done on November 20, 1999. The MRI scans were performed under Protocol 91H0301 [Appendix A].

Both subjects underwent MRI scans lying on their back. High-resolution scans were taken using slices in the coronal plane. The scans were transferred from the MRI workstation to my account at the OSC. Preprocessing included stripping off the MRI headers, transforming the scans to SGI BMP [Silicon Graphics, Inc., Mountain View, California] format, and rotating the image to obtain a set of axial slices. The lumbar region was identified in the coronal view, and the axial slices corresponding to this region were selected for segmentation. In the male subject 250 slices were selected. In the female subject 200 slices were selected. An example of the coronal view for each subject is shown in Figure 4.4.

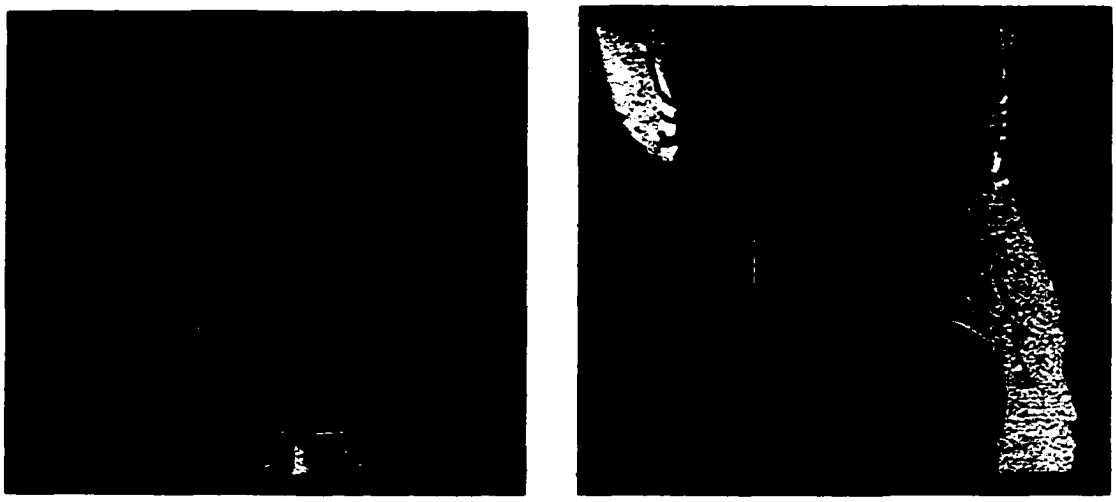

Figure 4.4: Coronal MRI views of the two subjects which will become the two virtual patients. The area outlined by the white box is the region of interest for each subject. Male subject (left). Female subject (right).

The next step in the processing was to segment out the layers of interest in each patient, thus creating the virtual patient model. The next four figures show representative axial slices and their resultant masks for the virtual male and the virtual female patient.

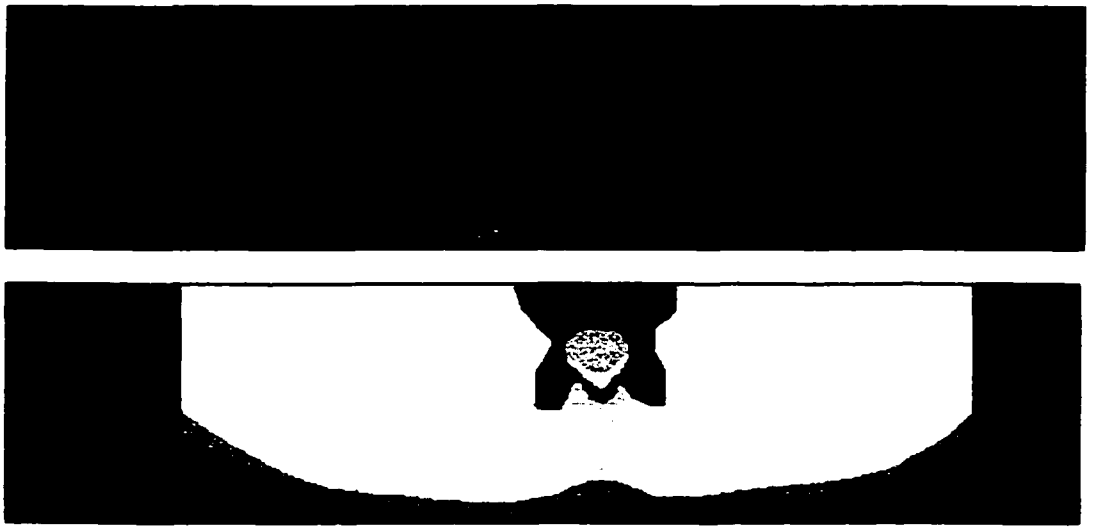

Figure 4.5: Axial MRI slice (top) and mask (bottom) from the male subject. This slice was number 368 showing the L1 disc body. Though it is difficult to see in black and white, there are 9 layers segmented out in the mask. They are (from bottom to top) the skin, the fat, subcutaneous muscle, interspinous ligament, fatty deposits, ligamentum flavum, epidural space, spinal cord, and spinous process/ vertebral body.

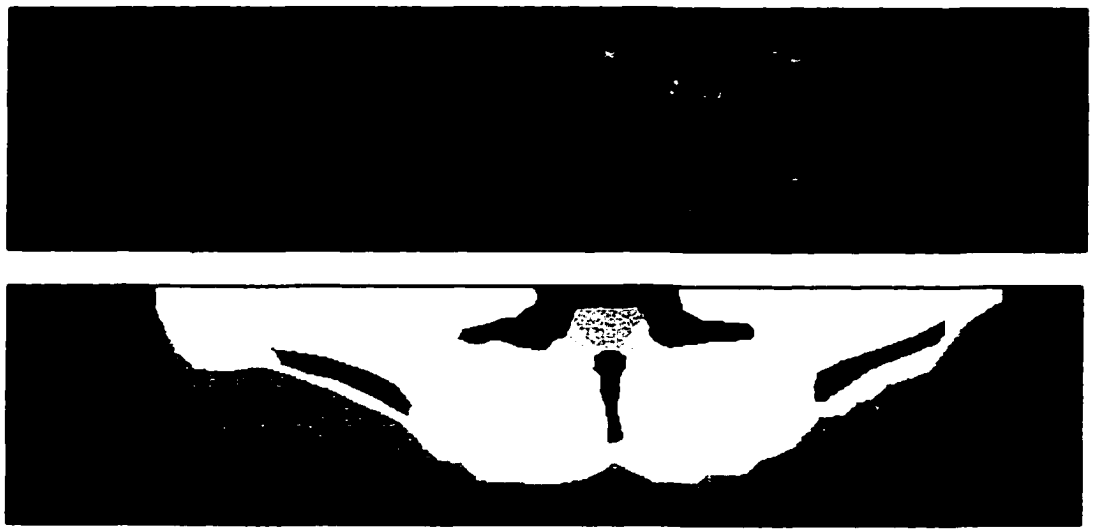

Figure 4.6: Axial MRI slice (top) and mask (bottom) from the male subject. This slice was number 289 showing the L4 spinous process.

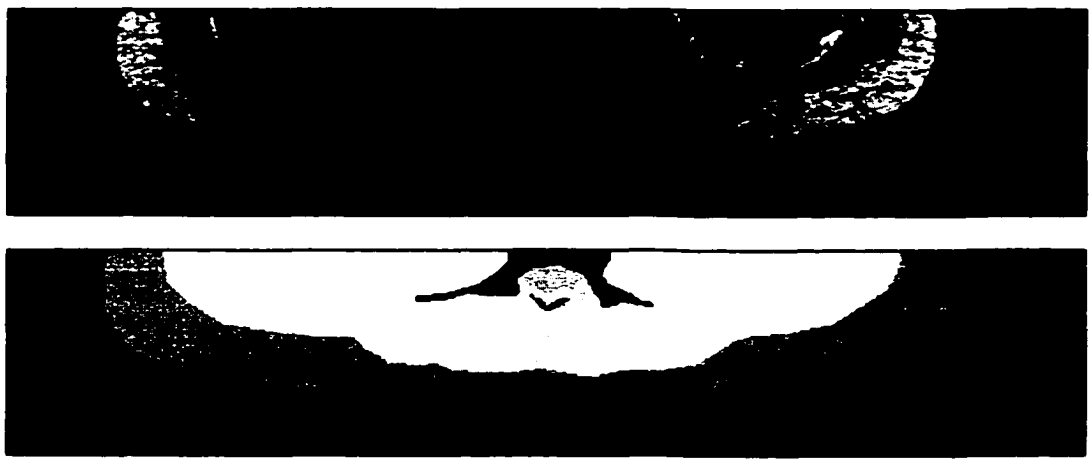

Figure 4.7: Axial MRI slice (top) and mask (bottom) from the female subject. This slice was number 268 showing the L2 disc body.

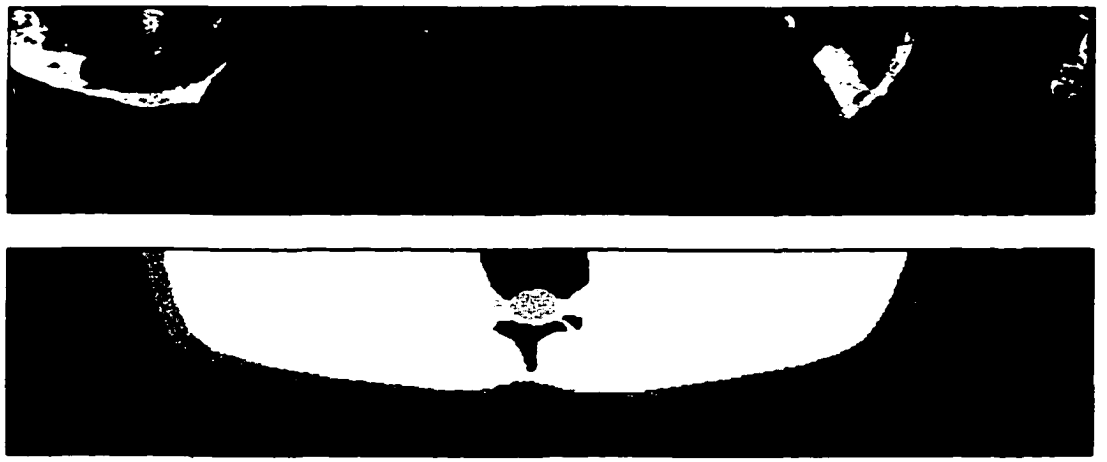

Figure 4.8: Axial MRI slice (top) and mask (bottom) from the female subject. This slice was number 388 showing the L1 spinous process. Note that while the arms are shown in the MRI scan they have been masked out, as they are not desired in the virtual patient.

A program named editmask, developed by Naeem Shareef [Shareef, 1996] was used to create the masks. Using this program, polygonal outlines of the regions to mask were drawn and then colored in with the given mask identification color. The program also had a pixel touch-up function, which was used to clean up the masks after each region was drawn.

From the four examples above it is obvious that the masks are dramatic simplifications of the anatomy. In fact, there are only ten layers of masks used in the segmentation of the virtual patients. These layers represent the air, skin, the subcutaneous fat, the bone (the spinous processes, and the iliac crests), the spinal column, the epidural space, the LF, the ISL, the fatty tissues around the spinal cord, and miscellaneous subcutaneous tissue (the large white region containing all

regions not of interest and muscle). The vertebral disc body was masked the same color as the bone.

This simplification has three causes. The first is that segmentation is an extremely tedious and time-consuming process. Therefore, only the regions that were of most interest for the lumbar epidural needle insertion procedure were segmented out, in particular those regions where the tissue force-feedback equations might differ. There are large areas of the MRI data set that are unlikely to be hit during the epidural needle insertion procedure; therefore, it is not necessary to segment these areas in detail. The second cause of the simplification is that in the MRI scans, though they are high resolution, it is very difficult to identify much of the anatomy. The third cause is the crudeness of the segmentation program; in particular, segmentation could only be performed in one plane. This made it difficult to follow a structure from one slice to another. A better segmentation algorithm would allow for multiple imaging modalities (MRI for the wet tissues, CT for the bony regions) and allow for segmentation simultaneously in multiple planes.

The end result of the segmentation process is that two virtual patients have been created. The masked MRI data sets will be used as input to the force-feedback algorithm, which assign a force to a given voxel based on the tissue type and depth. Despite the simplifications in the segmentation, the virtual patients should feel realistic since each major tissue layer on the path to the epidural space has been identified.

4.4 Assembling the simulator

Previous sections have described the creation of the force models for each tissue type in the epidural needle insertion procedure (section 4.2) and the development of two virtual patients (section 4.3). This section discusses the combining of all this input data along with a force-feedback device to create the virtual reality simulator.

The device used was a prototype needle insertion device from Immersion Corporation. This device allowed for needle positioning and force-feedback in 3-DOF. However, it was not well counterbalanced so free space did not feel free and the force-feedback in the x and y-axes was poorly defined. In the simulation, a wooden block was affixed to the device to support the needle shaft (z-axis) in a horizontal plane. The user was able to set the angle of insertion along this plane. Force-feedback was only applied along the needle shaft axis. The simulator did not constrain the user from changing the angle of insertion after initial puncture, however friction from the wooden block made it extremely difficult to change the angle of insertion while inserting the needle. The force-feedback device interfaced to my personal computer [Presario 4504, Compaq, Houston, TX] via a PCI card.

The linear resolution of the device was 0.02 mm/count. The resolution of the force was 0.003614 N/count. The maximum force the device could output was 7.4 N, which was well below the maximum force for some of the force models (ISL, LF, bone) so these models were maximized at a threshold of 7.4N.

The design of the device is similar to the Impulse Engine, with motors acting as pulleys to drive the needle shaft. The user holds the needle in an identical grip

using both devices. Because of the similar design, it was assumed that the minimal perceivable force for this device was similar to that for the Impulse Engine. This force was measured in the study described in section 3.4.1 and is 0.25N. The picture in Figure 4.9 shows the force-feedback device.

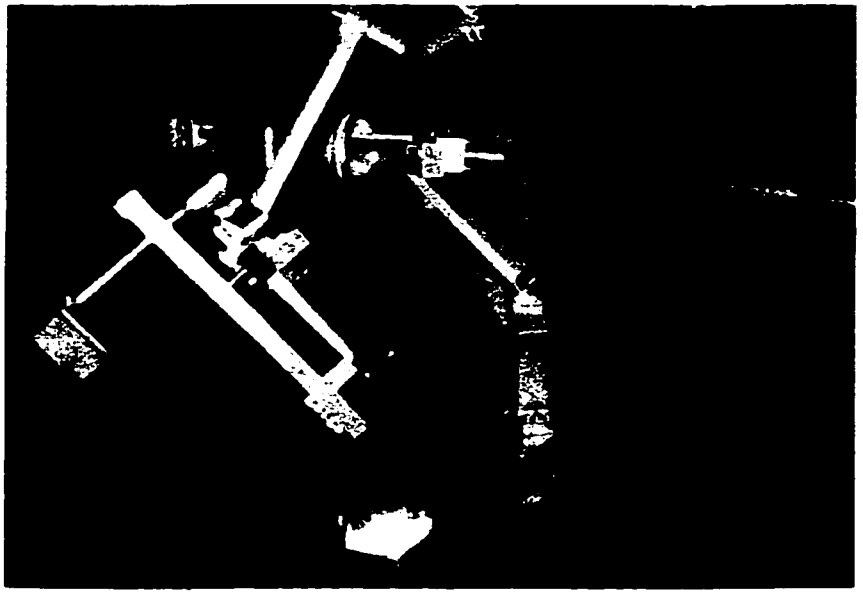

Figure 4.9: Picture of the force-feedback device used for the simulation. Notice the needle attached to the sliding shaft parallel to the ground.

To get the masks for the virtual patient, they were first translated to grayscale, and then using *xv* [SGI] they were saved in *PPM (portable pixmax) ASCII* image format. Next, the masks were transferred to my personal computer. A utility program was written to read in the PPM format and output the masks formatted for

the simulator (as *ASCII* arrays, with values 0 through 10 representing the various tissue types).

The data masks were read in and stored as input to the simulation. The first step for the user was to calibrate the device by holding the needle fully extended and pressing the "Calibrate" button. The user selected which virtual patient to use via a pull down menu. Next, the user set the angle of the needle to the desired insertion trajectory and pressed the "Set Trajectory" button. This caused the program to enter it's pre-processing loop where it: 1) Calculated the angle of insertion; 2) Identified the tissue vector from the data mask such that the tissue type for each voxel along the insertion axis was identified; 3) Created a force trajectory with a calculated force at 0.1mm intervals along the insertion axis. Next the user checked the "Enable forces" box which enabled the force-feedback.

The dialog box is shown in Figure 4.10.

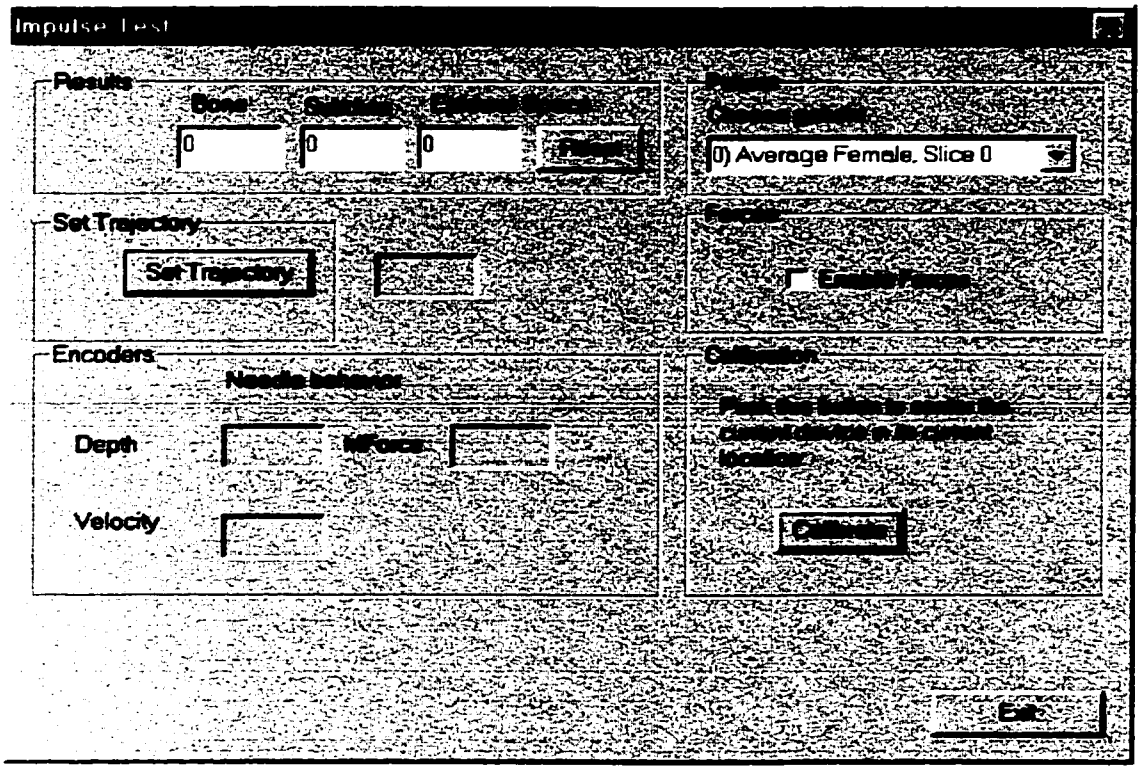

Figure 4.10: Dialog box for the epidural needle insertion simulator. This dialog box is displayed on the PC as the simulation is running.

The encoder count from the needle insertion axis was read and translated into needle tip position in mm. The average velocity was calculated over the past 100 clock cycles. If the average velocity was greater than a small (100 encoder counts / timer beats) forward velocity, the needle tip position was used to index into the force array to identify the desired force to output for the position in Newtons. The force was translated to motor command values (motor command = force in N / device resolution). A motor command was sent to the device to output the

appropriate force along the needle insertion axis. If the average velocity of needle insertion indicated the user was pulling out the needle or holding the needle still, the force output was set to a small force so that no feedback was felt on the needle.

The force-feedback device was extremely springy, very active even with no force output. Therefore, I experimented with adding a damper to the output force in an attempt to smooth out the jitters felt when a sharp change in force occurred (more details on this in section 5.6). The damping force equation was:

$Force[i] = ForceArray[i] + DC * V_{ave}$

Where i = needle insertion position

*DC = Damping constant = 0.2 (N * timer pulse /encoder counts)*

V_{ave} *= Average velocity (encoder counts / timer pulse)*

Various values for the damping constant were experimented with, and the value was finally set to 0.2. No mathematical stability analysis was performed on the system to design the damping algorithm.

The C++ program for the simulation is attached as Appendix A of this document. Table 4.1, on the following two pages, shows pseudocode for the main functions of the simulation.

User Action	Code Executed
- Start program.	
- Fully extend needle. - Press Calibrate button.	OnSetCenter() : - Sets zero position of device.
- Choose Virtual patient from pulldown list box.	OnPatientCombo() : - Reads in patient MRI data set.
- Position needle angle. - Press Set Trajectory button.	OnSetTrajectory() : - Reads the angle from the device. - Calculates the x and z coordinates of the needle trajectory with respect to the MRI data set in 0.1 mm increments. (0,0 lies at the first pixel of the data set that is reached). - Line up the tissue vector with the physical back model by padding the coordinates array with air until the skin surface is 1cm away from the start of the array. The fully extended needle is 1cm away from the skin of the physical model. - Step through the tissue vector in 0.1 mm increments. Identify the tissue type at each step. Calculate the force at each step using the equations derived in Section 3.6 based on the tissue type and the depth of penetration into that tissue and store this in the tissue force vector.

(continued)

Table 4.1: Pseudocode table illustrating the major functions of the simulator code. "Device" refers to the force-feedback device.

Table 4.1 (continued)

- Press Enable Forces button.	OnSpringEnable() : - Starts the haptic thread that gets the needle position from the device and sends calls OutputSpringForce() as quickly as possible.
	OutputSpringForce(): - Calculates the output force, translates this force into motor command values, and sends these values to the device. - If the average velocity is forward then the force is calculated by: ForceOut[i] = ForceVector[i] + 0.2 * V_{ave} If ForceOut[i] > 7.4N then set the force to 7.4N. - If the average velocity is 0, or pulling out, then: ForceOut[i] = 0.07228 N - If the tissue type is SPINE then the needle has passed through the epidural space. Output "Wet tap". - If the tissue type is EPIDURAL then needle insertion has stopped in the epidural space. Output "Success".

The focus of this research was to create realistic feeling force-feedback models; therefore no graphical display other than the control screen was incorporated into the simulation. Instead, a life-sized physical model was created of a patient. This physical model was placed over the force-feedback device to hide the hardware and increase the realism of the simulation. The physical model was

created from paper maché overlying a chicken-wire shell and covered with plaster to form a smooth surface, then painted a neutral tone. The physical model was dressed in a patient gown, and a drape used for the epidural procedure was affixed to the back.

Figure 4.11 shows a block diagram of the simulator. Figure 4.12 shows a user with the simulator.

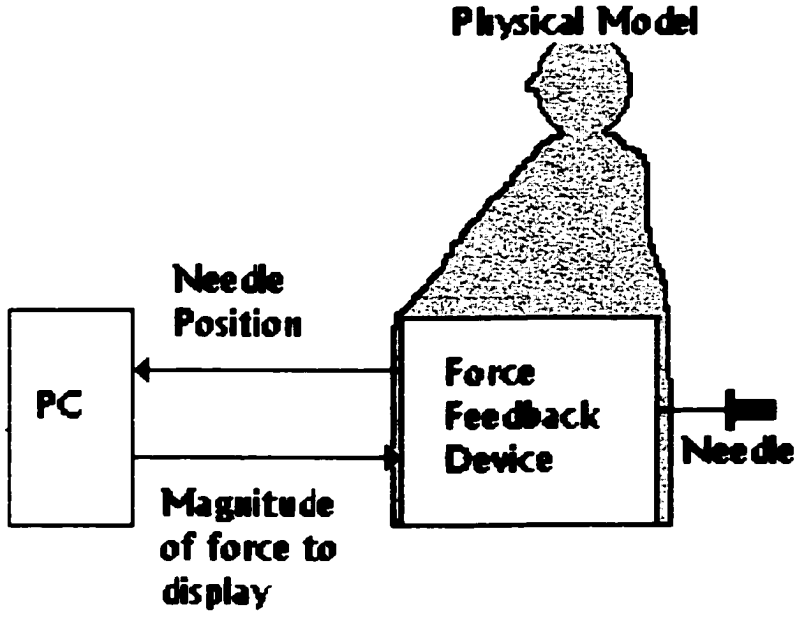

Figure 4.11: Block diagram of the epidural needle insertion simulator.

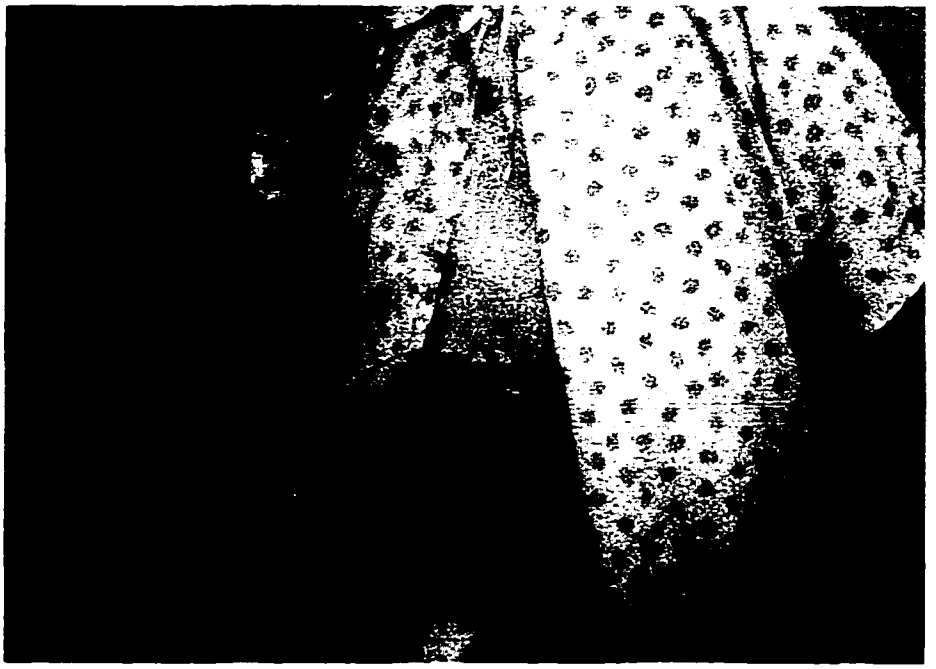

Figure 4.12: User with the haptics-only epidural simulator. The force-feedback device is concealed inside the physical model of the patient.

4.5 Discussion

Although I have made some simplification in the virtual patients and generalizations in the force models, the resultant force-feedback should feel realistic to the user. I am relying on distal attribution, the user's tendency to fill in the gaps between the simulation and reality, to increase the realism of the feel of the forces. Through this process, if the modeled force is similar enough to the force the user expects at each location then these cues will be strong enough cues to be attributed to the physically realistic feeling. For a completely novice user, the gaps in the

simulation will be filled with the expectation of how the student thinks the force will feel.

The biggest weakness in the force models developed to date is that they have been pieced together from data collected from human tissue samples and cadaveric swine studies. However, they do provide a foundation for force models for the epidural needle insertion procedure. The next logical step in developing realistic force models would be to develop an instrumented sterile needle that could be used to measure forces during actual epidural needle insertion procedures. Data collected *in vivo* from live human trials would provide the most accurate force displacement curves possible, and allow for more exact modeling of the needle insertion procedure.

The modeling methodology incorporates differences in patients related to differences in anatomy since it overlays the tissue force equations on top of a MRI data set. To change the virtual patient one just needs to create a new masked MRI data set from the patient's MRI images. This allows for differences in patients due to thickness of skin, body mass index, and anatomical differences. A future topic of research would be to collect *in vivo* needle puncture force data on subjects of different ages in an attempt to quantify age-related differences in tissue characteristics (such as calcification of the ligaments).

The needle withdrawal force was not modeled. Based on observation of the epidural needle insertion procedure, and talking with anesthesiologists I believe that this force is negligible and is also not clinically relevant.

In my research, I assume that the needle insertion force is independent of the velocity of insertion across the range of speeds typical for epidural needle insertion. Correlation studies of load displacement curves for needle insertion into a variety of specimens indicate that my assumption is valid. This conflicts with Brett's assumption that the force is velocity dependent. [Brett, 1997] Another avenue for future research would be to obtain more details of Brett's research and create an exhaustive study to explore the relationship of velocity and needle insertion force across the range of speeds of interest.

In this research, I was limited in my options for force-feedback devices. The force-feedback device used for the simulator was a prototype device developed by and loaned to me by Immersion Corporation. As mentioned in Section 4.4, this device was instable, causing jitters as the needle is inserted. I attempted to compensate for this instability with a damping equation based on the average velocity of needle insertion. A future topic of research would be to perform a stability analysis on the system to better design the damping algorithm. In future simulator development, I would also like to benchmark other force-feedback devices to see if they could more accurately reproduce my force models.

4.6 Conclusions

Specifications for force-feedback models for needle insertion have been developed. A force-feedback modeling methodology has been developed to create force-feedback models of needle insertion based on empirical data collected through biomaterials testing. Data has been collected for a variety of specimens and force

models for needle insertion into oranges and into the lumbar spinal region of the back have been developed using this methodology.

Two haptics-only simulators have been developed, one for the virtual orange and one for a virtual epidural needle insertion procedure. In the next chapter, these simulators will be evaluated.

CHAPTER 5

BENCHMARKING

5.1 Introduction

An important step in the development of a training simulator is to evaluate the ability of the simulation to model the physical task it is attempting to teach. Chapter 2 described the motivation for and the initial creation of the epidural simulator. In Chapters 3 and 4, a force-feedback methodology was developed and force models were created for needle insertion into oranges and into the lumbar spinal region of the back. Code was written to drive a force-feedback device using these models.

This chapter first describes user-testing studies performed in order to compare needle insertion into the virtual oranges developed with the modeling methodology to real oranges. These studies were used to evaluate the realism of the force models. A user study was planned to test the epidural models is described. This study did not take place due to difficulties in creating a usable simulator with the given hardware. The problems with the simulator and potential future evaluation studies are presented. Finally, the simulator is evaluated with respect to the design specifications presented in section 3.3.

5.2 Background

Verification that the simulator is identical to reality is not feasible because of the complexity of the natural system being modeled. There are too many factors involved to fully quantify the entire system. These factors include the complexity of the biomechanics involved in needle puncture, the limitations of the hardware device, and the complexity of the human perceptual system. Oreskes said that the natural system is never closed and that model results are non-unique. Models can be confirmed through a benchmarking process to demonstrate agreement between observation and prediction, but this confirmation is inherently partial and cannot be thought of in absolute terms. [summarized from Oreskes, 1994] However, it is possible to validate that the system has been built according to design specifications; in this case, the specifications outlined in section 3.3.

In a human factors conference paper, Mackenzie states the importance of evaluating simulator learning. He says, " The implementation and design of a simulation may differ from reality in crucial but deceptively subtle ways, changing the ways in which decision and diagnoses are made." He describes a study looking at anesthesiologists interacting with mannequin simulations of a trauma case. These simulators used computer models and hardware devices to re-create pharmacological and physiological responses to medical intervention. In the study, the mannequin's movement of the upper chest during breathing was contraindicative to the diagnoses and caused several of the subjects to misdiagnose the case. The mannequin simulator did not accurately simulate this subtle cue. [Mackenzie, 1999]

This example shows the importance of testing a virtual simulator before using it for medical training. It is necessary to test for any artifacts that are introduced by the simulation, or any key cues that are left out of the simulation. The learning experience may be flawed if the simulator does not simulate reality in key ways,.

Fasse and Hogan published a paper that discussed quantitative measurements of haptic perception. They state that objective quantitative measures of haptic quality are needed in order to compare different implementations of haptic displays. They describe two qualitative approaches for measuring haptic quality. The first is to define a degree of presence in the haptic environment by asking the test subject repeatedly if he/she is present in a given region of the environment. The other approach is to ask the subject to define the degree of reality in the environment. In this type of study, the subject is asked to interact with real and simulated objects, and state how real each object feels. Fasse and Hogan attempt to quantify haptic perception by modeling the metric structure of haptic spatial perception as Riemannnian geometry. They asked subjects to identify length and angles of virtual objects, and they found that the results were not consistent with their model. They found "the haptic perception of angle is distorted and uncertain, but in a way that could not be predicted by knowing the distortion and uncertainty of length perception." [Fasse, 1994]

The field of virtual reality medical simulation is still very new and much of the work being done is in simulator development. There were several papers presented at Medicine Meets Virtual Reality 1998 and 1999 [Westwood, ed., 1998 and

Westwood, ed., 1999] that describe current validation studies for simulators whose development was presented in prior years at the conference.

Gorman presented a paper describing two pilot studies in which surgical residents underwent a standardized training program on a virtual reality surgical trainer. He collected baseline data and demonstrated that learning was taking place by showing improved performance on the trainer. [Gorman, 1998] McCarthy presented a paper in which skill training on an arthroscopy simulator was assessed. He compared the simulator performance of three groups of subjects from beginner, intermediate, and expert skill levels. He found some evidence that the experts performed better on the simulator, but did not test whether the simulator can train the skills needed for the arthroscopy procedure. [McCarthy, 1998]

Smith performed a similar study using a different shoulder arthroscopy simulator, a Procedicus VA from Prosolvia Clarus. In his study, three groups of subjects (Orthopaedic surgeons, non-Orthopaedic surgeons with experience in minimally invasive surgery, and medical students) were required to complete four sets of tasks using the simulator. In the first two tasks, identifying the shoulder anatomy and the subacromial anatomy, the Orthopaedic surgeon group performed better than the other groups. The last two tasks involved touching targets with the force-feedback device acting as a fixed, then mobile, arthroscope. In both of these tasks there were no significant differences between the three groups. The Orthopaedic surgeons performed the task faster, but not more accurately [Smith, 1998].

5.3 Virtual vs. real oranges - man on the street pilot study

5.3.1 Overview

The goal of this study was to test whether the force models developed using the methodology described in Chapters 3 and 4 felt realistic to users. Virtual oranges were created to provide an avenue to test the quality of the force-feedback. The force models were benchmarked through direct comparison of needle puncture into physical oranges and into virtual oranges in a blind randomized trial. This study was performed under Protocol 94H0228. [Appendix A]

5.3.2 Methods

Using the force modeling methodology described in previous chapters and the orange model described in section 3.4.4.2, four virtual oranges were created. Oranges were chosen as the study model because they are readily available and have a complex layered anatomy. Immersion Corporation donated a hollowed out Impulse Engine that was used to hold a real orange in such a way that a needle could be inserted into it. Thus, the subjects were presented with two identical boxes and had no visual cues to the test condition. Figure 5.1 shows the experimental set-up from the rear view.

Figure 5.1: Photo showing experimental set-up, rear view. The device on the right is connected to the PC that will drive the force-feedback. On the left, a hollowed out Impulse Engine holds a real orange. A needle attached to a shaft in front of the orange will puncture the orange as the shaft is pushed.

Twenty-five subjects were recruited for the study. These subjects had no significant prior experience with needle insertion. At the start of the test, the subjects were given a needle and a real orange and were allowed to insert the needle into the orange as many times as they wanted in order to get the "feel" for the needle puncture into an orange. They were also allowed to repeat this process at any time during the testing.

For each trial, the subject was asked to insert needles into one of the Impulse Engines and indicate whether the specimen was a real or virtual orange. The side on which the real orange was placed was randomly chosen for each subject and remained fixed through each of the subject's trials.

The subject was asked to do eight trials, four on the left side Impulse Engine and four on the right. The order of the sides was randomized over the eight trials.

At the end of the test each subject was asked to fill out a survey form.

This study was performed on December 23, 1997 at the Ohio Supercomputer Center.

5.3.3 Results

The goal of this study was to test whether the subjects could differentiate between the real and the virtual models. A summary of the results is given Table 5.1.

	Identified as Virtual	**Identified as Real**
Virtual Model	46	54
Real Model	68	32

Table 5.1: Summary results across subjects and trials.

It is statistically non-tractable to prove that two stimuli are identical; therefore, confidence intervals were used to test for statistical equivalence in this data.

First, the probability that the virtual orange was identified as real was tested. The 95% confidence interval was $CI_{V|R}$= (0.44,0.64). Since this interval includes 0.5 there is evidence that the subjects could not distinguish whether the virtual object was real or virtual.

Next, I tested whether there was a bias towards a subject response of virtual or real when given a real orange as stimulus. The 95% confidence interval for real

but identified as virtual is $CI_{R|V} = (0.59,0.77)$. Since this interval lies to the right of 0.5 there appears to be a bias toward identifying real as virtual.

However, the data do not show a similar bias for the virtual model. The 90% confidence interval for the difference in the probabilities of real identified as virtual minus virtual identified as virtual is $CI_{[R|V - V|V]} = (0.11,0.33)$. Since this interval does not include zero, there does not appear to be a similar bias toward identifying virtual models as virtual.

After completing the eight trials, the subjects were asked to fill out a survey form. On the form nineteen subjects answered which device they thought contained the real orange. Ten of these nineteen subjects (52.6%) correctly identified the real orange. From the results, it appears these subjects could not tell the difference between a virtual model and a real orange.

5.3.4 Discussion

The challenge in developing a simulator for the epidural needle insertion procedure is that this technique relies mainly on haptic sensation for determining the tissue layers that the needle is puncturing. Thus, the force-feedback models must be extremely realistic. This orange realism study was designed to test if the force models developed through the described methodology felt realistic. The results from this study look promising, but are not conclusive.

One limitation with the study was that the subjects were not experienced in inserting needles. The other limitation was that the results were quite difficult to analyze statistically, as the hypothesis under test was that the two oranges felt the

same. Based on what was learned from this study a second study was developed using medical personnel as test subjects. The results of this second study are presented next.

5.4 Virtual vs. real oranges - health care professionals study

The goal of this study was the same as the pilot study, to test whether the force models developed using the methodology described in chapters 3 and 4 realistically simulate the physical world. In addition, this study addressed the limitations of the pilot study by using health care professionals with needle insertion experience as subjects, increasing the number of subjects, and changing the orange presentation to obtain statistically testable data. The hypothesis being tested was:

> Hypothesis: Hospital personnel with experience in needle insertion cannot tell the difference between inserting a needle into a real orange and into a virtual orange.

5.4.1 Methods

The virtual oranges were the same oranges developed for the pilot study. These four virtual oranges were created from the orange model described in section 3.4.4.2. The physical device setup was the same for this study as it was for the pilot study. Thus, the subjects were presented with two identical boxes and had no visual cues to the test condition.

Fifty subjects were recruited for the study from the medical personnel at the OSU Hospital Department of Anesthesiology. The subjects were medical students, residents, nurses, and attendings all of whom had experience in inserting needles. The equipment was set up in the resident break room and data was collected on

April 21-22, 1999. This study was performed under Protocol 99H0065. [Appendix A]

After signing consent forms, the subjects were given a needle and a real orange and were allowed to insert the needle into the orange as many times as they wanted in order to get the feel for the needle puncture into an orange. They were also allowed to repeat this process at any time during the testing.

For each trial, the subject was presented with the two Impulse Engines and asked to insert the needle into both devices and identify which of the pair was the virtual orange and which was the real orange. I recorded whether they correctly identified the oranges. Each subject was asked to do eight trials. The side on which the virtual orange was placed was randomized for each trial. Each subject was presented with the four virtual oranges two times each, and four real oranges two times each, for a total of eight trials per subject. At the end of the test each subject was asked to fill out a survey form.

Figure 5.2 shows the test setup both from the user's viewpoint and from the test administrator's viewpoint.

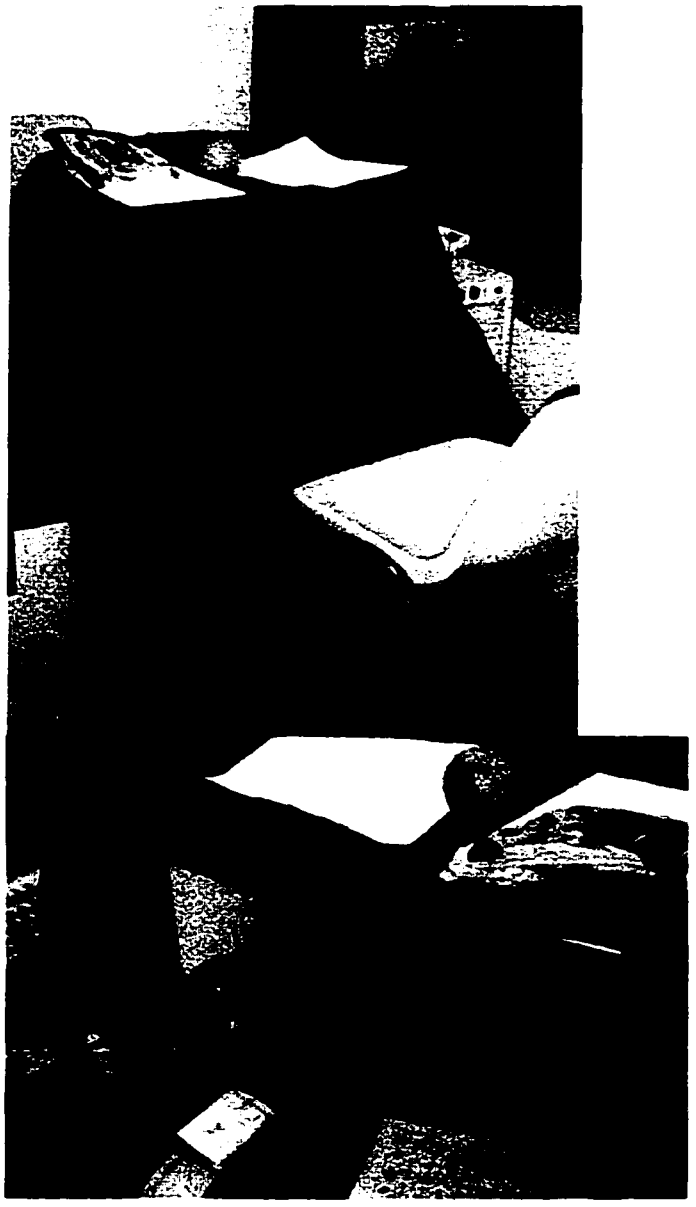

Figure 5.2: The setup for the orange study. The top picture shows the front of the test station, where the user sits in the chair and inserts the needles. The user cannot see any part of the mechanics of the setup. The picture on the bottom is from the viewpoint of the administrator. In this case, the real orange is on the right and the virtual orange is on the left.

5.4.2 Results

The number of correct responses for each subject was stored in an Excel spreadsheet and the results were plotted as a histogram, as shown in Figure 5.3.

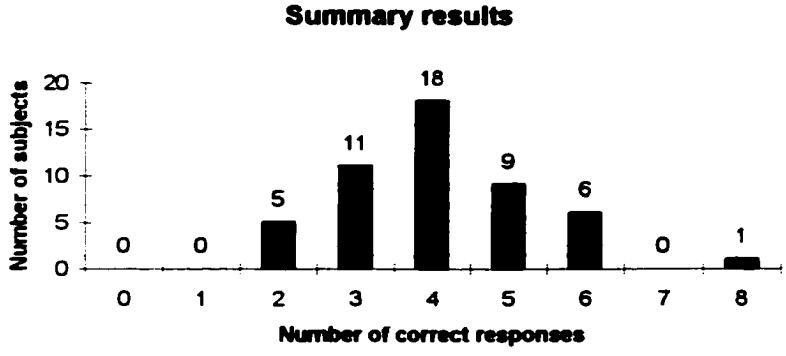

Figure 5.3: Summary results for the orange benchmark trials.

The hypothesis under test was that the subjects were unable to discriminate between the virtual orange trials and the real orange trials. To show this hypothesis was true, it must be shown that the subject's response data could have been taken from a normal distribution with a mean of four and a sample size of fifty. Using Minitab, I tested for normality using the Kolmogorov-Smirnov Test ($D = 0.035$, p-value > 0.15) and the Anderson-Darling Test ($R = 0.9937$, p-value > 0.1). Because the p-value is large in both of these tests it can be concluded that the data passes the normality test and the hypothesis is true.

5.4.3 Discussion

Testing the realism of the force-feedback models is important in order to create a computer simulation that closely resembles needle insertion into a real object. The results indicate that the subjects were unable to distinguish between the real oranges and the virtual orange models. This indicates that the force modeling methodology was sound for creating a realistic feeling virtual model.

Several subjects did indicate on their survey forms that one of the models felt "smoother", though they could not tell if it was the real orange or the virtual orange. As the tester, I could tell that the "smoother" orange was the virtual orange. This could be because the force models are a curve fit to real data, in effect smoothing all low frequency noise that could be caused by the needle going through the orange corpuscles. This led to the incorporation of a small random noise term into the epidural force models.

5.5 Construct validity study

A construct validity study was planned in order to validate the haptics-only epidural needle insertion simulator. Construct validity refers to the degree to which the simulation recreates the real epidural needle insertion experience. Anesthesiologists at three levels of experience with the epidural procedure were to be asked to use the simulator and their performance and comments were to be recorded. The hypothesis under test was that the subject's performance on the simulator would correlate with their skill level. To show construct validity, there must be a positive correlation between skill on the simulator and skill in reality. This study was to be performed under Protocol 99H0150. [Appendix A]

The study was not performed as planned because the force-feedback device was not able to accurately displace the force models in a realistic feeling fashion. In the methods section, details of the planned study will be presented. The results and discussion sections will present what went wrong, and potential future research to resolve these problems.

5.5.1 Methods

The haptics-only epidural needle insertion simulator was developed as described in section 4.4. A Pentium PC was borrowed from OSC to use as the host computer for the simulator. Attached to the PC was a force-feedback device borrowed from Immersion Corporation, with a physical torso dummy placed overtop of the device to add to the visual realism of the system.

Subjects were to be recruited for this study from the medical personnel at the OSU Medical Center and would have had prior experience in needle insertion. This subject population was chosen because of their skill with needle insertion. The goal of this study was to test the realism of the epidural simulator, not the subject's ability to insert needles. The subjects were to be voluntarily asked to participate in the study and were to be from three different populations, 1) Experts in the epidural procedure, 2) Intermediates (who have performed 5-10 epidural procedures), and 3) Novices who have experience in needle insertion but no experience with the epidural technique.

The subjects would have been asked to insert an epidural needle into the virtual reality simulated patient for each trial. The subjects were to perform eight trials, four for each virtual patient. The presentation order of the virtual patients

would be randomized. The subjects were to be asked to vocally describe what they were doing and feeling as they inserted the needle. Success/failure for each stick was to be recorded along with time taken to insert the needle. At the end of the study each subject was to be asked to fill out a survey form with questions about the realism of the force-feedback in general, and for each tissue layer in particular.

Time to insert and success rate were to be compared across the three subject groups to see if there was a significant difference. Also, the surveys were to be analyzed to test the effectiveness of the simulator, and the perceived realism. The data were to be analyzed to look for either a quantitative measure of validity (shown by a difference in needle insertion across our three subject groups) or a qualitative measure through the survey forms.

5.5.2 Results

As mentioned, the study did not proceed as planned. There were several difficulties in getting the force-feedback device to accurately portray the models. First, the motors on the device were only capable of a maximum output of 7.4N, which was significantly less than the maximum force for the force models of ligamentum flavum and bone. To deal with this difficulty, a threshold of 7.4N was set for the force output in the simulation. If a calculated force was larger than the threshold force, it was set to 7.3N to prevent burning out the motors on the device.

The second problem with the device was that the motors could be overdriven. So, it was not possible to display the bone model on the device. At the motor output of 7.0N, the user could continue to push the needle forward with

resistance. This is dramatically different than reality where bone is a solid structure incapable of needle penetration.

The third, and most problematic difficulty, was that when the force models showed a sharp increase in force the physical result was that the needle jittered in the user's hand unless a constant forward pressure was applied. This was possibly because of an insufficient bandwidth in the display of the forces due to the device and in the communications between the device and the computer.

The device was also very active, it was not well compensated. The effect of this on the needle shaft was that even when the output force was set to 0.0N, the device tended to want to spring the needle shaft forwards, towards the center of the device. The result was that a small negative force, along with the user holding the needle, was needed to keep the needle shaft extended outside the body. If the user did not maintain a hold on the needle, the conflict between the pushing force and the needle's tendency towards the center position would cause the shaft to jitter at the frequency of the update rate of the motor commands.

The result of these difficulties was that as the needle was inserted into the virtual patient, it jittered unless it was inserted with a constant forward motion. This forward motion could be either slow or fast, it was not velocity dependent. Instead, a forward pressure needed to be maintained on the needle to overcome the machine's jitters. This motion is not the natural way to insert the epidural needle. In actual practice, the needle is inserted until a resistance is felt, and then the anesthesiologist pushes at the resistance until he/she is comfortable with what layer

it is in, then a forward motion is used to push through the layer and this process is repeated at the next resistance.

During development of the simulator, I managed to get the force-feedback device to behave on a constant needle insertion motion and hoped that would be sufficient for the study. On August 12th, 1999, the simulator was set up in the resident break room at the OSU Hospital Department of Anesthesiology. After five subjects used the simulator it was apparent that the simulator was not acceptable because in order to use it, the subjects had to perform the procedure in a non-realistic fashion.

5.5.3 Discussion

I think that the study, as planned, would have shown insight into whether the simulator had construct validity. However, it was quickly apparent that the simulator, as designed with the current prototype force-feedback device, did not have construct validity. Therefore, it was not an acceptable method to test the realism of the force models for the epidural procedure.

One of the limitations with my research has been a lack of funding. I have been fortunate to get access to a lot of equipment, and several people have donated their time to my work, but what was really needed was funding to either develop or locate a force-feedback device that would have worked better. The device I borrowed was a prototype, and not guaranteed to be able to run the models. It was unfortunate that it could not.

A future goal would be to obtain funding or the facilities to develop the simulator on different hardware. Some of the problems, like the difficulty with

virtually constructing a hard surface contact, are still outstanding problems in the field. Others, such as the need for increased bandwidth, could be resolved with a faster data communication channel or with a smart force-feedback device that could store the force array on a local processor such that the update rate would be maximized with local data communications.

Another future feature that would greatly enhance realism would be to incorporate a loss-of-resistance cue, such as attaching a water filled syringe to the needle with a smart controller to open a valve upon puncturing the ligamentum flavum. The subjects who used the simulator did indicate that they rely heavily on this cue for sensing the current tissue layer the needle is puncturing. This feature would be reasonably inexpensive to incorporate. The main cost would be in the design of the circuit card to control the valve.

5.6 Comparison of simulator to the design specifications.

Design specifications (DS) for the force modeling methodology were presented in section 3.3. In this section, these specifications will be compared to the resultant force-feedback methodology for validation.

Force-feedback models were developed based on the results of data collected during biomaterials testing on a MTS machine (met DS-A). To create a virtual subject, these force-feedback models are overlaid upon MRI data sets. This allows one to easily create several patients with differences in thickness of skin, body mass index, and anatomical differences (met DS-B). However, further biomaterials testing is needed to quantify differences due to age and to incorporate these differences into the force models.

Force models were created for each tissue type encountered in the epidural needle insertion procedure from the puncture force curves obtained during MTS testing (met DS-G), and using high resolution MRI for registration of the puncture curves to the internal tissue layers (met DS-F). There are some limitations to the force models and these were discussed as each force model was presented in section 4.2 (met DS-D). Assumptions required to create the models were identified in section 3.4.3 and tested for validity in section 3.4.4 (met DS-E).

A study was performed to identify the minimum perceivable force using the Immersion Impulse Engine (met DS-C).

Using the force-feedback methodology, two virtual patients were created (met DS-H), along with two virtual oranges (met DS-I).

A haptics-only simulation was developed for the epidural needle insertion procedure (met part of DS-J). However, the simulation did not meet Massie's criteria for realistic feeling haptic sensations: free space did not feel free, solid objects did not feel stiff, and the virtual constraints were easily saturated. The system also was not stable while inserting the needle (did not meet the rest of DS-J).

5.7 Discussion

The studies presented in this chapter were used to examine the validity of the force models developed in chapters 3 and 4. The two orange studies offer evidence that the force modeling methodology results in virtual objects that cannot be distinguished from the real objects. The construct validity study shows that it is important to use a force-feedback device that is capable of accurately displaying the models, which have been developed. With an adequate budget, care should be

taken to benchmark several devices and find one that best displays the force models.

The design specifications have been reviewed for verification of the force modeling methodology with the original design criteria. As mentioned in the discussion of Chapter 4 there are several future studies which could be undertaken to improve the accuracy of the models.

I believe that user testing is a very important step in the design of virtual reality simulations. It is through such studies that the realism of the simulator and its effects on user training can be evaluated. This step is also important in designing training protocols that can use the simulator to educate and enhance medical training.

BIBLIOGRAPHY

Abragam A, Principles of Nuclear Magnetism, Clarendon Press, ISBN 019852014X, (1966); 264-353.

Bicci A, Canepa G, De Rossi D, Iacconi P, Scilingo E, "A Sensorized Minimally Invasive Surgical Tool for detecting Tissuatal Elastic Properties", in Proceedings of the 1996 IEEE International Conference in Robotics and Automation, Minneapolis, MN, (April 1996); 884-888.

Blezek DJ, Robb RA, "Haptic rendering of isosurfaces directly from medical images", in Medicine Meets Virtual Reality, JD Westwood et al. (Eds.), IOS Press, (1999): 67-73.

Bonica, JJ, McDonald JS., "Chapter 13: Epidural Analgesia and Anesthesia", Principles and Practice of Obstetric Analgesia and Anesthesia, Second Edition, Williams & Wilkins, Baltimore, 1995; 344-470.

Bostrom M, Singh S, Wilet, Dr. CW, "Design of an Interactive Lumbar Puncture Simulator with Tactile Feedback", IEEE Virtual Reality Annual International symposium, September 18-22, 1993; 280-286.

Brett PN, Parker TJ, Harrison AJ, Thomas TA, Carr A, "Simulation of resistance forces acting on surgical needles", Proceedings Institution of Mechanical Engineers, V211H; (1997):335-347.

Brown C, "Edwin A. Link and the Air Age", Progress, Technology, and the Romance of Motion, Roberson Museum and Science Center, Binghamton, New York, 1994.

Burdea GC., "Force and Touch Feedback for Virtual Reality", John Wiley & Sons, Inc., (1996).

Caputy A, Starr J, Reidel C, "Video-Assisted Endoscopic Spinal Surgery: Thorascopic Discectomy", Acta Neurochirurgica, (1995) 134: 196-199.

Chu K, Rutt B, "Polyvinyl alcohol cryogel: an ideal phantom material for MR studies of arterial flow and elasticity.", Magnetic Resonance in Medicine, vol. 37, no.2: (1997); 314-319.

Delingette H, "Toward Realistic Soft-tissue Modeling in Medical Simulation", Proceedings of the IEEE, Vol. 86, No. 3: (March 1998); 512-523.

Doyle WK, "Low End Interactive Image-Directed Neurosurgery", In Health Care for the Information Age, Proceedings of MMVR4, Weghorst SJ, Sieburg HB, Morgan KS (Eds.): IOS Press, Amsterdam ; Washington, DC : (1996); 1-11.

Durlach N I., Mavor A S., eds. "Virtual Reality: Scientific and Technological Challenges", National Academy Press, (1995); 161-187.

Eriksson E, Larsson N, Nitescu P, Appelgren L, Linder LE, Curelaru I, "Penetration Forces in Cannulation of the Dorsal Veins of the Hand: I. A Comparison Between Polyurethane (Insyte®) and Polytetrafluoroethylene (Venflon®) Cannulae", Acta Anaesthesiol Scand, 35 (1991); 306-314.

Fasse ED, Hogan N, "Quantitative Measurement of Haptic Perception", Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, (May 8-13, 1994): 3199-3204.

Gaiser RR, "Spinal, Epidural, and Caudal Anesthesia", In Introduction to Anesthesia, eds. Longnecker DE, Murphy FL, Ninth Edition, Dripps/Eckenhoff/Vandam: (1992); 221.

Gorman Dr. PJ, Lieser JD, Murray WB, Haluck RS, Krummel TM, "Evaluation of Skill Acquisition Using a Force Feedback, Virtual Reality Based Surgical Trainer" , in Medicine Meets Virtual Reality, JD Westwood et al. (Eds.), IOS Press, (1999): 121-123.

Hasser CJ, Daniels MW, "Tactile Feedback with Adaptive Controller for a Force-Reflecting Haptic Display. Part 1: Design", Proceedings of the 15th Southern Biomedical Engineering Conference, ed. Bajpai PK, Dayton, Ohio, (March 29-31, 1996);526-530.

Hiemenz L, Stredney D, Schmalbrock P, "Development of the Force Feedback Model for an Epidural Needle Insertion Simulator", In: Westwood JD, et al, eds. Medicine Meets Virtual Reality, IOS Press and Ohmsha, Amsterdam, 272, (1998).

Hiemenz L, Litsky A, Schmalbrock P, "Puncture Mechanics for the Insertion of an Epidural Needle", Proceedings of the 21st Annual Meeting if the American Society of Biomechanics, Clemson University, (Sept. 24-27, 1997); 36-37.

Hiemenz L., McDonald JS, Stredney D, Sessanna D, "A Physiologically Valid Simulator for Training Residents to Perform an Epidural Block", Proceedings of the 15th Southern Biomedical Engineering Conference, Dayton, Ohio, (March 29-31, 1996); 170-173.

Hodges M, "Virtual Reality in Training", Computer Graphics World, V21, No. 8, (August, 1998); 45-52.

Horlocker TT, "Epidural Anesthesia", Anesthesiology Reviews, ed. Faust RJ., Second Edition, Churchill Livingstone, New York (1994); 325-327.

Horlocker TT, Cucciara RF, Ebersold MJ, "Vertebral column and spinal cord surgery", In Cucchiara RF, Michenfelder JD (Eds.): Clinical Neuroanesthesia. Churchill Livingstone, New York, (1990); 325.

Immersion Corporation, San Jose, CA, http://www.immerse.com/, (1999).

Keller, PJ, "Basic Principles of Magnetic Resonance Imaging", GE Medical Systems, (1988).

Levine, MW, "Fundamentals of Sensation and Perception", Addison-Wesley Publishing Company, Inc., ISBN 0-201-04339-4, (1981); 11-33.

Massie TH, Salisbury JK, "The PHANToM Haptic Interface: A Device for Probing Virtual Objects", Dynamic Systems and Control, Radcliffe CJ ed.; (1994) 1:295-299.

Mackenzie CF, Harper BD, Xiao Y, "Simulator Limitations and Their Effects in Decision-making", Proc. Human Factors and Ergonomics 40th Annual Meeting, (1996); 747-751.

McCarthy A, Harley P, Smallwood R, "Virtual Arthroscopy Training: Do the "Virtual Skills" Developed Match the Real Skills Required?", in Medicine Meets Virtual Reality, JD Westwood et al. (Eds.), IOS Press, (1999): 221-227.

McDonald JS, Rosenberg LB, Stredney D, "Virtual Reality Technology Applied to Anesthesiology", In: Satava et al, eds. MMVR 3: Interactive Technology & the New Paradigm for Healthcare, IOS Press, Amsterdam, (1995); 237-243.

Merriam Webster's Collegiate Dictionary, 10th Edition, Merriam-Webster, Inc., Springfield, MA, (1994); 643.

Mieklejohn BH, "Distance from the Skin to the Lumbar Epidural Space in an Obstetric Population", Regional Anesthesia, V.15, No. 3, (May-June, 1990); 134-136.

Minsky M, Ouh-Young M, Steele O, Brooks, Jr. F, Behensky M, "Feeling and Seeing: Issues in Force Display", Computer Graphics, Special issue on the 1990 symposium of 3D Graphics, Vol. 24, No. 2 (March, 1990); 235-243.

Nemire K, Burke A, Jacoby R, "Human Factors Engineering of a Virtual Laboratory for Students with Physical Disabilities", Presence, Vol 3, No 3 (1994); 216-226.

Oreskes N, Shrader-Frechette K, Belitz K, "Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences", Science, V 263, (4 Feb, 1994); 640-646.

Raj PP, Handbook of Regional Anesthesia, Churchill-Livingstone, New York; (1985).

Reinig KD, Rush CG, Pelster HL, Spitzer VM, Heath JA, "Real-time Visually and Haptically Accurate Surgical Simulation", In: Weghorst et al, eds. MMVR 4: Healthcare in the Information Age, IOS Press, Amsterdam, 1996.

Rosen J, MacFarlane M, Richards C, Hannaford B, Sinanan M, "Surgeon-Tool Force/Torque Signatures - Evaluation of Surgical Skills in Minimally Invasive Surgery" , in Medicine Meets Virtual Reality, JD Westwood et al. (Eds.), IOS Press, (1999): 290-296.

Rosenberg LB, "Virtual Fixtures": Perceptual overlays enhance operator performance in telepresence tasks, Ph.D. Dissertation, Stanford University, June 1994.

Sack, WO, Essentials of Pig Anatomy; Horowitz/Kramer Atlas of Musculoskeletal Anatomy of the Pig, Veterinary Textbooks, Ithaca, New York; (1982).

SensAble Technologies, Cambridge, MA, http://www.sensable.com, (1999).

Shareef N, Wang D, Yagel R, "Segmentation of Medical Data using Locally Excitatory Globally Inhibitory Oscillator Networks", World Congress on Neural Networks WCCN '96, San Diego, California (September, 1996).

Shimoga, K, "Finger Force and Touch Feedback Issues in Dextrous Telemanipulation," Proceedings of NASA-CIRSSE International Conference on Intelligent Robotic Systems for Space Exploration, NASAm Greenbelt, MD, (September, 1992).

Sitzman BT, Uncles DM, "The Effects of Needle Type, Gauge, and Tip Bend on Spinal Needle Deflection", in Wedel DJ ed., Regional Anesthesia and Pain Management; Anesthesia Analgesia, Vol. 82 (1996) 297-301.

Smith S, Wan A, Taffinder N, Read S, Emory R, Darzi A, "Early Experience and Validation Work with Procedicus VA - The Prosolvia Virtual Reality Shoulder Arthroscopy Trainer" , in Medicine Meets Virtual Reality, JD Westwood et al. (Eds.), IOS Press, (1999): 337-343.

Stredney D, Sessanna D, McDonald JS, Hiemenz L, Rosenberg LB, A Virtual Simulation Environment for Learning Epidural Anesthesia. In: Weghorst et al, eds. MMVR 4: Healthcare in the Information Age, IOS Press, Amsterdam, 1996; 164-175.

Tan HZ., Pang XD, Durlach NI, "Manual Resolution of Length, Force, and Compliance", Proceedings of the ASME Winter Conference: Advances in Robotics, (1992); 13-18.

Towler MA, McGregor W, Rodeheaver GT, Cutler PV, Bond RF, Phung D, Morgan RG, Thacker JG, Edlich RF, "Influence of Cutting Edge Configuration on Surgical Needle Penetration Forces", The Journal of Emergency medicine:V6 (1988); 475-481.

Westbrook JL, Uncles DR, Sitzman BT, Carrie LE, "Comparison of the Force Required for Dural Puncture with Different Spinal Needles and Subsequent Leakage of Cerebrospinal Fluid", Anesthesia Analgesia: Vol. 79 (19914); 769-772.
Notes:

Westwood JD, et al. (Eds.), Medicine Meets Virtual Reality, Art, Science Technology: Healthcare (R)evolution, IOS Press, (1998).

Westwood JD, et al. (Eds.), Medicine Meets Virtual Reality, The Convergence of Physical & Informational Technologies: Options for a New Era in Healthcare, IOS Press, (1999).

Zarzur E, MD., "Anatomic Studies of the Human Ligamentum Flavum", Anesthesia Analgesia Vol. 63, (1984); 499-502.

APPENDIX A

STUDY PROTOCOLS

91H0301	Submillimeter Resolution Imaging Using a Magnetic Resonance Device. Petra Schmalbrock, Ph.D., Ohio State University, Department of Radiology
94H0228	The Investigation of the Use and Efficacy of Advanced and Virtual Computer Interfaces for Biomedical Applications. Don Stredney, Ohio Supercomputer Center
94H0333	Epidural Simulator Efficacy Study. John S. McDonald, Robert H. Small, Leslie L. Hiemenz, OSU Medical Center, Department of Anesthesiology
94H0415	The Acquisition of Structural and Force Data from Cadaver Specimens for Use in Virtual Surgical Simulations. Don Stredney, Ohio Supercomputer Center
99H0065	Testing the Realism of the Force Models for a Needle Insertion Simulator. Robert H. Small, Leslie L. Hiemenz, OSU Medical Center, Department of Anesthesiology
99H0150	Validation of the Epidural Needle Insertion Simulator. Robert H. Small, Leslie L. Hiemenz, OSU Medical Center, Department of Anesthesiology

APPENDIX B

C++ SOURCE CODE FOR THE HAPTICS-ONLY EPIDURAL NEEDLE INSERTION SIMULATOR

```
/* File:       imptest.cpp

 Description:  Using the new prototype Immersion device, implement the epidural
        simulator. Reads in the g_masked MRI data set and calculates a force curve
        based on my tissue equations (see Ch 2 of the diss for derivations).

        ISSUES:
        - BONE???  FFD motors are not strong enough to implement hitting bone.
                        Use max force

        Modified from Immersion demo code for the Impulse Engine.
 */

// included files
#include <conio.h>
#include <assert.h>
#include "..\Include\impulse.h"
#include "device.h"
#include "resource.h"
#include "hThread.h"
#include "fstream"
#include <iostream.h>
#include <iomanip.h>
#include <math.h>

// module definitions
#define NUM_AXES_DISPLAYED 3
#define NUM_DIGS_DISPLAYED 6
#define NUMPATIENTS      4
#define X_AXIS           0x01
#define Y_AXIS           0x02
#define Z_AXIS           0x04
#define DEVICE_RES       -0.003614
#define FORCE_RES        1000

// model definitions
#define SKINSURFACE      10
#define SKINMAX          6.0372
#define FATFORCE         1.974
#define LFMAX            6.1330
#define LFMIN            2.1
#define SPINEFORCE       2.437
#define EPIDFORCE        2.437
#define PIGCORRECTION    4.053
#define MUSCLEFORCE      7.728
```

```
#define ISLFORCE        11.52
#define MAXFORCE        20.0
#define MUSCLEMAX       8.407

// definitions to identify tissue g_masks in MRI data set
#define AIR             0
#define SKIN            9
#define FAT             6
#define MUSCLE          1
#define ISL             8
#define LF              7
#define BONE            2
#define MISCTISSUE      5
#define EPID            4
#define SPINE           3
#define UNDEF           10

// macro definitions
#define SPRING_IS_ENABLED  (SendDlgItemMessage(g_hwnd,
IDC_SPRINGENABLE, BM_GETCHECK, 0, 0))


// Module global variables
HINSTANCE       g_hInst;
HWND            g_hwnd;
UINT            g_timerID;
HANDLE          g_hSemaphore;

CDevice*        g_pDeviceArray[IE_MAXIMUM_DEVICES];
INT             g_NumDevices;
CDevice*        g_pCurrentDevice;
DEVICE_INPUT    g_DeviceInput;

UINT            g_bSpringEnabled;
CHapticThread   g_SpringThread;

// Function prototypes
BOOL CALLBACK   DialogFunc      (HWND, UINT, WPARAM, LPARAM);
BOOL            OnInit          (HWND hwnd);
BOOL            OnExit          ();
BOOL            OnSetCenter     ();
BOOL            OnSetTrajectory ();
BOOL            OnTimer         ();
BOOL            OnPatientCombo  (WORD msg);
BOOL            OnSpringEnable  ();
BOOL            OnDACOutput     ();
```

```
void            read_mask(int patientIndex);
double          averageZVelocity(int instVelocity);

// Force feedback global variables
BOOL            g_trajSet = false;
double          g_forceVector[FORCE_RES];
int             g_tissueVector[FORCE_RES];
int             g_mask[330][220]; // x, z
int             g_xmax, g_zmax;
double          g_xres, g_zres;
BOOL            g_done = false;
BOOL            g_tooFar = false;
BOOL            g_hitSpace = false;

// inline functions

// Function: fAquireSemaphore
// Description: gets the one, global semaphore of this application.
//          We will use this function, along with fReleaseSemaphore
//          to ensure that only one of our two threads is accessing
//          the device at any time.

inline BOOL fAquireSemaphore(void)
{
   DWORD dwResult;

   dwResult = WaitForSingleObject(g_hSemaphore, 500);

   assert(dwResult == WAIT_OBJECT_0);

   return(dwResult == WAIT_OBJECT_0);
}

// Function: fReleaseSemaphore
// Description: releases the one, global semaphore of this application

inline BOOL fReleaseSemaphore(void)
{
   BOOL bResult = ReleaseSemaphore(g_hSemaphore, 1, NULL);
   assert(bResult);
   return bResult;
}

// WinMain
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
            LPSTR szCmdLine, int nWinMode)
```

```
{
   UINT uResult;
   BOOL bResult;

   // initialize globals
   g_hInst          = hInstance;
   g_hwnd           = NULL;    // this will be set during WM_INITDIALOG message
handling
   g_bSpringEnabled   = FALSE;

   // Initialize the API
   uResult = Imp_Initialize(0);

   if ( uResult != IE_SUCCESS )
   {
      // we failed Initialization, let's get out of here.
      MessageBox( NULL,
                  "Error: Initialization FAILED",
                  "Error",
                  MB_OK | MB_ICONSTOP );
      return 0;
   }

   // Open the devices: note that we can get the success of these calls
   //    later through the IsInitialized method

   CDevice* pDevice;
   g_NumDevices = 0;
   for (int i=0; i<IE_MAXIMUM_DEVICES; i++)
   {
      pDevice = new CDevice();
      bResult = pDevice->Open(IE_DEVICE_N + i);
      if (bResult)
         g_pDeviceArray[g_NumDevices++] = pDevice;
      else
         delete pDevice;
   }

   // if we didn't find any devices, abort.
   if (g_NumDevices == 0)
   {
      MessageBox( NULL,
                  "Error: No devices found on this system.",
                  "Error",
                  MB_OK | MB_ICONSTOP );
      Imp_End();
```

```
        exit(1);
    }

    DialogBox(hInstance, MAKEINTRESOURCE(IDD_MAIN), NULL, (DLGPROC)
DialogFunc);


    // close and delete all devices
    for (i=0; i<g_NumDevices; i++)
    {
        g_pDeviceArray[i]->Close();
        delete g_pDeviceArray[i];
    }

    Imp_End();
    return 0;
}


// Function:    DialogFunc
// Description: This is the Callback used in our DialogBox call.  It is serving
//              as  the main window loop in our application
// Returns:     BOOL- TRUE if message was handled

BOOL CALLBACK DialogFunc(HWND hwnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    switch (message)
   {
   case WM_INITDIALOG:
        return OnInit(hwnd);

    case WM_COMMAND:
        switch(LOWORD(wParam))
        {
        case ID_SETCENTER:
            return OnSetCenter();

        case ID_SETTRAJECTORY:
            return OnSetTrajectory();

        case IDOK:
        case IDCANCEL:
            return OnExit();

        case IDC_SPRINGENABLE:
```

```
            return OnSpringEnable();

        case IDC_DEVICECOMBO:
            assert( (HWND)(lParam) == GetDlgItem(hwnd, IDC_DEVICECOMBO));
            assert( LOWORD(wParam) == IDC_DEVICECOMBO);
            return OnPatientCombo(HIWORD(wParam));

        case IDC_DACOUTPUT:
            return OnDACOutput();
        }
        return FALSE;

    case WM_TIMER:
        return OnTimer();
    }

    return FALSE;
}


// Function:    OnInit
// Description: This function is called in response to the WM_INITDIALOG
//              message.
// Returns:     BOOL- this will returned from the dialog function, for
//              a WM_INITDIALOG message, we should return FALSE iff we
//              explicitly SetFocus to one of our controls.

BOOL OnInit(HWND hwnd)
{
    // store our window handle for easy access
    g_hwnd = hwnd;

    // create our semaphore
    g_hSemaphore        = CreateSemaphore(NULL, 1, 1, "ImpTestSemaphore");

    // Timer is only for display purposes, so 50 Hz (20 ms) should be fine.
    g_timerID = SetTimer(g_hwnd, NULL, 20, NULL);


    // initialize the combo box
    char buffer[256];
    for (int i=0; i<NUMPATIENTS; i++)
    {
                if (i == 0)
                        wsprintf(buffer, "%d) Average Female, Slice 0", i);
                else if (i == 1)
```

```
                wsprintf(buffer, "%d) Average Female, Slice 1", i);
            else if (i == 2)
                wsprintf(buffer, "%d) Skinny Male, Slice 0", i);
            else if (i == 3)
                wsprintf(buffer, "%d) Skinny Male, Slice 1", i);

SendDlgItemMessage(g_hwnd, IDC_DEVICECOMBO,
        CB_ADDSTRING, 0, (LPARAM)buffer);
    }
        // Choose average female patient, slice 0 to start
        SendDlgItemMessage(g_hwnd, IDC_DEVICECOMBO, CB_SETCURSEL,
0, 0);
        read_mask(0);

        g_trajSet = false;

    // Choose Device 0
    g_pCurrentDevice = g_pDeviceArray[0];

    // set AdcEdit Boxes to 0
    for (i=0; i<3; i++)
        SetDlgItemText(g_hwnd, IDC_DAC0 + i, "0");

    return TRUE;
}

// Function:    OnExit
// Description: This function is called in response to the ID_OK button
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL OnExit()
{
    if (SPRING_IS_ENABLED)
    {
      // This will end the Spring Thread
        // clear the checkbox
        SendDlgItemMessage( g_hwnd, IDC_SPRINGENABLE, BM_SETCHECK,
 0, 0);
        // notify ourselves of the change:
        OnSpringEnable();
    }

    KillTimer(g_hwnd, g_timerID);

    // close the semaphore
```

```
    BOOL bResult;
    bResult = CloseHandle(g_hSemaphore);
    assert(bResult);

    EndDialog(g_hwnd, 0);
    return TRUE;
}


// Function:    OnDACOutput
// Description: This function is called whenever the  Output
//              button is pushed
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.
BOOL    OnDACOutput()
{
 // resets the program
        if (g_tooFar)
                MessageBox( NULL,
                 "Wet Tap",
                 "Punctured Dura",
                 MB_OK | MB_ICONINFORMATION);
        else if (g_hitSpace)
                MessageBox( NULL,
                 "SUCCESS",
                 "Stopped at epidural space",
                 MB_OK | MB_ICONINFORMATION);
        g_tooFar = false;
        g_hitSpace = false;
        g_done = false;
        g_trajSet = false;
    return TRUE;
}


// Function:    OnPatientCombo
// Description: This function is called whenever the Patient Combo Box is clicked.
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL OnPatientCombo(WORD msg)
{
    int         patientIndex;

    if (msg == CBN_SELCHANGE)
    {
```

```
      patientIndex = SendDlgItemMessage(g_hwnd, IDC_DEVICECOMBO,
CB_GETCURSEL, 0, 0);
        read_mask(patientIndex);
        g_trajSet = false;
        }

   return TRUE;
}

// Function: read_mask
// Description:  Reads in the patient data set.  For now, it can be
// either drew or me, and slice 0 or 1.

void read_mask(int patientIndex)
{
        int x,z;
        FILE *g_maskFile;

        if (patientIndex == 2)
        {
                // read in male (v2: 330 x 220)
                g_maskFile = fopen("v2/mask1.dat","r");
                g_xmax = 330;
                g_zmax = 220;
                g_xres = 0.7;
                g_zres = 0.3;
        }

        else if (patientIndex == 3)
        {
                // read in male (v2: 330 x 220)
                g_maskFile = fopen("v2/mask2.dat","r");
                g_xmax = 330;
                g_zmax = 220;
                g_xres = 0.7;
                g_zres = 0.3;
        }
        else if (patientIndex == 0)
        {
                // read in female (v1: 305 x 100)
                g_maskFile = fopen("v1/mask1.dat","r");
                g_xmax = 305;
                g_zmax = 100;
                g_xres = 0.5;
                g_zres = 0.7;
        }
```

```
        else if (patientIndex == 1)
        {
                // read in female (v1: 305 x 100)
                g_maskFile = fopen("v1/mask2.dat","r");
                g_xmax = 305;
                g_zmax = 100;
                g_xres = 0.5;
                g_zres = 0.7;

        }

        int temp;
        for (z = g_zmax-1; z >= 0; z--)
        {
                for (x = 0; x < g_xmax; x++)
                {
                        fscanf(g_maskFile,"%i",&temp);
                        g_mask[x][z] = temp;
                }
        }


        fclose(g_maskFile);
}


// Function:    OnSetCenter
// Description: This function is called whenever the Set Center button
//              is clicked.
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL OnSetCenter()
{
   BOOL bResult;

   fAquireSemaphore();

      bResult = g_pCurrentDevice->SetZeroPositionHere();

   fReleaseSemaphore();

   return bResult;
}

// Function:    OnSetTrajectory
```

```
// Description: This function is called whenever the Set Trajectory button
//              is clicked.  It calculates the force vector for the specified
//                                   trajectory.
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL OnSetTrajectory()
{
        double          skinSurface = SKINSURFACE;
        BOOL            atFat;
        BOOL            punct;
        double          atFatDepth = 1000;
        double          depth;
        double          punctDepth;
        double          modelF;
        double          tissueDepth;
        BOOL            through;
        BOOL            inMuscle = false;
        BOOL            inISL = false;
        double          angle;
        double          l_fv[FORCE_RES];
        int             l_tv[FORCE_RES];
        int             firstSkin;

        SetDlgItemText( g_hwnd, IDC_TRAJSTAT, "Wait");

        // set up tissue vector for this trajectory.
    fAquireSemaphore();
        g_pCurrentDevice->GetDeviceInput( &g_DeviceInput );
        angle = g_DeviceInput.Axes[0] / 60;
    fReleaseSemaphore();

        // change angle to radians
        angle = angle * 3.14159 / 180;

        int midX = g_xmax / 2;
        int xi, zi;
        double d;
        double xcalc, zcalc, xindex, zindex;

        for (int h = 0; h < FORCE_RES; h++)
        {
                d = h;
                d = d/10; // d is in mm, h is in .1 mm
                xcalc = d * sin(angle);
                zcalc = d * cos(angle);
```

```
        xindex = (xcalc / g_xres) + midX;
        zindex = (zcalc / g_zres);

        xi = int(xindex);
        zi = int(zindex);

        if ((zi < g_zmax) && (xi < g_xmax))
                g_tissueVector[h] = g_mask[xi][zi];
        else
                g_tissueVector[h] = UNDEF;

        l_tv[h] = g_tissueVector[h];

        if ((g_tissueVector[h-1] == AIR) && (g_tissueVector[h] == SKIN))
                firstSkin = h;
}

// line up tissue vector with Jo's (the paper mache back model)
// skin which is 1cm in.
if (firstSkin > 100)
{
        int ind = 100;
        for (h = firstSkin; h < FORCE_RES; h++)
        {
                g_tissueVector[ind] = g_tissueVector[h];
                l_tv[ind] = g_tissueVector[h];
                ind++;
        }

        for (h = ind; h < FORCE_RES; h++)
        {
                g_tissueVector[h] = g_tissueVector[ind-1];
                l_tv[h] = g_tissueVector[h];
                ind++;
        }
}

atFat = false;
punct = false;

// set forceArray
for (int i=0; i<FORCE_RES; i++)
{
        depth = (float)i / 10;
```

```
// before hit skin surface
if (g_tissueVector[i] == AIR)
        g_forceVector[i] = 0.217;

// skin puncture
else if (g_tissueVector[i] == SKIN)
{
        while (!atFat)
        {
                // skin to puncture
                if (!punct)
                {
                        modelF = 0.0235 +
                0.0116*(depth - skinSurface) -
                0.0046*(depth - skinSurface)*(depth - skinSurface) +
                0.0025*(depth - skinSurface)*(depth - skinSurface)
                                        *(depth - skinSurface);
                        if (modelF > SKINMAX)
                        {
                                punct = true;
                                punctDepth = depth;
                        }
                }

                // skin release
                else
                {
                        modelF = SKINMAX -
                0.4516*(depth - punctDepth) -
                0.5287*(depth - punctDepth)*(depth - punctDepth);
                        if (modelF < FATFORCE)
                                atFat = true;
                }
                g_forceVector[i] = modelF;
                l_fv[i] = g_forceVector[i];
                i++;
                depth = (float)i / 10;
        }

        if (atFat)
                g_forceVector[i] = FATFORCE;
        l_fv[i] = g_forceVector[i];
}

// AT FAT
else if ((g_tissueVector[i] == FAT) ||
```

```
        (g_tissueVector[i] == MISCTISSUE))
         g_forceVector[i] = FATFORCE;

// muscle
else if (g_tissueVector[i] == MUSCLE)
{
        if (!inMuscle)
        {
                through = false;
                punct = false;
                tissueDepth = depth;

                while (!through)
                {
                        // facia to puncture
                        if (!punct)
                        {
                                modelF = 6.0266 +
                0.8287*(depth - tissueDepth) +
                0.1078*(depth - tissueDepth)*(depth - TissueDepth);
                                if (modelF > MUSCLEMAX)
                                {
                                        punct = true;
                                        punctDepth = depth;
                                }
                        }

                        // facial release
                        else
                        {
                                modelF = MUSCLEMAX -
                2.2543*(depth - punctDepth) +
                0.2902*(depth - punctDepth)*(depth - punctDepth);
                                if (modelF < MUSCLEFORCE)
                                        through = true;
                        }
                        modelF = modelF - PIGCORRECTION;
                        g_forceVector[i] = modelF;

                        l_fv[i] = g_forceVector[i];
                        i++;
                        depth = (float)i / 10;

                }
                inMuscle = true;
                i--;
```

```
                }
                else
                        g_forceVector[i] =
                        MUSCLEFORCE - PIGCORRECTION;

        }

        // interspinous ligament
        else if (g_tissueVector[i] == ISL)
        {
                if (!inISL)
                {
                        tissueDepth = depth;
                        while (!inISL)
                        {
                                modelF = 8.4592 +
                                        0.9598*(depth - tissueDepth);
                                if (modelF > ISLFORCE)
                                        inISL = true;
                                g_forceVector[i] = modelF - PIGCORRECTION;
                                i++;
                                depth = (float)i / 10;
                        }
                        i--;
                }
                else
                        g_forceVector[i] = ISLFORCE - PIGCORRECTION;
        }

        // bone
        else if (g_tissueVector[i] == BONE)
        {
                tissueDepth = depth;
                while (i < FORCE_RES)
                {
                        modelF = 8.0265  +
                                2.210*(depth - tissueDepth) +
                                1.4814*(depth - tissueDepth)*(depth -
tissueDepth)
                                - PIGCORRECTION;
                        g_forceVector[i] = modelF;
                        l_fv[i] = g_forceVector[i];
                        i++;
                        depth = (float)i / 10;
                }
                i--;
```

```
}

// ligamentum flavum puncture (need to correct equations by adding
// drag as the data was collected after dissecting to the LF layer.)

else if (g_tissueVector[i] == LF)
{
        through = false;
        punct = false;
        tissueDepth = depth;
        double drag = g_forceVector[i-1];
        while (!through)
        {
                // If to puncture
                if (!punct)
                {
                        modelF = - 0.0085 +
                1.5029*(depth - tissueDepth) -
                0.0583*(depth - tissueDepth)*(depth - tissueDepth);
                        if (modelF > LFMAX)
                        {
                                punct = true;
                                punctDepth = depth;
                        }
                        modelF = modelF + drag;
                }

                // If release
                else
                {
                        modelF = LFMAX -
                3.4693*(depth - punctDepth) +
                 0.7177*(depth - punctDepth)*(depth - punctDepth);
                        if (modelF < LFMIN)
                                through = true;
                        modelF = modelF + drag;
                }
                g_forceVector[i] = modelF;
                l_fv[i] = g_forceVector[i];
                i++;
                depth = (float)i / 10;

        }
        i--;
}
```

```
                // epidural space or spinal region
                else if (g_tissueVector[i] == EPID)
                        g_forceVector[i] = EPIDFORCE;

                // epidural space or spinal region
                else if (g_tissueVector[i] == SPINE)
                        g_forceVector[i] = SPINEFORCE;

                // past the masked data set then stretch out the last tissue
                else if (g_tissueVector[i] == UNDEF)
                        g_forceVector[i] = g_forceVector[i-1];

                l_fv[i] = g_forceVector[i];
        } // end set g_forceVector

        // add in noise
        srand(1);
        double noise;

        for (i = 0; i<FORCE_RES; i++)
        {
                noise = (((double)(rand()%10))/100) - 0.05;
                g_forceVector[i] = (g_forceVector[i] + noise);
                l_fv[i] = g_forceVector[i];
        }

        g_trajSet = true;
        SetDlgItemText( g_hwnd, IDC_TRAJSTAT, "Done");

   return g_trajSet;
}

// Function: averageZvelocity
// Calculate the average velocity in the along the z-axis
// average of the past 100 instantaneous velocities.

double averageZVelocity(int instVelocity)
        {
                static int elems = 0;
                static int current = 0;
                double removed;
                static int velArray[100];
                static double velSum = 0;

                if (elems < 99)
                {
```

```
                velArray[current] = instVelocity;
                current++; elems++;
                velSum = velSum + instVelocity;
        }
        else
        {
                if (current == 99)
                        current = 0;
                else
                        current++;
                removed = velArray[current];
                velArray[current] = instVelocity;
                velSum = velSum - removed + velArray[current];
        }
        return (velSum/elems);
}

// Function:    OutputSpringForce
// Description: this is the function we will pass to our SpringThread object,
//              it will be called as fast as possible by the CHapticThread
//              class.

VOID __stdcall OutputSpringForce(void *pv)
{
   INT ForceOut[3];
        static int spr = 0;
        double depth;
        int velocity;

   fAquireSemaphore();
        CDevice *pDevice = (CDevice*)(pv);
   pDevice->GetDeviceInput( &g_DeviceInput );
        fReleaseSemaphore();

                // set x and y forces to roughly center needle.
/*              if (spr < 0)
                        spr = spr+10;

                ForceOut[0] = spr;
                ForceOut[1] = -spr;
*/
                // this didn't work well.  So, created a fixture to keep FFD in a
           // single plane.  THerefore, set the x and y forces to 0.
                ForceOut[0] = 0;
                ForceOut[1] = 0;
```

```
// convert encoder value to mm (linear resolution = 0.02mm/count)
depth = g_DeviceInput.Axes[2] * -0.02;
velocity = averageZVelocity(g_DeviceInput.Velocity[2]);

if ((g_trajSet) && (depth < 5) && (g_tooFar))
        {
                MessageBox(NULL,"WET TAP!!!",
                        "Passed through epidural space",MB_OK);
                g_tooFar = false;
                g_hitSpace = false;
                g_trajSet = false;
}
else if ((g_hitSpace) && (depth < 5) && (g_hitSpace))
        {
                MessageBox(NULL,"SUCCESS!!!",
                        "Successful epidural needle insertion",MB_OK);
                g_tooFar = false;
                g_hitSpace = false;
                g_trajSet = false;
        }

else if ((g_trajSet) && (velocity < 100))
{
        // set needle insertion force
        // convert desired force in N to motor commands
        // motor command = N / DEVICE_RES
        // (DEVICE_RES = 0.003614)

        ForceOut[2] = ((g_forceVector[(int)(depth*10)])/DEVICE_RES)
                        +.2 * velocity;

        // threshold to max output force.  Should not be needed, but
        // device cannot handle forces bigger than 7.4N... which is
        // less than the force needed to puncture LF...

        ForceOut[2] = min(1700,  ForceOut[2]);
        ForceOut[2] = max(-1700, ForceOut[2]);

// look for endpoint
        if (g_tissueVector[(int)(depth*10)] == EPID)
                g_hitSpace = true;
        else if (g_tissueVector[(int)(depth*10)] == SPINE)
                g_tooFar = true;

} // if g_trajSet
```

```
                // otherwise, set to small force.  Should be zero but need to
                // compensate for the hardware trying to reach it's
                // equillibrium position.
                else
                        ForceOut[2] = 0.07228/DEVICE_RES;

                fAquireSemaphore();
                g_pCurrentDevice->OutputForces
                                (X_AXIS | Y_AXIS | Z_AXIS, ForceOut);
                fReleaseSemaphore();
}


// Function:    OnSpringEnable
// Description: This function is called whenever the Spring Enable checkbox
//              is clicked.
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL
OnSpringEnable()
{
   int i;

   if (SPRING_IS_ENABLED)
   {
     fAquireSemaphore();
      // if it's not active, let's kill the haptic thread.
      if (!g_SpringThread.IsRunning())
      {
         g_SpringThread.Start(1, OutputSpringForce, (void *)(g_pCurrentDevice));
         g_pCurrentDevice->EnableMotors();
      }
     fReleaseSemaphore();

   }
   else
   {
      // if it's not started, let's start the haptic thread.
      if (g_SpringThread.IsRunning())
      {
         g_SpringThread.Stop();
        fAquireSemaphore();
         g_pCurrentDevice->DisableMotors();
        fReleaseSemaphore();
      }
```

```
    }
     return TRUE;
}


// Function:    OnTimer
// Description: This function is called whenever the WM_TIMER message is received
// Returns:     BOOL- this will returned from the dialog function, indicating
//              whether it has processed the message.

BOOL OnTimer()
{
    fAquireSemaphore();
       g_pCurrentDevice->GetDeviceInput( &g_DeviceInput );
    fReleaseSemaphore();
    return TRUE;
}
```

APPENDIX C

C SOURCE CODE FOR THE ORANGE SIMULATOR

```
/* Epidserv.c:  Written by Leslie Hiemenz

        This program reads in the 4 virtual orange force curves.  The user
        can select which virtual orange is displayed.  The program reads
        the position of the needle shaft from the Impulse Engine, uses the
        position to index into the force array, and outputs the given force
        to the Impulse engine.

  Last updated: March, 1999.

  */

#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
#include <string.h>
#include <math.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "imm3000.h"


#define SHAFT_LENGTH  72.0  /* mm */
#define NUM_ORANGES 4
#define NUM_FORCES 200
#define HISTORY 5
#define DRAG 0
#define DELTA 0.3

double forces[NUM_ORANGES][NUM_FORCES];

void read_forces ()
{
  int i,j,k;
  FILE              *fp;
  float o1, o2, o3, o4;

  if ((fp = fopen("ORANGES.TXT", "r")) == NULL) {
     perror ("ORANGES.TXT");
     exit(1);
   }

  for (i=0; i<NUM_FORCES; i++)
```

```
    {
      fscanf(fp, "%f %f %f %f", &o1, &o2, &o3, &o4);
      forces[0][i] = o1;
      forces[1][i] = o2;
      forces[2][i] = o3;
      forces[3][i] = o4;
    }
  fclose(fp);
} /* read in force file */

void epidserv_die(int sig)   /* signal handler to shutdown gracefully */
{
  printf("Dying on signal %i\n", sig);
  haptic_off();
  close_haptic();
  printf("Haptic device(s) reset.\n\n");
  exit(1);
}

void run_trial(int orange)
{
  double insertion;
  int cont = 1;
  double previns[HISTORY];
  int curr_index = 4;
  int prev_index = 0;
  double force;
  int index;
  int i;
  double impulseForce;
  double findex;

  /* for the first cm the force is 0... allowing the previns variable
     to be initialized */
  insertion = read_probe();
  while (insertion <= 6.0 )
  {
        force_out(0, PROBE);
    insertion = read_probe();
  }

  /* then loop, if the direction vector is forward set the force according to
     the lookup table (linearly interpolated). If the direction is backwards,
     set the force to be the drag force. Exit when insertion < 5 mm. */

  while (cont)
```

```
  {
    insertion = read_probe();

    if ((insertion > 6.0) && (insertion <= 65))
        {
      findex = (insertion - 6) / 0.3;
                index = findex;
                force = forces[orange][index];
                // remove the interpolation because the real orange actually has
                // a kinda gritty feel...
                // force = ((forces[orange][index+1] - forces[orange][index])/(DELTA))*
                // (findex - index) + forces[orange][index];

      /* I calculate force in Newtons, need to change it to the
                        command the Impulse Engine wants... */
      impulseForce = -0.5 * (force / 0.0035779);
      force_out(impulseForce, PROBE);
                // printf("ins: %f  index %i force: %f impForce %f\n",
                //        insertion, index, force, impulseForce);
    }
    else
      cont = 0;
  } // while (cont)

  printf("Withdraw probe.\n");
  while (read_probe() > 0.05) ;

  printf("-------------------------------------------------\n");
}

main(int argc, char *argv[])
{
  int m;
  char buf[80];

  open_haptic();
  signal(SIGINT, epidserv_die);   /* shutdown properly after ctrl-c    */
  signal(SIGTERM, epidserv_die);  /* etc. so haptic/network stuff gets */
  signal(SIGQUIT, epidserv_die);  /* shutdown properly                 */
  signal(SIGHUP, epidserv_die);
  signal(SIGABRT, epidserv_die);
  haptic_off();
  printf("Calibrating, please wait.. ");
  fflush(stdout);
  zero_probe();
  printf("Probe is ready.\n");
```

```
  printf("Enter orange (1-4, or q to quit) \n");
  fflush(stdin);
  gets(buf);

  read_forces();

  while (buf[0] != 'q')
   {
    if (buf[0] == '1')
         {
              printf("Orange 1\n");
     run_trial(0);
    }
    else if (buf[0] == '2')
         {
              printf("Orange 2\n");
     run_trial(1);
         }
    else if (buf[0] == '3')
         {
              printf("Orange 3\n");
     run_trial(2);
         }
    else if (buf[0] == '4')
         {
              printf("Orange 4\n");
     run_trial(3);
         }

   force_out(0, PROBE);

   printf("Enter orange (1-4, or q to quit) \n");
   fflush(stdin);
   gets(buf);

   } /* while not quit */
  close_haptic();
  return(0);
}
```