Haptic Simulation of Naturally Occurring Textures and Soil Properties

By

Donald F. Green

B.A., Computer Science (1993)

Boston College

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 1998

Signature of Author...
Department of Mechanical Engineering
May 8, 1998

Certified by...
J. Kenneth Salisbury, Jr.
Principal Research Scientist
Thesis Supervisor

Accepted by..
Ain. A. Sonin
Chairman, Department Committee on Graduate Students

Haptic Simulation of Naturally Occurring Textures and Soil Properties

By

Donald F. Green

Submitted to the Department of Mechanical Engineering on May 8, 1998, in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering.

Abstract

Methods for developing realistic haptic (force feedback) simulations of soils and rocks are presented. Mathematical models of the dynamics of a virtual probe mechanically interacting with a virtual object are developed to provide the basis for analysis and simulation. The models then incorporate stochastic inputs in order to provide the haptic simulations with a more natural, less synthetic feel. The stochastic input parameters are derived by analyzing actual force data sensed while probing a subject media with the haptic display itself; in this case used as a force controlled manipulator.

A method for sensing friction properties of rigid, textured surfaces and using the data collected to drive a realistic haptic texture simulation is presented. Static friction coefficients and surface height deviations are sensed by directly stroking the surface under examination with a probe fitted on the end of a PHANToM haptic display device. Test surfaces range from pieces of sandpaper of varying coarseness to acetate. A simulation of the texture may then be rendered using the mechanical model of textured surface-probe interaction augmented by statistical variation of the friction properties of the surface.

An algorithm is presented for adding texture properties to three-dimensional object models. The method is based on determining surface normals of the virtual object and assigning statistically varying friction properties and surface height deviations to area patches on the object's surface using the methods described above.

Finally, a dynamic model of probe/soil interaction is used to render a haptic simulation of loose grained soils such as sand. Certain friction properties are again statistically varied in order to improve the realism of haptic display user's experience.

Thesis Supervisor: J. Kenneth Salisbury, Jr.
Title: Principal Research Scientist

Acknowledgments

There have been many people influential to my development as a person and as an engineer and while it is impossible to name everyone to whom I feel a debt of gratitude I'm going to take a run at acknowledging those most relevant to the work presented here.

To my wife Patrice, to whom this document is dedicated, I owe it all. I would still be scraping paint off windowsills if it were not for your love and confidence in me. I promise, someday I will graduate for the last time.

To my mentor Dr. Jerry Burchfiel, who has selflessly and tirelessly guided me for the last eight years, from basic algebra to group theory. Thank you Jerry, due to your friendship and tutelage I am an engineer.

To my advisor Dr. J. Kenneth Salisbury, who took me on as an advisee and was a patient and helpful guide. The difference between a merely grueling or truly miserable educational experience at this place is one's advisor. I am convinced you are the best, thank you Ken.

To Dr. James Gips, who fulfilled the role of the influential undergraduate professor that encourages and motivates. Thank you Jim, your genuine interest and continued friendship have helped me gain both confidence and perspective.

To my research group peers, Brian Anthony, Ela Ben-Ur, Jesse Hong, Arrin Katz, Craig Latimer, Dr. Akhil Madhani, Mark Ottensmeyer, and Daniel Theobald. Thank you guys, your friendship has made coming to the lab a pleasure.

To Dr. Richard Weissman and Leslie Regan the Mechanical Engineering Department Secretary. Among other things I have to thank Rich for, it was he who contacted Leslie about my coming to the Mech. E. department and Leslie who steered me to ask Ken about becoming an advisee. I have remained consistently clueless about administrative affairs here at the Tech. and Leslie has been like a fairy godmother, time and again smoothing out my path. Thank you Rich and thank you Leslie.

Finally, to my mother. Your unquestioned love has provided me with a bedrock of emotional strength.

The research reported in this thesis was supported in part by the John. A. Lyons Fellowship and in part by NASA/JPL Contract Number 959774.

List of Figures

Chapter 1 Introduction

This thesis presents an approach to virtual touch simulations of rigid textured surfaces by directly sensing mechanical properties such as static friction coefficients and high spatial frequency and low amplitude surface height deviations. The simulations and data acquisition is performed using a haptic display device developed by Massie and Salisbury called the PHANToM [Massie 93]. The PHANToM is essentially a three-degree of freedom robot manipulator with high-resolution position sensing and force control capabilities. It can exert a three-dimensional force vector at the end-effector with a high degree of accuracy. For a more detailed description of the characteristics of the PHANToM device the reader is referred to [Massie 93][Massie and Salisbury] and the World Wide Web site of Sensable Devices, Inc. (http://www.sensable.com).

The mechanical properties of the PHANToM make it an excellent platform to begin research on remote sensing and simulation of texture properties because of its high system stiffness capability, low intrinsic friction characteristics, and high spatial position sensing resolution. These factors allow the PHANToM to be accurately position controlled while also permitting reasonably accurate feed-forward force control and even force sensing, as we shall see.

1.1 Motivation and Approach

One of the primary motivations for my exploration of haptic texture simulation is to provide planetary scientists, geologists, and civil engineers with a useful tool for evaluating soil properties. Haptic displays such as the PHANToM are capable of providing a compelling presentation of remotely sensed soil property data. A haptic presentation transports the operator from the abstraction of numerical representations to an intuitive physical representation. This physical form of information conveyance can present extremely complicated data sets with an efficiency that is otherwise impossible.

In order to be scientifically useful, haptic simulations of soils and rocks must be based upon accurate and accepted mechanical models of soils and surface properties.

Throughout my research into texture and soil simulation I have taken an approach that begins with a detailed mathematical model of a probe's mechanical interaction with a given media. From this model real time force and position control systems are derived to drive the soil and texture simulations.

The useful degree of detail in these models is constrained by the electro-mechanical limitations of the PHANToM display system and computational limitations of the processors that control it. I have included analysis of these factors in the discussions that follow.

Another motivation for improving haptic texture rendering techniques is to enhance the realism of haptic simulations in general. Just as the addition of visual texturing greatly enhanced the 'realism' of computer generated graphics, improvements in haptic texturing will have a similar impact. However, in order to be practically useful and to achieve widespread adoption a technique is required that is readily adaptable to existing modeling algorithms commonly used for rending complicated three-dimensional force feedback displays. This algorithm design requirement was also an important factor in guiding the approaches to texturing that I explored.

1.2 Previous Work

The study of textured surfaces in computer imaging applications is a more mature area of research than its haptic counter-part. As such, it provides a natural source of ideas and research results relating to texture properties in general, as well as the sensing and simulation of textures in computer graphics applications. The simplest primitive element constituting a texture in the image domain is the color, or more simply, the gray tone of a given pixel element. Analogous primitive elements of a texture in the haptic domain (three dimensional inertial space) can be the surface height Z for a point (x,y), a friction coefficient μ for a point in a plane (x,y), or a surface normal perturbation **N'** for a point on a surface (x,y,z) with a surface normal **N**. Due to these parallels many of the analysis and rendering techniques studied in image texture research have application in the haptic domain.

A. Ravishankar Rao's doctoral thesis "A Taxonomy of Textures" [Rao] provides a useful image texture classification system that is also helpful for describing three-dimensional textures. His classifications fall into three basic categories, strongly ordered, semi-ordered, and disordered textures. Figure 1.1 shows representative examples of each type of texture.

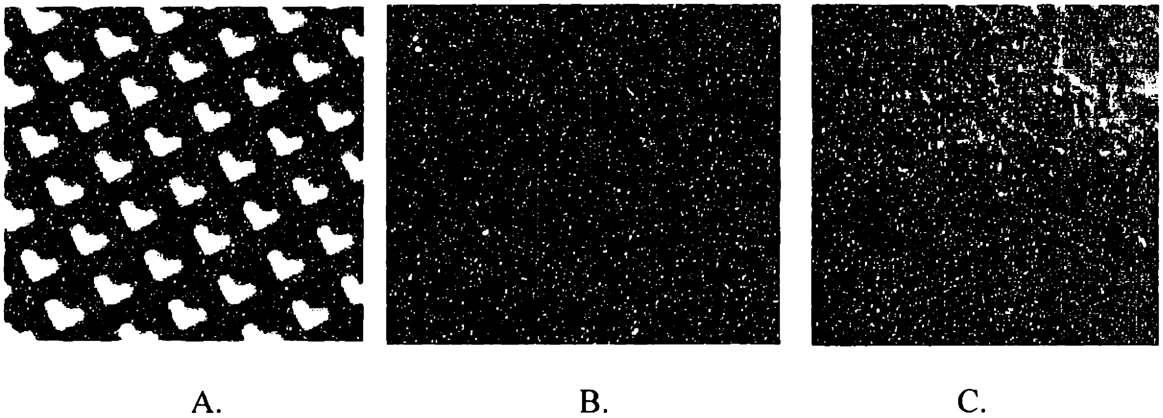


Figure 1.1 Basic Texture Classes A.) Strongly Ordered B.) Semi-Ordered C.)Disordered

The characteristics distinguishing strongly ordered textures are global pattern regularity and orientational dependence. These types of textures have been analyzed and quantified using spectral, auto-correlation, and co-occurance matrice analysis methods among others to varying degrees of success [Rao p.127][Chen C.H].

Semi-ordered textures exhibit localized pattern regularity and orientation. Researchers have used localized spectral methods for analysis such as correlating subsets of the texture image with the gradient of two-dimensional Gaussian filters [Rao p.18][Kass]. The flow like patterns exhibited by semi-ordered textures suggest direction fields and integral curves of differential equations in the phase plane. It would be interesting to explore describing semi-ordered textures as the graphical description of differential equations and discovering methods for backing out a differential equation or set of equations describing the texture from the image.

Disordered textures do not have the repetitive, regular patterns and directional dependence of the other texture primitives and the most successful approaches to analyzing and describing these image textures have been statistical. Early work in automated disordered texture analysis focused on using co-occurance matrices to define and compute certain features of image textures such as correlation, contrast, mean value, variance, and entropy [Haralick 73]. Other research has been done using spectral methods for texture identification. For example, N. Gramanopoulos used Fourier analysis of satellite imagery for automated terrain identification [Gramanopoulos]. A good survey of techniques used for studying primarily disordered textures can be found in [Haralick 79]. Another interesting technique introduced by A. Pentland discusses using the fractal dimension of an image texture as a measure of roughness [Pentland].

I began my research into haptic texture rendering with Margaret Minsky's doctoral thesis and seminal work on haptic texture synthesis ``Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force Feedback Display." In this work the author presents a method of simulating three-dimensional textures using only two actuated degrees of freedom capable of exerting only lateral forces [Minsky p.48-51]. This work demonstrated how actively controlled lateral forces were capable of generating convincing simulations of various surface textures. Minsky used an algorithm that generated lateral forces proportional to the local gradient of a textured surface.

Siira and Pai present an algorithm for statistical representation and simulation of surface textures using a two degree of freedom haptic display [Siira & Pai]. Their work utilized the fact that many 'real-world' surfaces have Gaussian height distributions when measured from a reference height. They demonstrate a texture synthesis method that generates lateral forces opposed to the direction of motion and proportional to the changing surface height with encouraging results. However, no attempt was made to actually measure real surfaces and then simulate them.

Fritz and Barner present a texture rendering method involving pre-computed arrays of texture vectors used to perturb a virtual object's surface normals [Fritz]. This technique maps textures onto surfaces and allows filtering of the textures to achieve desired texture properties. Their work was motivated by a desire to provide a systematic way to synthesize different feeling textures without regard to simulating real world textures. They also use statistical methods to populate their 'texture lattices' and then perform various filtering operations to achieve desired texture effects.

Chen and Taylor present a haptic texturing algorithm based on a stick-slip friction model [Chen and Taylor]. Their algorithm computes lateral forces to simulate texture and friction effects based on a model of the probe-surface interaction that includes a flexible probe and indentations uniformly distributed over a virtual surface. As the probe catches in an indentation on the surface lateral force increases, opposing the direction of motion and proportional to the compliance of the probe model. When this lateral force is great enough the probe slips and moves to a new position on the surface until it hits the next snagging point. The implementation is a procedural approach because catch points are computed at run time rather than mapping out a surface before hand.

1.3 Haptic Display System Characteristics

The haptic display system used to conduct this research was a 200Mhz Pentium Pro machine running Microsoft's Windows NT 4.0 operating system and a Sensable Devices, Inc. model "A" PHANToM system. Controller programming was done using the low-level io-libraries provided by Sensable Devices. Typical control loop update rates achieved during the sensing and simulation process were around 7.5 to 8 Khz.

Different end-effectors were fashioned and substituted for the usual thimble end-effector that comes with the PHANToM device. Figure 1.2 shows a picture of the different end-effectors used while conducting the research for this thesis.

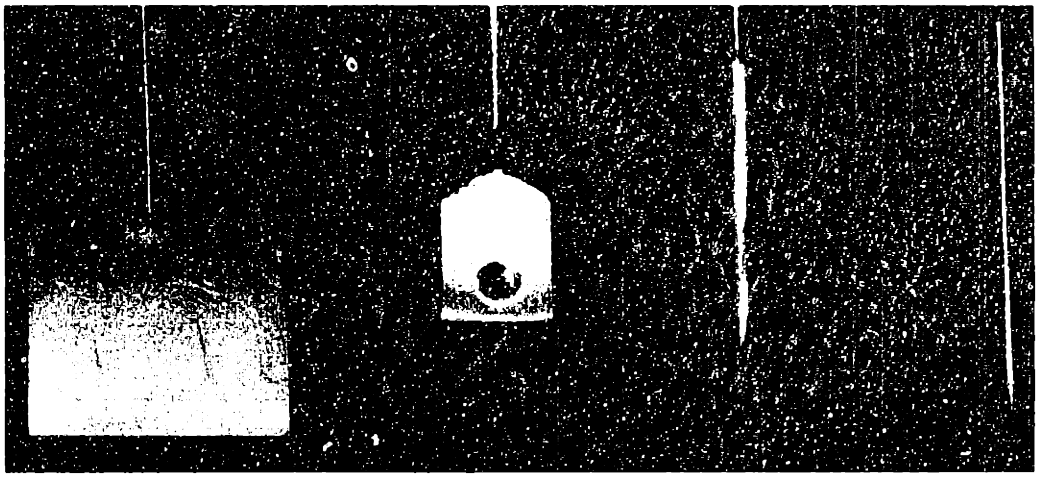


Figure 1.2 End-Effectors Used

Proceeding from left to right these end-effectors are: a plow-blade for sampling forces generated while moving through sand, a threaded connector to secure the PHANToM to a single axis load cell mounted at a 90 degree angle to measure the X and Y axis response, a threaded connector to securely join the PHANToM to the load cell mounted horizontally in order to measure the Z axis, and finally a steel stylus used for the texture sampling process. The plow blade is 28 mm wide by 24mm high. The stylus is about 5 cm long and is 3 mm in diameter.

1.4 System Identification

An effort is made to characterize the haptic system using stochastic system identification techniques. The principle behind this system identification approach is to spread out an impulse function's power input in time, reducing the peak to average signal power in order to avoid exciting nonlinear responses in the system to be tested. The magnitude of the Fourier transform of an impulse function is flat line across the frequency band, indicating that the input signal provides an equal amount of power across the frequency band. The inverse Fourier transform of a frequency domain signal with constant magnitude but uniformly distributed pseudo-random phases will result in a pseudo-random time domain signal with a Gaussian probability density function. In effect, we are spreading the power of an impulse out in time. Figure 1.3 shows the Fourier transform of the input signal used in the system identification process presented in this chapter and Figure 1.4 shows the time domain plot.

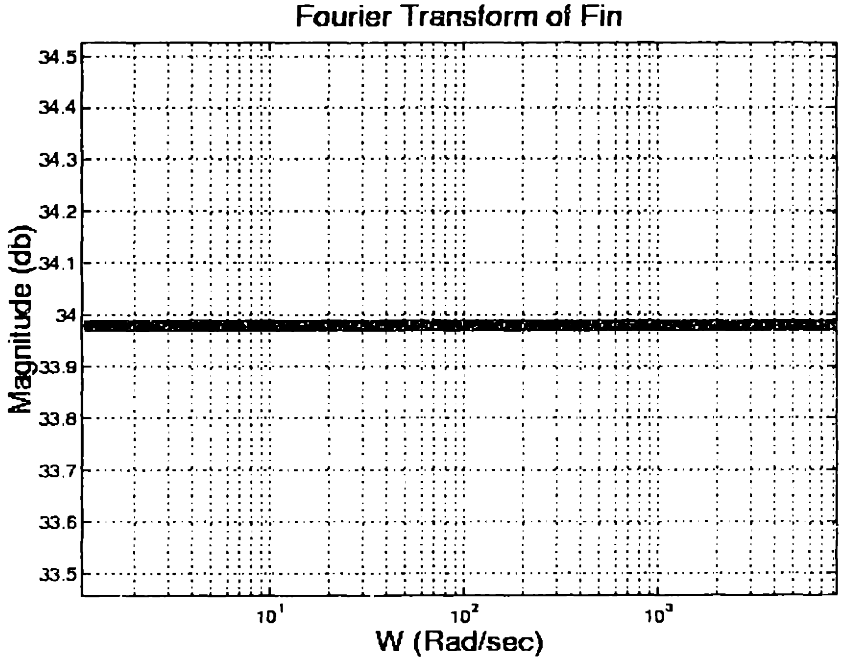


Figure 1.3 Input Signal Fourier Transform Magnitude

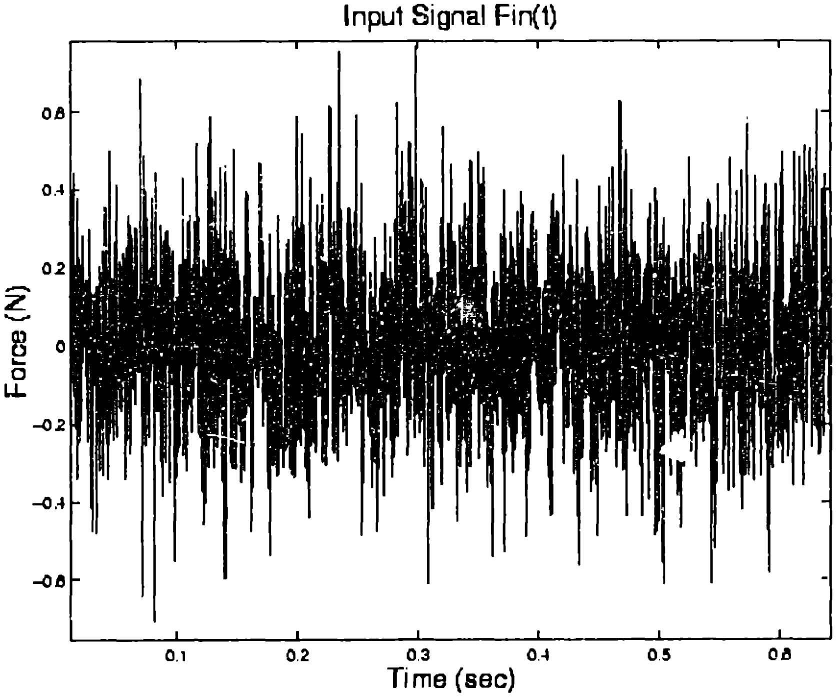


Figure 1.4 Input Signal

These figures illustrate how a pseudo-random input can have the same Fourier transform behavior as an impulse function, that is a signal with equal magnitude across the frequency band. It can therefore be used to compute the transfer function of a system in the same manner as an impulse response transform, by dividing the output signal Fourier transform to produce the familiar equation Output(ω)/Input(ω) = transfer function.

The input signal is constructed in the Fourier domain in order to ensure that it has certain special properties. First, we desire a signal with the aforementioned constant magnitude and pseudo-random phase. Second, steps must be taken to construct the signal so that its inverse Fourier transform will result in a time domain signal with all real valued coefficients. Even functions have this property by virtue of their symmetry, that is a function *f* is said to be even if *f(-x) = f(x)*. In the frequency domain this translates into a function $F(-\omega) = F^{*}(\omega)$, where F^{*} is the complex conjugate of *F*. An excellent reference on this material can be found in [Strang]. Therefore, the input function is constructed to be an even function. Equation 3.1 illustrates the computations involved for creating an input vector f in the time domain with 32768 (2^{15}) real valued samples, pseudo-random phase, and constant magnitude.

```
for $n \in [0,1\ldots,2^{14}-1]$
$\theta[n] = \text{uniform\_random}() \bullet 2\pi$
$F[n] = \cos(\theta[n]) + i \bullet \sin(\theta[n])$

$F[2^{14}] = F[0]$

for $n \in [1,2,\ldots,2^{14}-1]$
$F[2^{14}+n] = F^{*}[2^{14}-n]$

$f = \text{IFFT}(F)$
```

Equation 1.1 Algorithm for Computing the Input Signal for System Identification

The computations in equation 3.1 show how to create a symmetric, periodic function with constant magnitude and pseudo-random phase. First, one half of the signal is computed as shown for the first 2^{14} elements. Then, that first half of the signal is mirrored into the second half, assigning the $(2^{14}+n)^{th}$ value of F the complex conjugate of the $(2^{14}-n)^{th}$ value of F. The pseudo random number generator returns uniformly distributed pseudo random numbers between 0 and 1. The input signal F is then used to drive the PHANToM, the output Y is measured, and the system transfer function is computed as shown in Equation 1.2

$$
\begin{aligned}
&F(\omega) \equiv \text{Input Signal Fourier Transform} \\
&F^{*}(\omega) \equiv \text{conjugate}(F(\omega)) \\
&Y(\omega) \equiv \text{Output Signal Fourier Transform} \\
&H(\omega) \equiv \text{Transfer Function} \\
&Y(\omega) = H(\omega)*F(\omega) \Rightarrow Y(\omega)*F^{*}(\omega) = H(\omega)*F(\omega)*F^{*}(\omega) \Rightarrow \\
&\frac{Y(\omega)*F^{*}(\omega)}{F(\omega)*F^{*}(\omega)} = H(\omega)
\end{aligned}
$$

Equation 1.2 Computing the system transfer function

Each of the Cartesian space axes of PHANToM was tested on the assumption that the force inputs will not significantly excite dynamics in the other principle directions. While it is unlikely that the Cartesian space defined at run time for the PHANToM when 'homing' the device exactly corresponds to the principle dynamic axes of the system, I have assumed that it is close enough to give reasonable and useful results about the dynamics of the PHANToM system. This is a reasonable assumption because the PHANToM was intentionally designed to have diagonal transformation matrices between the rotation of the motors and their translational effects at the endpoint [Massie 96 p.10]. Figure 1.5 shows a photograph of the sampling setup. I have defined the coordinated axes as indicated in figure 1.5 and use this coordinate scheme throughout the rest of the thesis. The principle directions correspond to those defined by the PHANToM low level software drivers and I have chosen the labeling scheme for these axes as indicated. When the PHANToM is in its 'homed' position, as shown in figure 1.5, each single motor effects a force in only one of the principle directions. This special configuration allows a de-coupled examination of the frequency response of the device in each of the principle directions

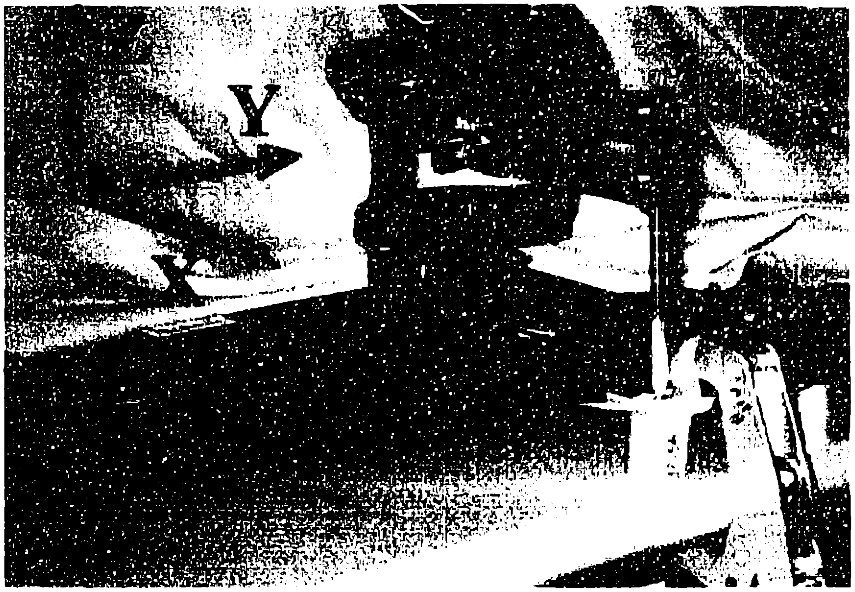


Figure 1.5 System Identification Setup

Figure 1.5 shows the end-effector second from the right of figure 1.2 attached to a load cell and the end link of the PHANToM arm. The end-effector is threaded onto the load cell. The load cell is threaded into a flat piece of aluminum, which is then clamped securely to the tabletop.

The experiments were conducted using the above-mentioned computer and operating system equipped with a National Instruments Lab PC 1200+ with 12 bit A/D resolution. An Entran Devices, Inc. 50 Newton rated tension/compression single axis load cell was used along with a Vishay Intertechnology, Inc. model 2311 signal conditioning amplifier. The amplifier low-pass filtering facilities were used and set with a corner frequency of 6141.6 Radians per sec (1000 Hz).

The computed transfer function is that of the output force from the PHANToM as seen by the load cell over the commanded input force from the *software*. Thus the transfer function gain will include dynamics of the entire PHANToM system including those of the amplifier box used to drive the haptic device. I chose to do this rather than measuring input current to the PHANToM motors in order to provide data that included dynamics of the system as used by most haptics researchers using the PHANToM systems.

Figures 1.6 through 1.8 show normalized histograms of the input and output forces sampled during the experiments for each axis. The distributions show that system exhibits linear behavior, justifying the assumption of linearity and the approach to system identification outlined above.

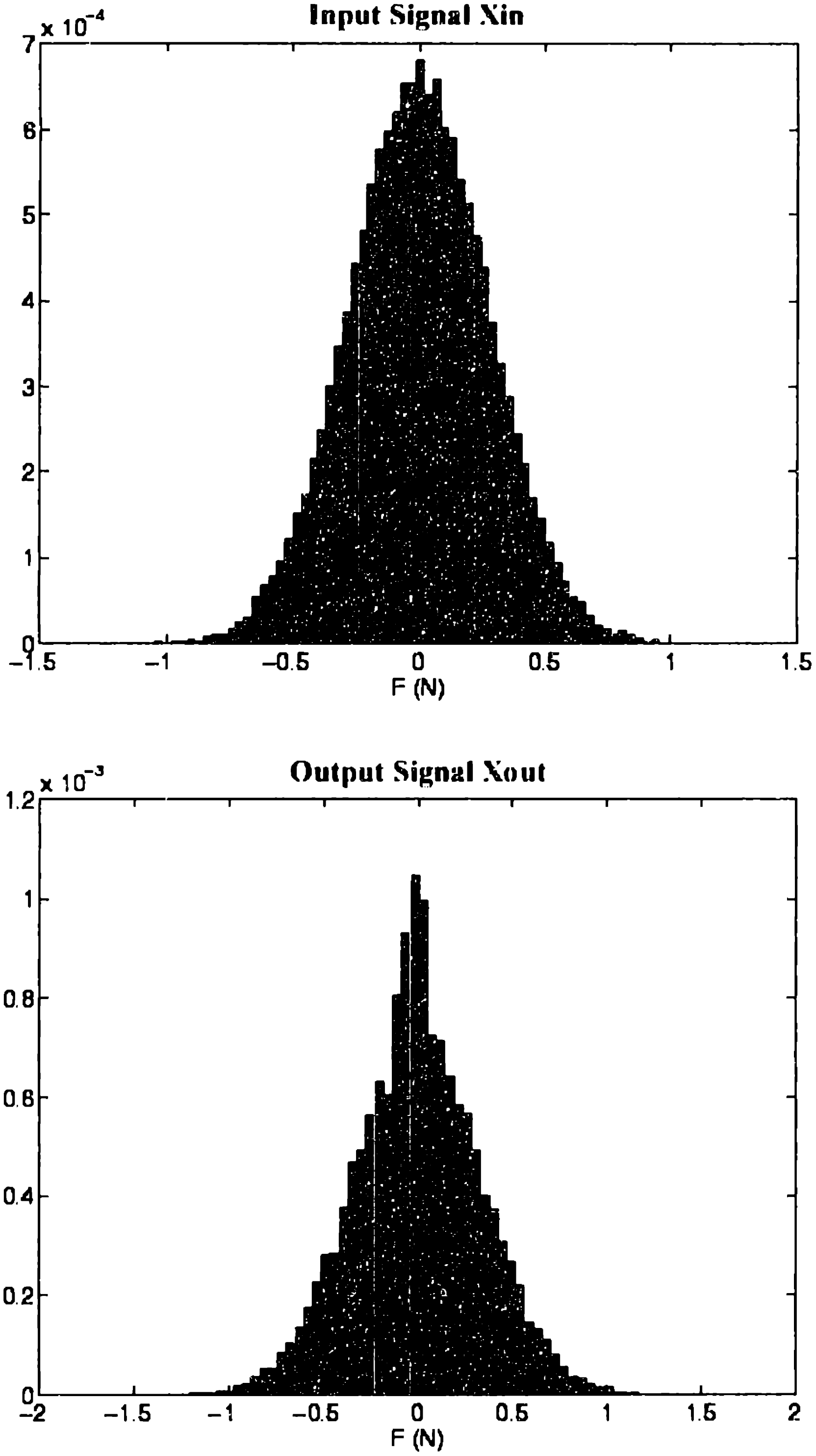


Figure 1.6 X Axis Input and Output Force Signal Histograms

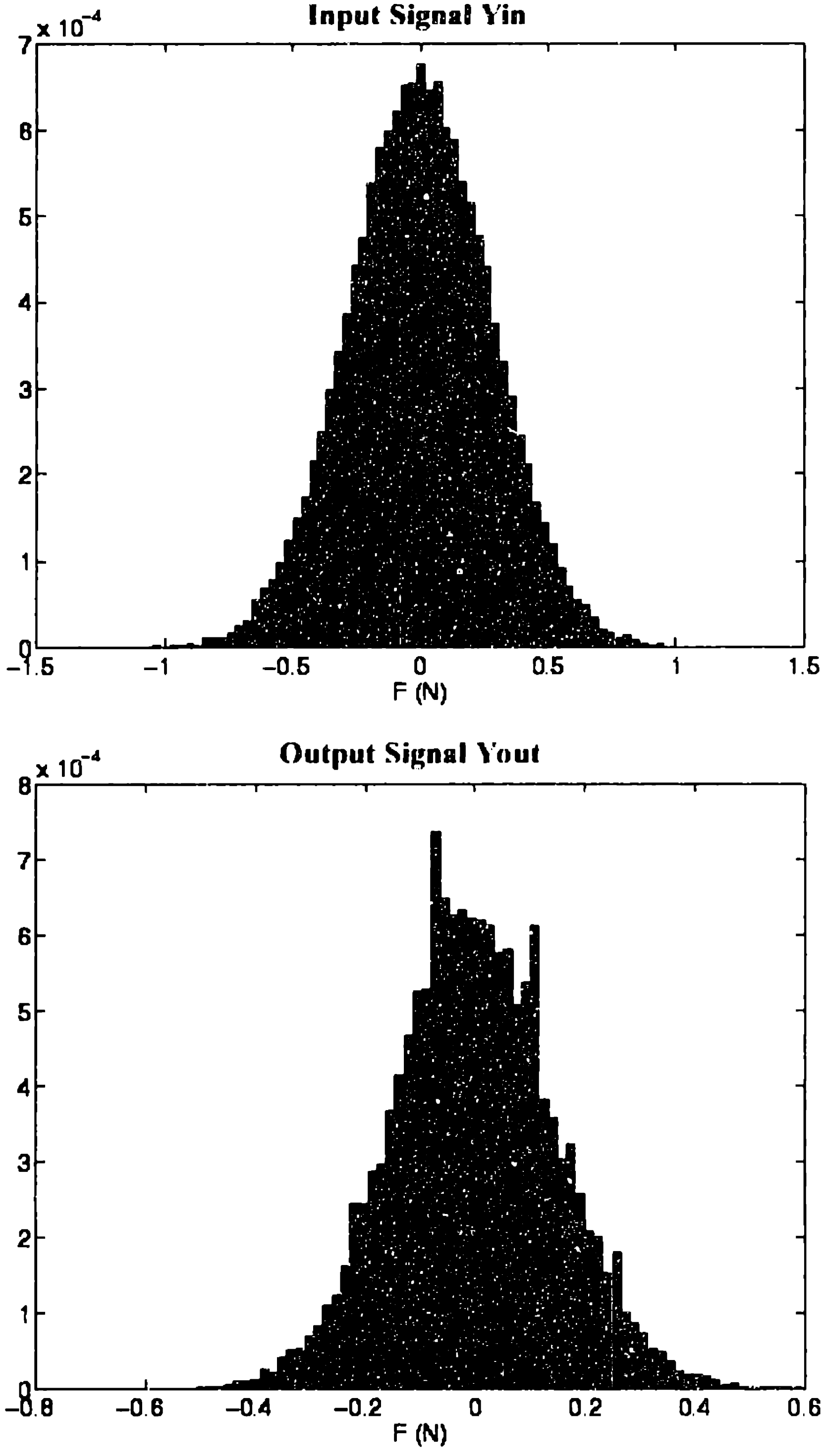


Figure 1.7 Y Axis Input and Output Force Signal Histograms

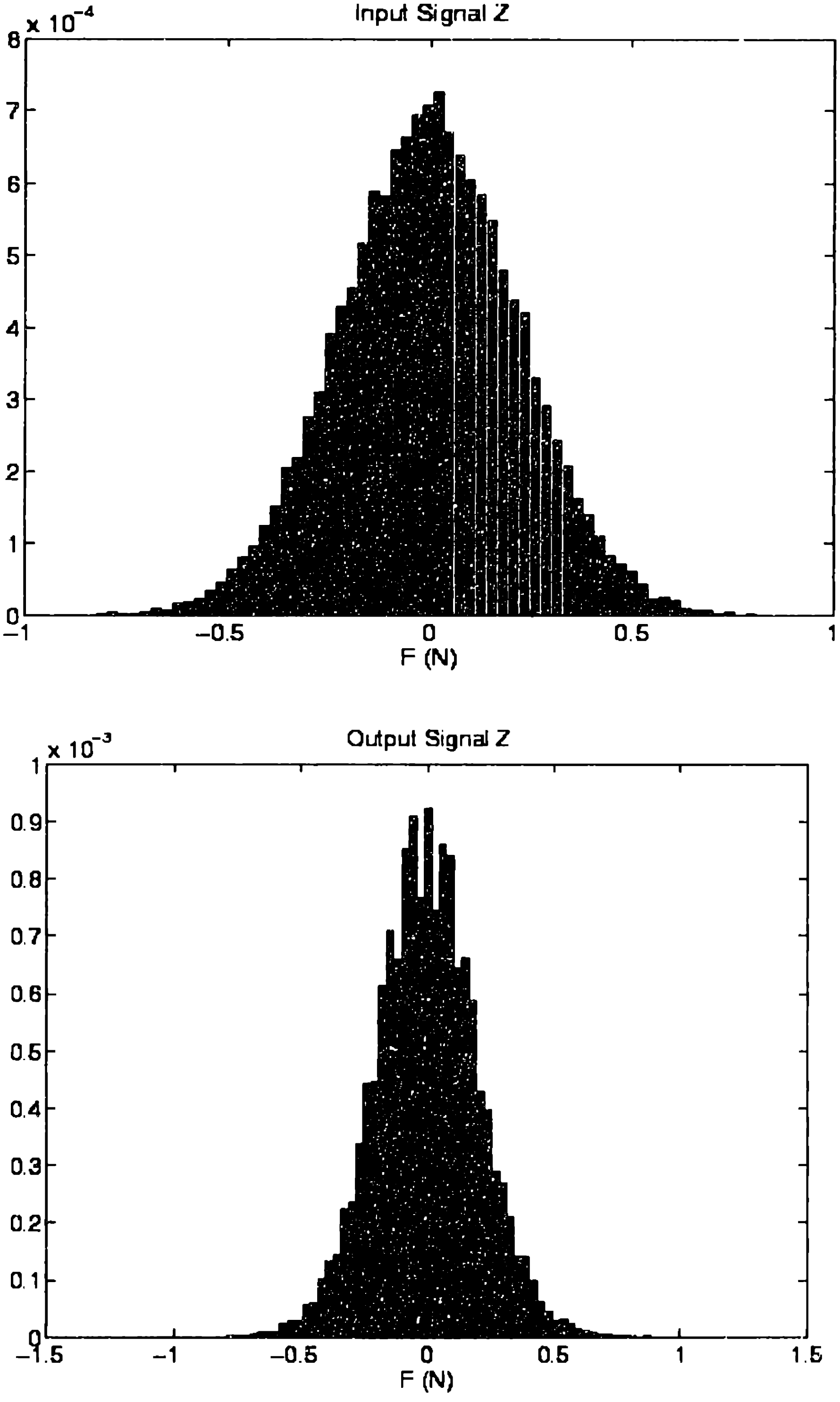


Figure 1.8 Z Axis Input and Output Force Signal Histograms

Figures 1.9 through 1.11 show the computed transfer function magnitudes for each of the principle axes of the PHANToM system. Overall, the computed transfer function magnitudes clearly indicate that the PHANToM exhibits a nice flat response out to a point depending on the axis examined but in the range of 100 to 200 Rad/sec (16 to 32 Hz). The system then shows classical 2^{nd} order under-damped behavior, with natural frequencies at about 400 Rad/sec in Y, 600 Rad/sec in X, and 700 Rad/sec in Z (64, 95, 111 Hz). Higher frequencies seem to indicate a second resonance but the data acquisition rate and resolution precludes commenting on it with any certainty. The most clearly represented 2^{nd} resonance appears in figure 1.10, the magnitude of the Y axis response.

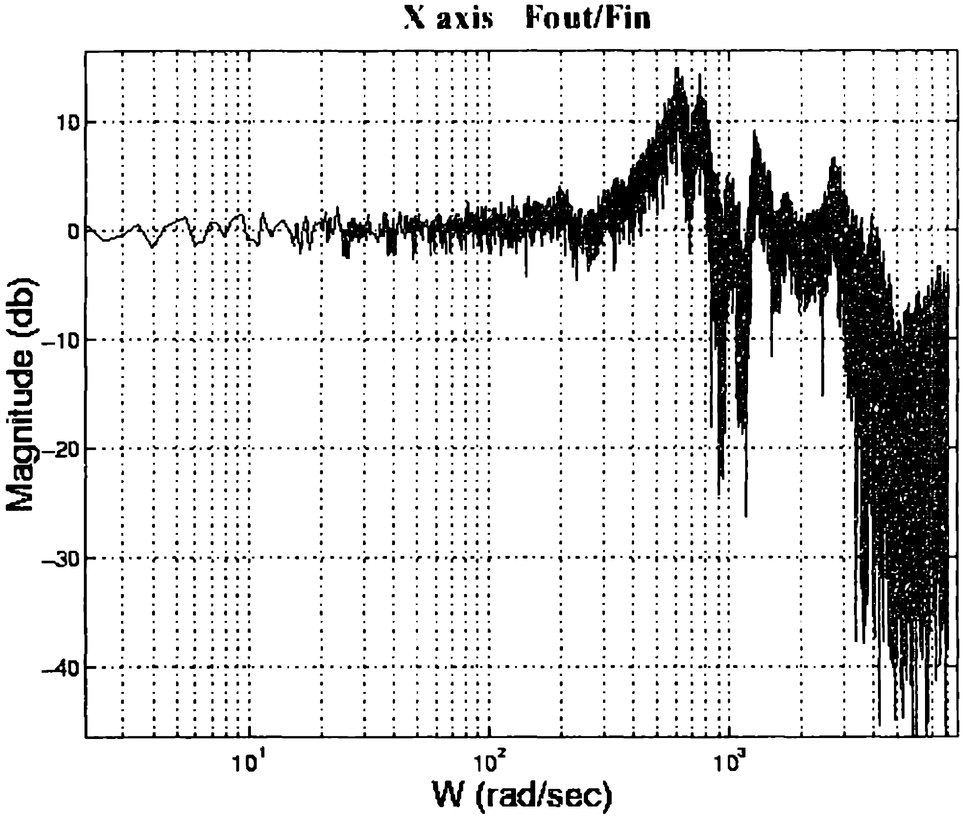


Figure 1.10 X Axis Transfer Function Gain

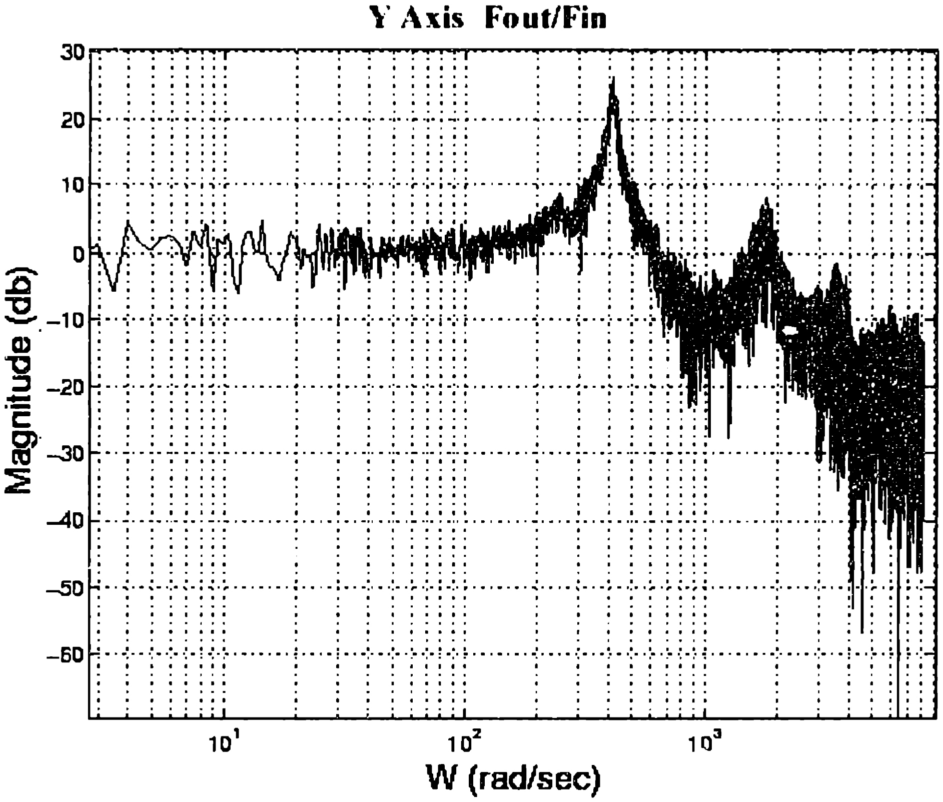


Figure 1.11 Y Axis Transfer Function Gain

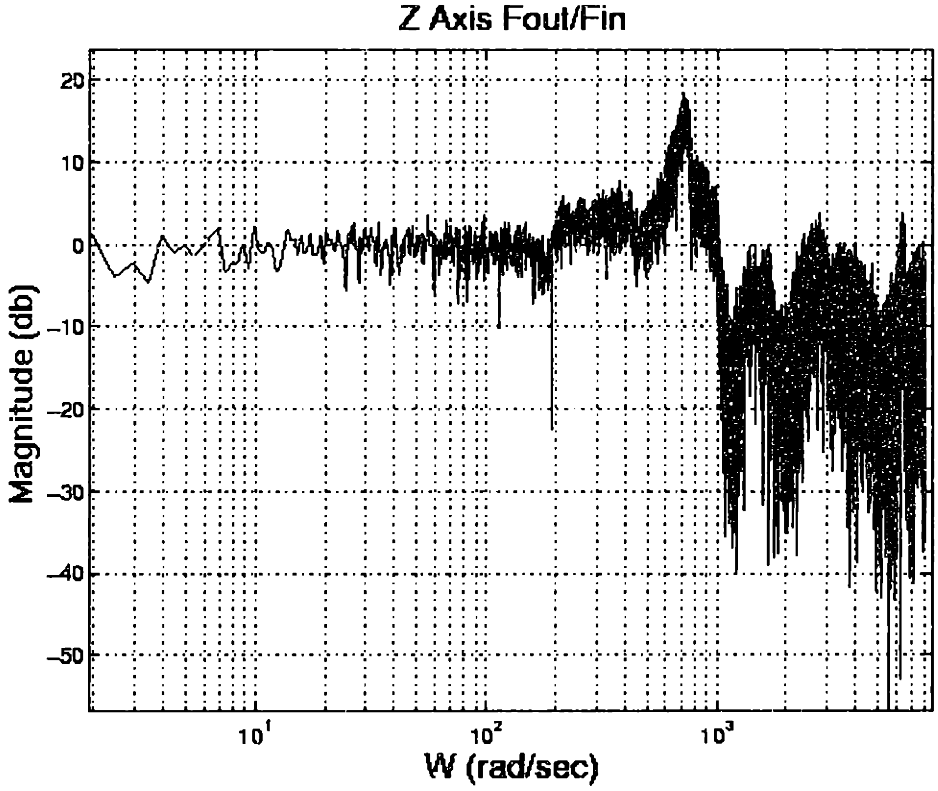


Figure 1.12 Z Axis Transfer Function Gain

The transfer function gains indicate a 4th order system response overall. The clearest representation is in figure 1.11, which plots the Y-axis gain. Figure 1.13 illustrates a possible lumped parameter model of the system.

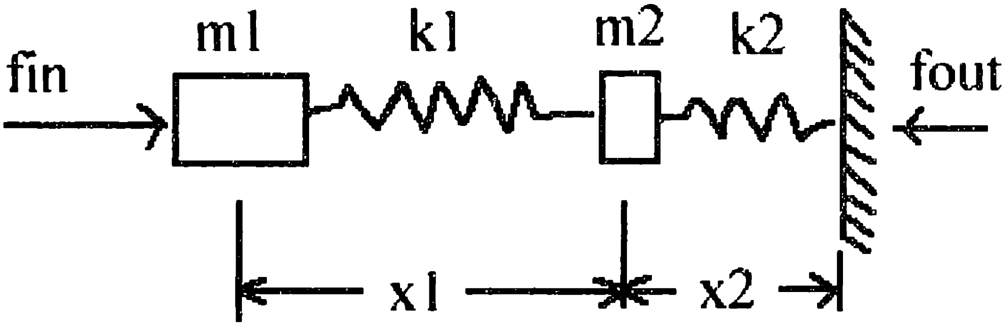


Figure 1.13 Lumped Parameter Model of PHANToM System

The lumped parameter model consists of a pair of mass elements and a pair of spring elements. The mass m1 represents the inertia of the PHANToM structure reflected to the user at the end point. On their web site, Sensable Devices estimates this value to be 75 grams or less. The spring k1 represents the stiffness of the PHANToM structure between the motor and the end point. This value was measured to be 3065 newtons per meter. The spring k2 represents the stiffness of the load cell itself. It is estimated to be ten times that of the PHANToM structure. The mass m2 represents the mass seen by the load cell, in this case the mass of the end-effector used and it was found to be 7.5 grams. The transfer function of this un-damped lumped parameter model has the form shown in equation 1.3.

$$H(s)=\frac{1}{\left[\left(\frac{m1\cdot m2}{k1\cdot k2}\right)\cdot s^4+\left(\frac{m1}{k1}+\frac{m2}{k2}+\frac{m1}{k2}\right)\cdot s^2+1\right]}$$

Equation 1.3 Transfer Function of Lumped Parameter Model in Figure 1.13

Figure 1.14 shows the frequency response for the transfer function of the lumped parameter model. The model results in a good match to the gain plot computed from the recorded data plotted in figure 1.11. The mass m1 was adjusted from a max value of 75 grams to 15 grams in order to achieve the response shown in figure 1.14. The maximum value of 75 grams stated on Sensable Devices' web site resulted in a lower resonant frequency for the first peak than is observed in figure 1.11. The second resonant peak matched up well with the recorded data without adjusting any parameters.

The X and Z axis gain plots also show similar system behavior but the plots suffer more from noise in the measurements. Qualitatively speaking both the X and Z axes should exhibit more structural stiffness because of the orientation of the device links with respect to the direction of applied force and this expectation is born out in the data. Future work will be directed towards expanding on the preliminary analysis presented here with the intent of publishing the results for the benefit of other researchers working with the PHANToM device.

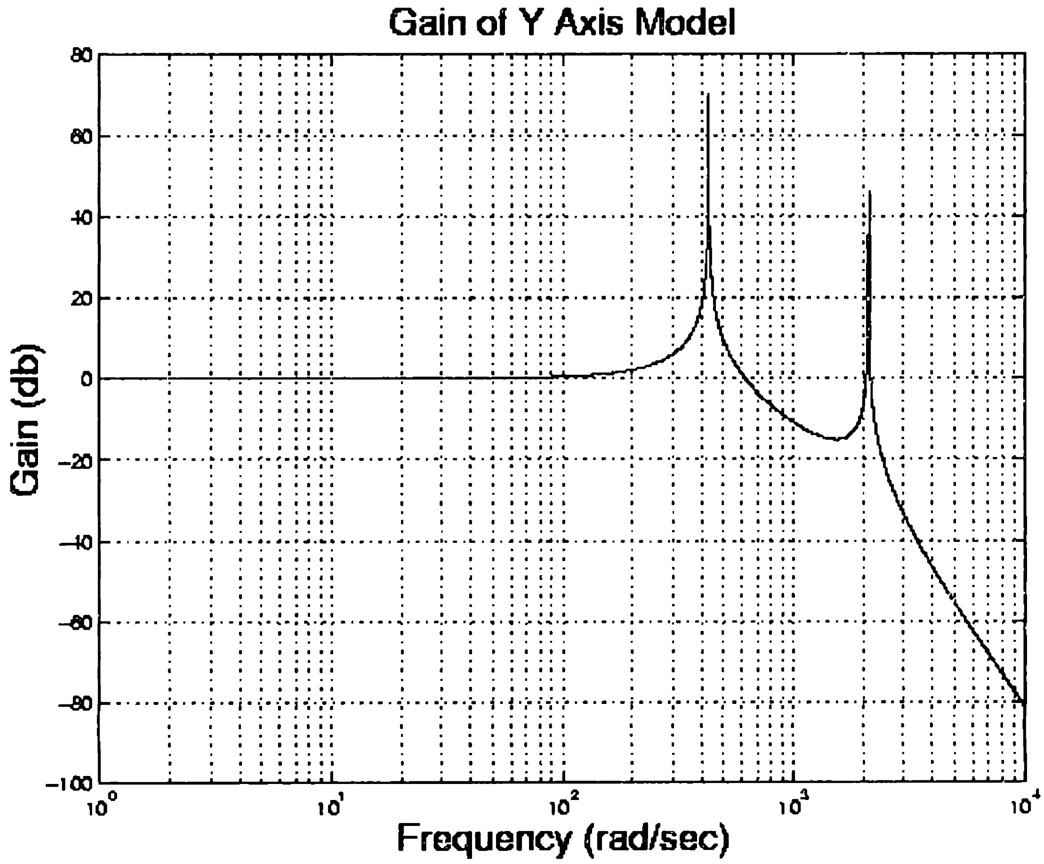


Figure 1.14 Gain of System Model for Y Axis

Chapter 2 Rigid Texture Sensing & Simulation

This chapter presents methods developed for mechanically sensing textures from the surfaces of rigid objects using the PHANToM haptic display device. Algorithms for interpreting the data gathered from the sensing process in order to drive a haptic simulation of the texture are also shown. The material in this chapter is oriented towards sensing and accurately simulating rigid and disordered textures (see chapter one, section two for review of texture types) as found on the surfaces of rocks.

The sensing process involves commanding the PHANToM to stroke the textured surface under hybrid force/position control. While the PHANToM moves across the textured surface the commanded lateral forces are recorded as well as the changing vertical position of the stylus tip as recorded by the encoders on the PHANToM actuators. The hybrid control algorithm commands the PHANToM to follow a certain position trajectory in the horizontal plane under proportional or proportional-derivative control while exerting a constant force normal to the surface. The recorded forces are then divided by the commanded normal force and the resulting values are interpreted as a vector of static friction coefficients according to the well known equation for static friction force $F_{friction} = \mu F_{normal}$.

The resulting vector of friction coefficients (and height values) were found to have Gaussian probability distributions. The mean and standard deviation of the computed friction coefficients and the standard deviation of the recorded height values provide a compact statistical representation of the texture which can be used to drive a realistic feeling, procedurally based haptic simulation of the surface. The simulation of the textured surface is based on a stick-slip contact model of a complaint probe stroking a textured surface.

2.1 Physical Model

The mechanical interaction between the palpating probe and the surface is modeled using the basic static friction force equation from adhesion theory, $F_{friction}=\mu F_{normal}$ [Mitchell p. 306]. The three most important physical constants for simulating a rigid textured surface are the static friction coefficients, stiffness of the surface, and the stiffness of the probe. Each of these parameters has a pronounced effect on the way a texture will feel. Figure 2.1 illustrates the physical model we use to inform our approach to analyzing the data taken during the sensing stage of the process, and also the simulation of the sensed texture.

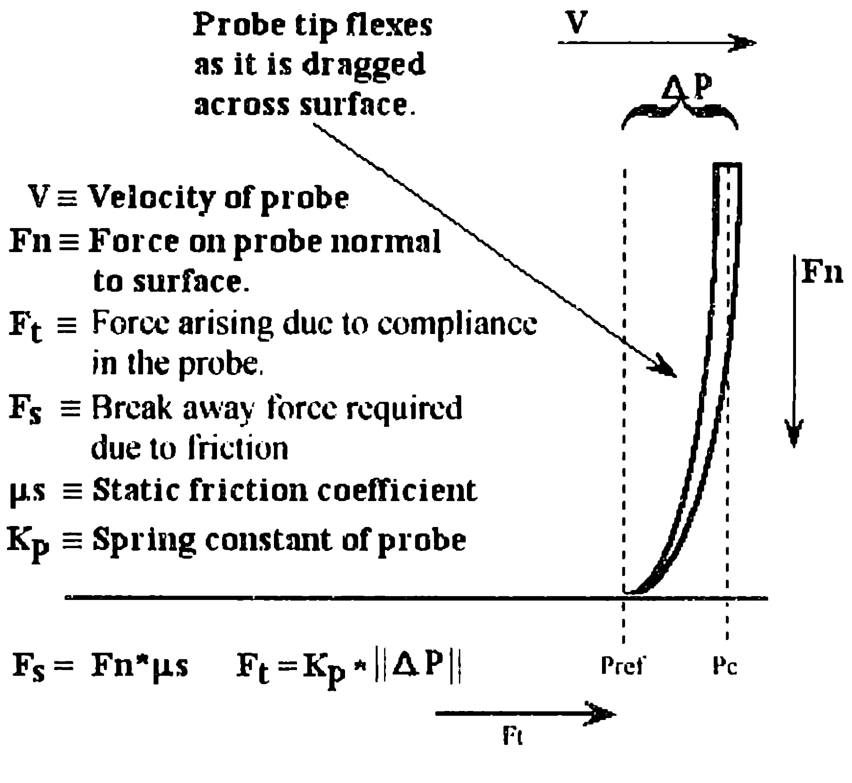

Probe tip flexes as it is dragged across surface.
V
ΔP
Fn
V ≡ Velocity of probe
Fn ≡ Force on probe normal to surface.
Ft ≡ Force arising due to compliance in the probe.
Fs ≡ Break away force required due to friction
μs ≡ Static friction coefficient
Kp ≡ Spring constant of probe
Fs = Fn*μs
Ft = Kp * ||ΔP||
Pref
Pc
Ft

Figure 2.1 Model of Probe/Surface Interaction

Figure 2.1 shows a flexing probe involved in a stick-slip interaction with the surface that it is touching and moving across. Imagine a person holding the probe at the top and stroking the surface. The force input provided by the person holding the top of the probe is resolved into vertical (Fn) and lateral (Fl) components in the graphic, where

Fn is the component of input force normal to the surface and Fl is the component in the direction of motion. Compliance in the probe and static friction effects due to the friction properties of the surface being palpated cause a displacement to occur between the top and tip of probe. The restoring force at the tip of the probe in the direction of motion, Ft, is proportional to this displacement with the constant of proportionality, Kp, which represents the stiffness of the probe. This force builds as the displacement grows larger until it is finally strong enough to overcome the static friction force resisting motion of the tip. The static friction force resisting motion is proportional to the normal force, Fn, applied by the person holding the probe with the constant of proportionality being the static friction coefficient μs. When the force at the probe tip in the direction of motion finally overcomes static friction the probe tip slips, snapping forward to relieve the tension in the probe. At that point the probe tip stops moving and the cycle begins all over again. This analysis ignores inertial effects that could cause an overshoot of the sticking position predicted in this manner. It also ignores potential dynamics from Coulomb friction effects as well, which would prevent all the tension in the probe from being relieved after a slip motion.

It has been observed that stick slip friction is commonly exhibited in point contact mechanical interactions involving minerals [Mitchell p.310]. This provides a theoretical justification for choosing this model for a probe stroking a surface such as sandpaper. It also shows that the model should extend to the surfaces of rocks as well, disregarding large geometric perturbations of the surface.

2.2 Implementation

The first step in the process of simulating real textured surfaces involves taking a sample of the surface of interest. A hybrid control scheme is used during the sampling process that directs the PHANToM to follow a given trajectory at a constant velocity, under proportional or proportional-derivative control. While following the commanded trajectory a constant normal force is exerted. The lateral forces generated by the trajectory following control algorithm and the changes in height of the end-effector are recorded. Typical sampling rates are 4.5Khz and the loop update rate typically runs at about 10Khz. Figure 2.2 shows a photograph of a PHANToM taking a sample.

Figure 2.2 Sampling a piece of sandpaper

Each element of the recorded force vector is then divided by the applied normal force and the result is interpreted as a vector of static friction coefficients based on the basic static friction force equation, $F_n*\mu = F_{fric}$. When the computed friction coefficients are plotted in histogram form they consistently fall into an approximately normal distribution. Figures 2.3 and 2.4 in section 2.3 show histograms of the computed static friction coefficients for a number of different texture samples and are accompanied by a discussion of their interpretation.

The mean and standard deviation of the computed static friction coefficients are then computed and used during the simulation of texture, along with the standard deviation of the recorded height values. The following is a description of the process involved in computing the output force to be exerted on the PHANToM during a texture simulation.

Upon starting the computation loop a perturbation for the base surface height is computed for current x,y position **Pc** (in the horizontal plane) using a Gaussian pseudo-random number generator that returns values with zero mean and the same standard deviation found for the sampled height values. The next step is to check if the endpoint of the PHANToM has penetrated the virtual surface. If it has not, no force is output. If it has, then the normal force magnitude, Fn, is computed to be proportional to that depth of penetration and the x,y point of contact **Pref** is recorded. A static friction coefficient μs is computed for this contact point by another Gaussian pseudo-random number generator that returns values with the characteristic mean and standard deviation computed from the sampled forces. The 'break-away' friction force magnitude Fs is then defined to be Fs = μs*Fn. During the next control loop iteration, if we assume contact with virtual surface is maintained, the process proceeds as follows. A new **Pc** value will have been returned by the PHANToM encoders if we have moved in the horizontal plane and a lateral force is output proportional to this distance and applied in the opposite direction of the movement. The virtual probe tip is 'stuck' at the position **Pref** and is resisting the user's motions. When the user applies enough force to move the actual position of the PHANToM, **Pc,** far enough away from the sticking point, **Pref,** that the output force resisting motion exceeds the break away force the probe tip 'slips'. The reference position, **Pref**, is then reset to be the current position, **Pc**, a new static friction coefficient is computed, and the process repeats. A pseudo-code illustration of the algorithm appears below.

```
WHILE ( simulating ) {
        Cp = get_current_position();
        surface_perturbation = get_gauss_rand_surf();  // get gaussian random surface
                                                       //  perturbation

        collision = Test_For_Object_Collision(Cp, surface_perturbation);
        IF ( collision ) {
            IF ( new_contact ) {             // check if new contact sequence
                new_contact  =  FALSE;
                new_μs_req  =  TRUE;
            }
            IF( new_μs_req ) {           // check if new friction coefficient is needed
                μs  =  get_gauss_rand_mu(); // get guassian random friction coefficient
                new_μs_req = FALSE;
            }
            $Fn = Ksurf \bullet \|surface(Cp) - Cp\|$;  // Compute normal force magnitude
            $F\mu = \mu s \bullet Fn$;                  //  calculate break away friction force
            $\Delta p = surface(Cp) - surface(Lp)$;     // stylus flex distance vector
            $Fs = Kstylus \bullet \|\Delta p\|$;        // calculate stylus flex force
            IF ( $Fs > F\mu$ ) {
                Lp = CP;               // set stick point to current point
                new_μs_req = TRUE;
            }

            $Fout = Fn \bullet N + Fs \bullet \frac{\Delta p}{\|\Delta p\|}$;  // compute output force

        }
        ELSE {              //  no contact with virtual object
            Lp  =  Cp;
            Fout = 0;
            new_contact  =  TRUE;
        }
        Output_Force(Fout);
}
```

2.3 Interpretation of Sample Data

Examples of histogram plots generated from sampling various grit values of sandpaper appear in figures 2.3. and 2.4. The vertical axis indicates the number of coefficients whose value fell within a given bin value as indicted on the horizontal axis. The plots in figure 2.3 are normalized by the number of samples and number of bins used to show how well the data falls into a normal distribution. Figure 2.4 shows the raw histograms with common axes for easier visual comparison of the mean value and standard deviations. For comparison, I have superimposed a normal curve onto the histogram plots using the same mean and standard deviation as the data in the plot but with an arbitrary magnitude chosen by hand to match the data display.

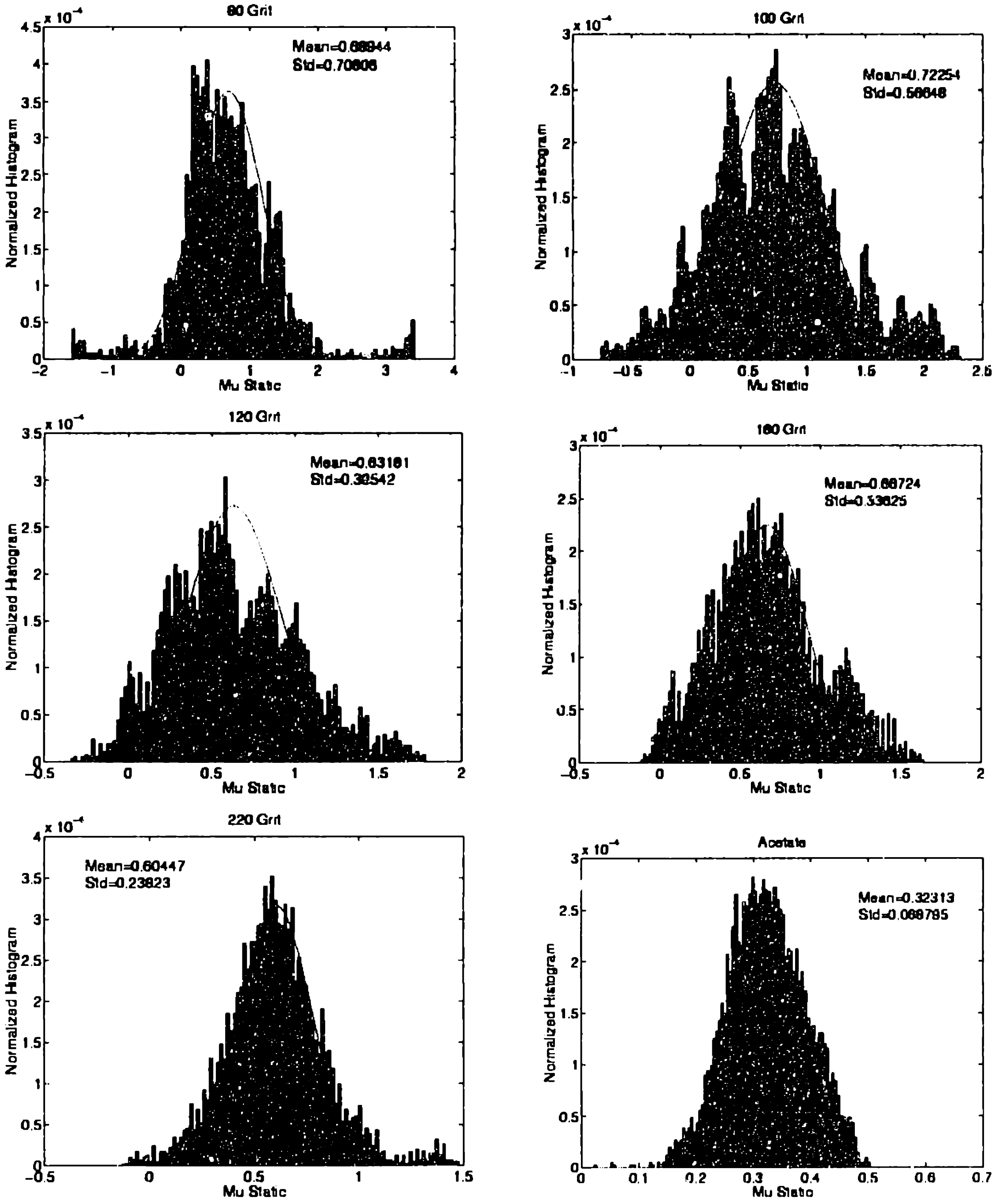


Figure 2.3 Normalized Histograms of Computed Friction Coefficients

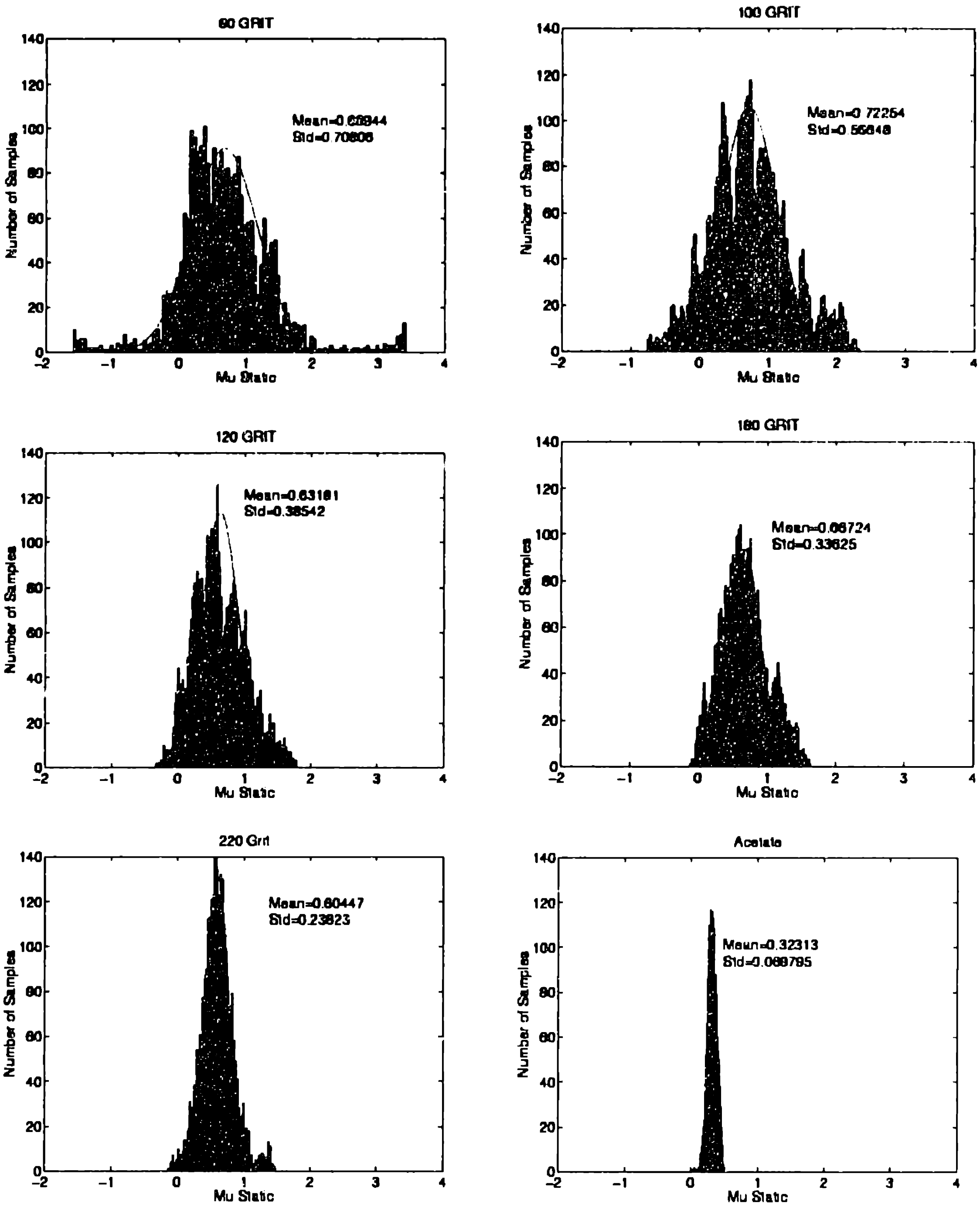


Figure 2.4 Un-Normalized Histograms of Computed Friction Coefficients

The plots illustrate some important and interesting information. First of all, they verify the ability of the PHANToM to collect useful data about the properties of the surface to be simulated. They also illustrate and confirm that the computed friction coefficients from the sensed data do fall into normal distributions. The less coarse

dependence of ordered and semi-ordered texture properties is lost and the assumption of random surface friction and height deviations is no longer valid. Spectral methods of analyzing the surfaces of objects with regular texture pattern would be a more appropriate approach.

Dr. Gunnar Jansson, Department of Psychology, Uppsala University is in the process of performing a series of experiments testing the correlation between haptic perception of virtual objects and textures using the PHANToM device and normal human haptic perception of real objects [Jansson]. Initial experiments using the software developed for this thesis (see Appendix) show good correlation between the user's perception of the coarseness of the simulated texture and that of the actual real textured surface being imitated [Billberger]. These encouraging results support the utility of this approach to sensing and simulating naturally occurring textures. In chapter three I will discuss methods for adding the textures generated in this way to three-dimensional haptic objects.

Chapter 3 Application of Textures to Three-Dimensional Virtual Objects

In order to be useful to researchers and practitioners in the field of haptic simulation the methods for texture simulation developed here must be compatible with algorithms already established for rendering haptic objects. This functional requirement was kept in mind during the development process and in this chapter I will show how these textures can be readily adapted to many of the common methods for rendering three-dimensional haptic objects. Primitive object rendering methods such as those presented in [Massie and Salisbury], polygonal mesh object modeling methods demonstrated by [Zilles and Salisbury], and ray based rendering techniques [Basdogan] are examples of general haptic rendering approaches that could utilize the texturing methods presented here.

3.1 General Description of Approach

A common element of many different techniques of haptically rendering solid objects is the computation of surface normal direction at the point of contact between the user's virtual presence point and the virtual object. This surface normal is then used to guide the direction of the force to be applied to the user while its magnitude can be varied according to material properties, state of motion, etc. The computation of the surface normal at the point of contact is the basic requirement for using the textures developed here in a hapic simulation.

The general methodology to be followed involves integrating stick slip friction and virtual probe compliance into the mechanical model of the probe/surface interactions. The difference from the material presented in chapter two being that the simulated surfaces are no longer simply planar ones but have more complicated three-dimensional geometry. Recall from chapter two that the texture simulations are based upon a model that includes a compliant stylus experiencing stick-slip friction as it moves across the surface of a virtual object. The friction coefficient for a given point on the surface determines the magnitude of tangential force required to break the probe away from that sticking point and move to a new point in the direction of motion. The positional error between the sticking point and the stylus' end point position as reported by the PHANToM encoders determine the magnitude of the force opposing motion directed tangent to the surface at the sticking point.

3.2 Texturing Algorithm

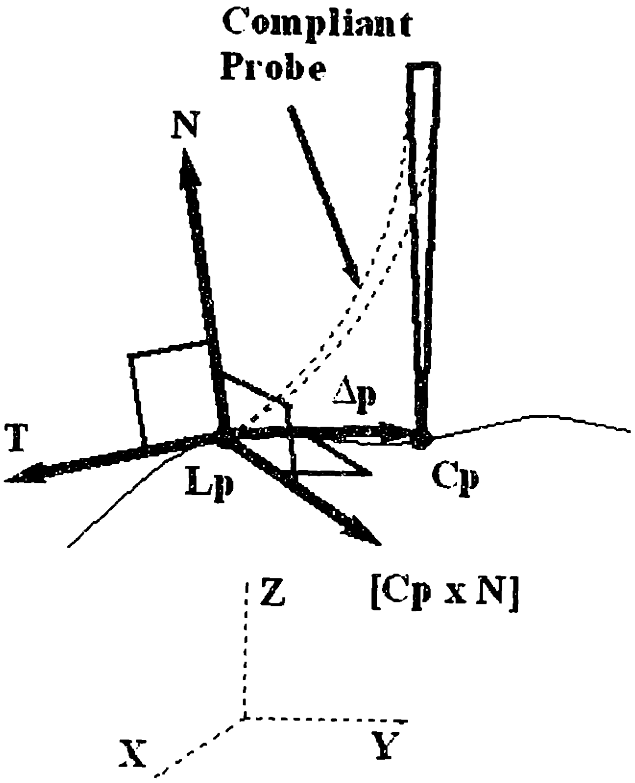


Figure 3.1 Calculating Tangent Vector for Friction Force Output

Figure 3.1 illustrates an exaggeration of the stick-slip concept on a three dimensional surface. The vector difference **Δp** between the sticking point **Lp** and the current probe surface intersection point **Cp** is used to calculate the vector **T** which is tangent to the surface at the stick point **Lp** and points in the direction opposing motion. The vector cross product of **Δp** and the surface normal is a vector normal to the plane defined by **Δp** and the surface normal. This vector **[Cp x N]** is used to calculate **T** by taking its cross product with the surface normai at **Lp** and normalizing the result. Equation 3.1 elaborates on the concept to show how the forces are calculated.

Lp ≡ Sticking Point
Cp ≡ Current point on surface
Ksurf ≡ Surface Stiffness
Kstylus ≡ Stylus stiffness
Δsurf ≡ Dcpth of virtual presence point beneath virtual surface
N(p) ≡ Surface Normal at point **p**
Fn(p) ≡ Output force normal to surface at point **p**
Ft(p) ≡ Output force opposed to motion and tangent to surface at point **p**

$$\mathbf{\Delta p} = \mathbf{Cp} - \mathbf{Lp}$$

$$\mathbf{T} = \frac{[\mathbf{\Delta p} \times \mathbf{N(Lp)}] \times \mathbf{N(Lp)}}{\left\| [\mathbf{\Delta p} \times \mathbf{N(Lp)}] \times \mathbf{N(Lp)} \right\|}$$

$$\mathrm{Fmag} = \mathrm{Ksurf} \bullet \Delta\mathrm{surf}$$

$$\mathbf{Fn} = \mathrm{Fmag} \bullet \mathbf{N(Lp)}$$

$$\mathrm{Fp} = \mathrm{Kstylus} \bullet \left\| \mathbf{\Delta p} \right\|$$

$$\mathbf{Ft} = \mathrm{Fp} \bullet \mathbf{T}$$

Equation 3.1

A pseudo-code algorithm for calculating the output force for a textured surface proceeds below. The function 'surface()' in the pseudo code refers to a function which returns the point on the virtual objects surface as appropriate to the particular collision detection and force rendering algorithms being used by the programmer. For instance, a basic method for computing this point is finding the closest point on the surface to the user presence point. In the pseudo-code below the normal force is proportional to the depth of penetration into the virtual object, a common practice for haptic algorithms. The function get_gauss_rand() refers to retrieving a pseudo random friction coefficient from a Gaussian random number generator. The Gaussian random number algorithm returns normally distributed values with a mean and standard deviation to match the desired texture. The other variables and functions are as defined above in equation 3.1 or are self-explanatory. As in the planar case the probe sticks until the lateral force from the flexing probe overcomes the static friction force opposed to motion.

```
WHILE( simulating ) {
    Cp = get_current_position();
    collision = Test_For_Object_Collision(Cp);
    IF ( collision ) {
        IF ( new_contact ) {                 // check if new contact sequence
            new_contact = FALSE;
            new_μs_req = TRUE;
        }
        IF( new_μs_req ) {                   // check if new friction coefficient is needed
            μs = get_gauss_rand();           // get guassian random friction coefficient
            new_μs_req = FALSE;
        }
        $Fn = Ksurf \bullet \|surface(Cp) - Cp\|$;   // Compute normal force magnitude
        $F\mu = \mu s \bullet Fn$;                       // calculate break away friction force
        $\Delta p = surface(Cp) - surface(Lp)$;      // stylus flex distance vector
        $Fs = Kstylus \bullet \|\Delta p\|$;            // calculate stylus flex force
        IF( $Fs > F\mu$ ) {
            Lp = CP;                          // set stick point to current point
            new_μs_req = TRUE;
        }
        $T = [\Delta p \times N] \times N$;         // calculate tangent vector at stick point
        $T = T/\|T\|$;                            // normalize
        $Fout = Fn \bullet N + Fs \bullet T$;       // compute output force
    }
    ELSE {                  // no contact with virtual object
        Lp = Cp;
        Fout = 0;
        new_contact = TRUE;
    }
    Output_Force(Fout);
}
```

Chapter 4 Soil Simulation

The approach to soil simulation I have taken is based on physically modeling the forces generated when a flat, straight, and rectangular plow blade moves through dry, sandy soil. A simple example of the kind of interaction under discussion is pushing beach sand around with a spatula or shovel. The mechanical model used to drive the simulation is a modification of one presented in original research done by McKyes and Ali [McKyes]. In the original model, the forces on a plowing surface may be calculated based upon the geometry of the plowing surface and certain soil properties. The geometry of the plow blade included in the model are the dimensions and orientation of the blade surface with respect to the direction of motion through the soil and the three soil properties density, cohesion, and angle of internal friction. Both the Viking and Pathfinder Mars exploration missions also used the McKyes and Ali model as the basis for a method of roughly determining Martian soil properties [Moore, Clow, Hutton] [Rover Team]. Unlike the texturing work in this thesis, the research presented in this chapter is simply an attempt to simulate soil without actually sensing soil properties and populating the model with parameters derived from a sensing process. However, in section 4.4 I do explore the PHANToM's capabilities to sense soil properties with encouraging results.

4.1 Dynamic Model

The basis of the model is to assume that at any instant as a plow moves through the soil a volume of soil is disturbed whose boundaries can be determined by a shear failure boundary determined from certain mechanical properties of the soil, the geometry of the plow blade, and its angle of attack. From this volume the forces acting on the blade can be computed based on the density, angle of internal friction, and cohesion of the soil, as well as any load on the surface of the soil and friction between the plow blade and soil.

The sandy soils being simulated may be considered cohesionless and shear strength arises from the friction between particles. Cohesion shear strength arises from ionic bonding between soil particles and has been found to be the dominant source of shear strength in clay-based soils but has little to no effect in sand [Liu p235-236]. The cohesion term is included in the model below for completeness' sake and as a reference for future work on simulating clay.

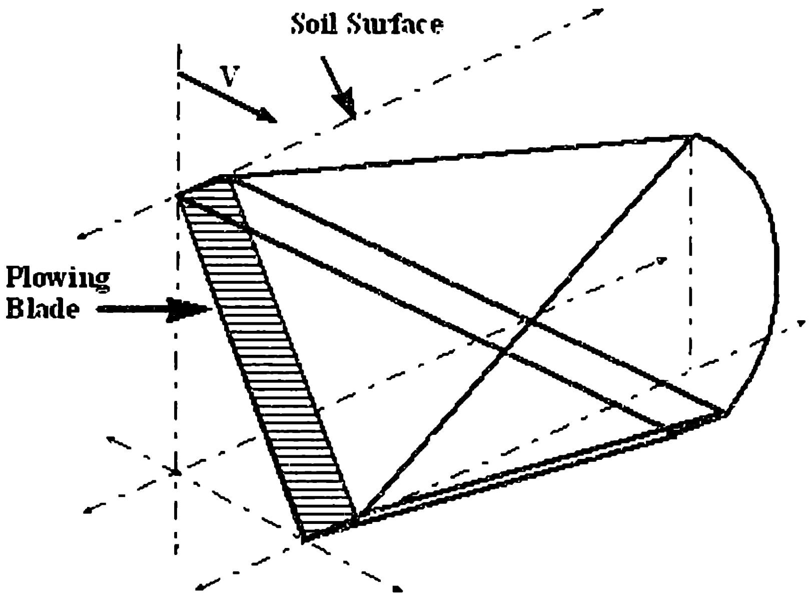


Figure 4.1 Disturbed Soil Volume

Figure 4.1 illustrates the basic idea of the McKyes and Ali construct. It consists of a triangular center volume bordered by conic sections on either side (in figure 4.1 one side omitted for clarity). From these sections we can compute the forces on the blade by computing the force factors from the separate volume sections and summing the results. Figure 4.2 shows the model for the center section of the disturbed soil volume.

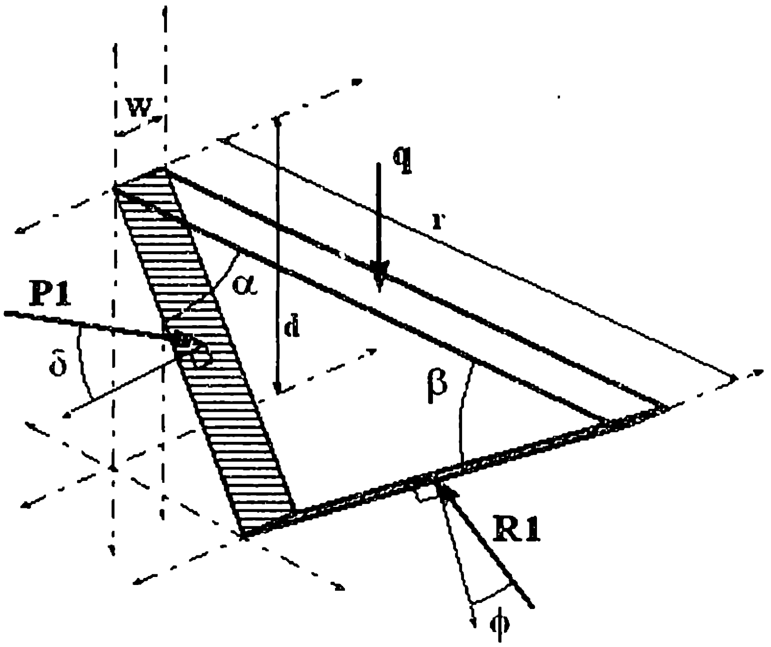


Figure 4.2 Model for Center Volume

The force generated on the plow blade from this section is derived from a force balance equation. The variables and constants involved are listed below; all variables are in SI units.

w ≡ Width of Blade (m).
d ≡ Depth of Blade Penatration Into Soil (m).
$P1$ ≡ Magnitude of Force Blade Exerts On Soil (N).
$R1$ ≡ Magnitude of Force Exerted on Soil Shear Failure Plane by Undisturbed Soil (N).
q ≡ Load on Surface (N / m^2).
r ≡ Radius of Failure Zone At Surface (m).
δ ≡ Soil / Metal Interface Friction Angle (Rad).
φ ≡ Soil Internal Friction Angle (Rad).
α ≡ Angle of Plow Blade From Horizontal (Rad).
β ≡ Angle of Soil Shear Failure Plane (Rad).
c ≡ Cohesion of Soil
γ ≡ Bulk Density of Soil.
$\theta_1 \equiv \alpha + \beta$ $\qquad$ $\theta_2 \equiv \beta + \phi$

The forces are then separated into their horizontal and vertical components in the force balance equations and then combined to solve for the force P1 that the blade is exerting on the soil volume. The horizontal force balance equation may be written

$$P1\sin(\theta_1) - R1\sin(\theta_2) = \frac{c\,d\,w\cos\beta}{\sin\beta}$$

Equation 4.1

where the right hand side of the equation represents force from the cohesion of the soil. Similarly, the vertical forces can be represented by

$$P1\cos\theta_1 + R1\cos\theta_2 = \frac{\gamma\,d\,r\,w}{2} + c\,w\,d + q\,r\,w$$

Equation 4.2

where the right hand terms, read from left to right, represent forces from the density of the soil, cohesion of the soil, and surcharge load on the surface of the soil. Now, with two equations and two unknowns, namely P1 and R1, we solve for R1 in tenns of P1 in equation 4.1 and substitute into equation 4.2. Solving for P1 we find

$$P1 = \frac{w\left[0.5\,\gamma\,d\,r + c\,d\left[1 + \cot\beta\cot\theta_2\right] + q\,r\right]}{\cos\theta_1 + \sin\theta_1\cot\theta_2}$$

Equation 4.3

which is the solution to the magnitude of the component of force the plowing blade must be exerting on the displaced soil volume to balance the center section. This force magnitude is the resolved into its horizontal and vertical components,

$$\mathbf{F}_{\mathbf{P1}} = P1\left\langle\sin\theta_1, \cos\theta_1\right\rangle$$

Equation 4.4

resulting in a two dimensional force vector.

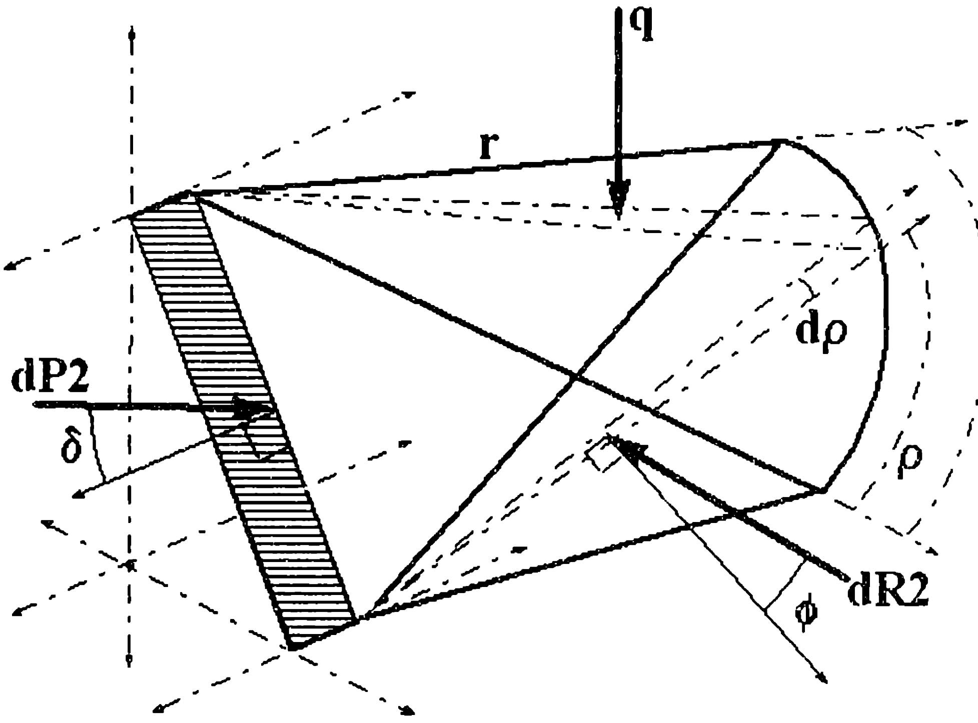


Figure 4.3 Model For Side Volume

Figure 4.3 shows the model used to compute forces on the plowing blade from the side wedge. The analysis is based on integrating forces from differential volume elements dρ over the total angle ρ' subtended by the section. Pausing then to properly define the new variables

dρ ≡ Differential volume element (m^3).
ρ ≡ Angular displacement of dρ (Rad).
ρ' ≡ Total angle subtended by section (Rad).

we approach the problem as before, computing the force balance equation for each differential element. First, the force balance in the horizontal direction

$$dP2 \sin\theta_1 - dR2 \sin\theta_2 = \frac{c\, r\, d\, d\rho \cos\beta}{2 \sin\beta}$$

Equation 4.5

where the term on the right hand side of the equation is the force arising due to the cohesion of the soil. The vertical forces sum as follows

$$dP2\cos\theta_1 + dR2\cos\theta_2 = \frac{1}{6}\gamma\ d\,r^2\,d\rho + \frac{c\,d\,r\,d\rho}{2} + \frac{1}{2}q\,r^2\,d\rho$$

Equation 4.6

where the right hand terms, from left to right, are forces due to the soil's bulk density, its cohesion, and surcharge load on the surface. Proceeding as before we solve for dR2 in terms of dP2 in equation 4.5 to eliminate the dR2 term, and substitute the result into equation 4.6 in place of dR2. Solving for dP2 we find

$$dP2 = \frac{\left[\frac{1}{6}\gamma\,d\,r^2 + \frac{1}{2}c\,r\,d\left[1 + \cot\beta\cot\theta_2\right] + \frac{1}{2}q\,r^2\right]d\rho}{\cos\theta_1 + \sin\theta_1\cot\theta_2}$$

Equation 4.7

as the solution to the magnitude of the force that the blade is exerting on the differential soil volume dρ. This resolves into the horizontal and vertical components shown in equation 4.8.

$$\begin{aligned} dH_{P2} &= dP2\sin\theta_1\cos\rho \\ dV_{P2} &= dP2\cos\theta_1 \end{aligned}$$

Equation 4.8

These force components are then integrated over the total angle subtended by the wedge ρ' to compute the total force the plow blade exerts on the side conic section of disturbed soil.

$$\begin{aligned} H2 &= \int_0^{\rho'} dH2 = \frac{\left[\frac{1}{6}\gamma\,d\,r^2 + \frac{1}{2}c\,r\,d\left[1 + \cot\beta\cot\theta_2\right] + \frac{1}{2}q\,r^2\right]\sin\theta_1\,\sin\rho'}{\cos\theta_1 + \sin\theta_1\cot\theta_2} \\ V2 &= \int_0^{\rho'} dV2 = \frac{\left[\frac{1}{6}\gamma\,d\,r^2 + \frac{1}{2}c\,r\,d\left[1 + \cot\beta\cot\theta_2\right] + \frac{1}{2}q\,r^2\right]\cos\theta_1\,\rho'}{\cos\theta_1 + \sin\theta_1\cot\theta_2} \\ \mathbf{F}_{P2} &= \langle H2, V2\rangle \end{aligned}$$

Equation 4.9

We now have a solution to the problem of computing force on the blade of a plow as it moves through soil of a given bulk density and cohesion with a given internal friction angle. The force on the plow then is the vector sum in equation 4.10.

$$\mathbf{F} = -(F_{P1} + F_{P2}) \quad .$$

Equation 4.10

4.2 Implementation

In the implementation of the soil simulator values were taken from a source text on soil mechanics [Liu p. 411] to match the properties of cohesion c, bulk density γ, and angle of internal friction ϕ for various sandy soils. As discussed in the introduction to this chapter the soil types that the PHANToM and dynamic model are suited to mimicking are limited to loosely packed, medium-to-fine grained sands, silts, and dry clays. High cohesion factors require forces too large for the PHANToM to render and large grain sizes, relative to the plow-bladed size, would break down the validity of the model.

In a departure from the McKyes model I choose a failure plane angle β based upon the Rankine theory for passive lateral earth pressures [Liu p.401]. McKyes and Ali's plowing theory model selects a failure plane angle β that minimizes the horizontal force term arising from the soil density [McKyes p.48]. This minimization process is too costly in terms of processing time for a control loop and so this simplification was made. The Rankine model provides a method for computing forces on earth bearing walls in a state of motion relative to the soil.

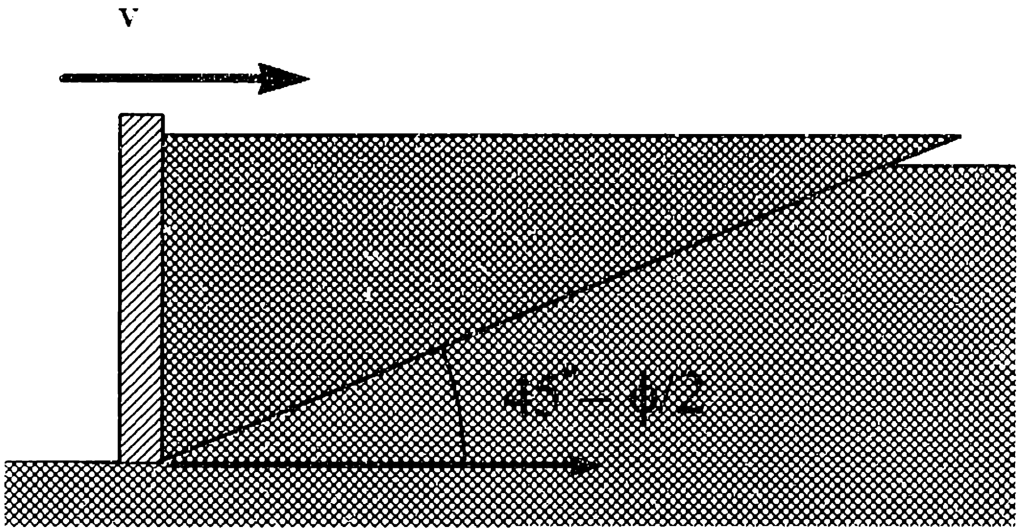


Figure 4.4 Rankine Model Failure Plane Angle

Figure 4.4 shows the predicted failure plane angle as an earth retaining wall moves toward the soil mass.

In order to give the simulation a more realistic feel the friction angle ϕ and the failure plane angle β are perturbed with the Gaussian random distribution algorithm using the base values discussed above as the mean and standard deviations found through trial and error to give the most natural feeling results.

4.3 Results

The soil simulation results are encouraging. A reasonably convincing simulation of probe/soil interaction is created using the methods described. The soil parameters of density, and internal friction angle may be varied to achieve palpably different feeling soils. Cohesion needs to be very large (hundereds to thousands of Kilo-Pascals) to achieve perceptible changes in the soil behavior and was left at or near zero for most simulations. This validity of this choice is supported in the soil mechanics literature [Liu pp. 235-242] which contains statements to the effect that dry sandy soils are virtually cohesionless. Cohesion becomes a dominant factor when examining the shear strength of clay based soils.

The plow blade dimensions were also changed to observe the effect on the simulated soil interaction and the results were as would be intuitively expected. Specifying a wider blade in the model causes more resistance to movement while a narrow blade achieves the opposite effect.

Experiments were conducted to observe the effect of velocity on the impedance experienced by a real plow blade moving through real sand. The original intent of these experiments was to verify the validity of the model's velocity independence but other interesting information was discovered as well. This information, discussed in detail below, will be studied further in future work on modifying the soil simulation model.

Figure 4.5 shows a photograph of the sampling apparatus. A PHANToM is outfitted with a small plow blade (see figure 1.2 for close up of plow blade) and commanded to follow a special trajectory at a certain depth in the sand.

Figure 4.5 Sand Impedance Testing Setup

The trajectory the PHANToM is commanded to track is a simple semi-circular arc in a horizontal plane with a radius equal to the length of the upper link of the device and a center of rotation corresponding to the PHANToM's center of rotation about its Z axis (see figure 1.5 for PHANToM coordinate space definition). The plow blade is oriented such that the plane defined by its surface remains perpendicular to the instantaneous direction of motion during the course of the trajectory. The angle of attack of the plow blade (the angle α in the model) is a constant 90°. It is assumed that any centripetal force effects on the output will be negligible at the rotational velocities commanded during the experiments. The experimental results are shown in the graphs of figure 4.6.

Figure 4.6 depicts the magnitude of the force commanded by the PHANToM to track the trajectory described by the horizontal axis at the angular velocity specified above each individual graph. Six experiments were conducted using angular velocities 0.25, 0.5, 0.75, 1.0, 1.25, and 1.5 radians per second. The experimental procedure consisted of leveling out the sand and positioning the plow blade at the starting point at a specific depth in the sand. Care was taken to ensure that the plow blade began its path through the soil sample at the same depth for each experiment with an estimated accuracy of 2-3 mm. The commanded vertical position remains constant throughout the trajectory and at the beginning of each experiment roughly three-quarters the plow blade is immersed in the sand. Finally, the PHANToM is commanded to start its trajectory from the starting point at the angular position -π/4 and moves at the specified velocity to the end point at the angular position π/4.

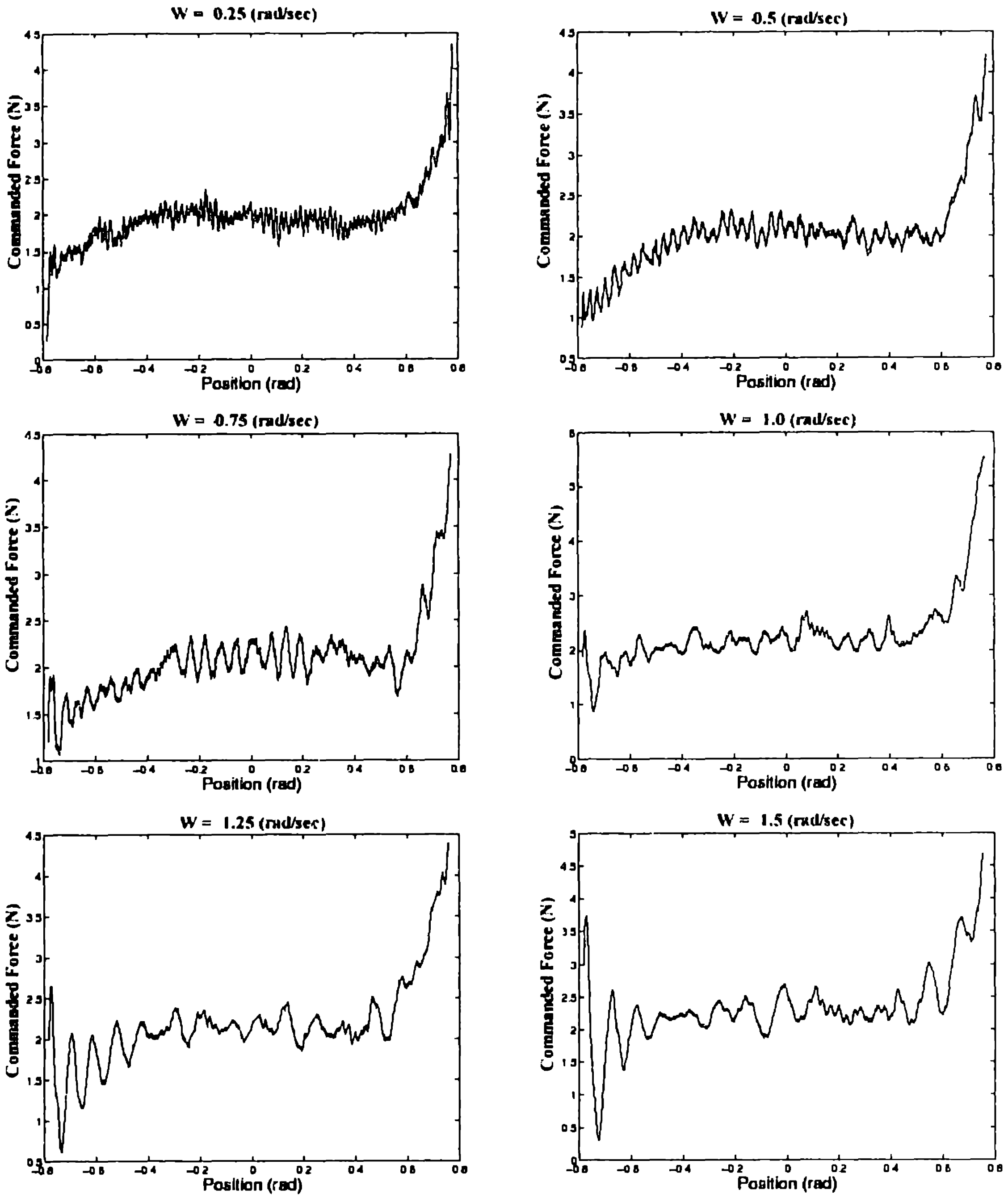


Figure 4.6 Force Required for Trajectory Tracking Through Sand at Different Velocities

Each graph shows an initial building up of the average force, then a leveling off, followed by a final build up as the trajectory is completed. The graphs also show consistent oscillatory behavior, though at different spatial frequencies for different command velocities. In the time domain the oscillations occur at the same frequency. That frequency is roughly 400 radians per second (60 Hz) which corresponds to the first peak seen in the frequency response of the Y axis in figure 1.11. This makes perfect sense because the command trajectory is exactly that trajectory that would experience the dynamics observed in the Y axis frequency response for the entire path due to the configuration of the PHANToM.

Observation of the sampling process indicates that the initial build up of force corresponds to an accumulation of sand piling up against the blade. When the average force levels off the pile has reached it's largest possible mass and as new sand is piled up in front of the blade other accumulated sand slips off the sides and back. Finally, the build up of force at the end of the trajectory is assumed to be caused by the approach of the blade to the side of the container at which point the sand in front on the blade begins to be compressed rather than pushed.

The graphs clearly indicated that during the 'steady-state' of the motion through the soil (from about -0.4 to 0.4 rad) the average impedance force experience by the plow blade is velocity independent. This confirms an important and somewhat counter-intuitive assumption of the model.

A final comment about the oscillations in the path is that the initial oscillations are quite large for the higher command velocities. This is due to the nature of the proportional controller used for trajectory tracking. The higher velocity runs require large initial force input and thus experience greater overshoot. This effect is what is observed in the graphs.

Chapter 5 Conclusions and Future Work

5.1 Summary

This work has presented mathematical models and techniques for creating natural feeling haptic simulations of real textured surfaces and soils. For haptic texturing a stick-slip friction model is utilized along with normally distributed friction coefficients. Surface textures are characterized by the mean and standard deviation of static friction coefficients directly sensed through palpation with the PHANToM device. The texturing method renders a statistically accurate haptic simulation of a specific rigid surface texture. The utility of the method extends to textured surfaces created through random cumulative processes, all of which will have the property of Gaussian random height distributions. The textures on the surfaces of many rocks have this property by virtue of the geological processes through which they are commonly formed.

The texture rendering algorithm developed is shown to be extensible to many of the different haptic rendering techniques commonly used by researchers and practitioners in the field of haptics. Specifically, the method can be easily added to ray based, polygonal mesh, and geometric primitive based methods of haptic simulation

A method for rendering haptic simulations of sandy soils is also presented. Sandy soil simulations are created by modeling a volume of soil experiencing shear failure due to force applied by a plowing surface. The model is augmented by statistically varying certain physical and geometric parameters. While the work presented in this thesis regarding soil simulation is not based upon directly sensing the relevant mechanical properties of the soil it is demonstrated that the PHANToM device is capable of doing so. The validity and utility of the soil model is discussed and tested and encouraging results recommend further study to be undertaken to expand on the soil simulation work begun here.

Two primary themes appear in this thesis providing a thread of continuity to this thesis and the approach I have taken to exploring the field of haptic simulation. The first is the extensive use of the rendering device itself as a data acquisition platform. I believe my efforts have shown that the PHANToM is a capable platform for collecting useful information about the environment around it. That information can in turn be used to drive a simulation rendered the same platform.

The second theme appearing in this thesis is liberal use of stochastic analysis and processes to analyze data collected with the PHANToM , drive simulations rendered by the device, and to perform system identification on the display itself. The use of statistical information has resulted in compact representations of complicated mechanical and geometric properties while generating very realistic virtual touch simulations.

5.2 Suggestions for Future Work

An obvious extension of these texturing techniques is to couple them with methods for acquiring a model of the overall geometry of objects. Stereo vision, radar, laser range finders, or mechanical palpation are some of the possible ways to acquire detailed models of the gross geometry of an object to be haptically simulated. Overlaying the correct texture onto an overall geometry of a virtual object will result in truly compelling simulations.

The soil model needs to be studied further in order to account for unmodeled dynamics occurring as the sand piles up and slides away as the plowing surface moves through it. A cylindrical plow would be closer to the human experience of touching sand as far as recreating the impedance felt when touching and moving through the soil with a finger. It would also simplify some aspects of the model and may remove the oscillatory build up and loss of sand mass being pushed when moving through the media. A cylindrical plow has the further virtue that the impedance force exerted on the user can be more independent of orientation of the cylinder with respect to the direction of motion since it is a solid of revolution.

Further work will be conducted to study the capability of the PHANToM to acquire soil property values and to simulate wet sand and other types of soils. Also, the system identification work of the PHANToM system will be continued and extended to verify and elaborate on the work begun here.

Appendix Palpate Program Users Manual

A.1 Introduction

The Palpate program is a user oriented implementation of the texture sensing and simulation algorithms described in chapter two of this thesis. The program was first written in the winter of 1997-1998 in order to allow other researchers of haptic interfaces to explore haptic texturing and also the psycho-physical properties of virtual texture representations. The user interface of the program was implemented in Sun's JAVA language version 1.1.3 while the PHANToM control software was written in Microsoft's Visual C++ 5.0.

The program requires a PC type computer running Microsoft's Windows 95 or Windows NT 4.0 or compatible operating systems. Computational power requirements are not especially demanding for the program (see chapter one of this thesis for a discussion of system demands) and the program should run acceptably on any PENTIUM based PC with 100Mhz or better processor speed.

A.2 Obtaining and Installing the Program

The Palpate program is available for ftp down-load. If interested in obtaining a copy you may contact the author at dfg@ai.mit.edu to arrange for a copy to made available.

To install the program unpack the zip file to the directory *c:\palpate*. It is important for proper program operation that this path be used. After unpacking the compressed files the following directory structure should exist on your local drive:

c:\palpate

c:\palpate\lib

c:\palpate\configs

c:\palpate\samples

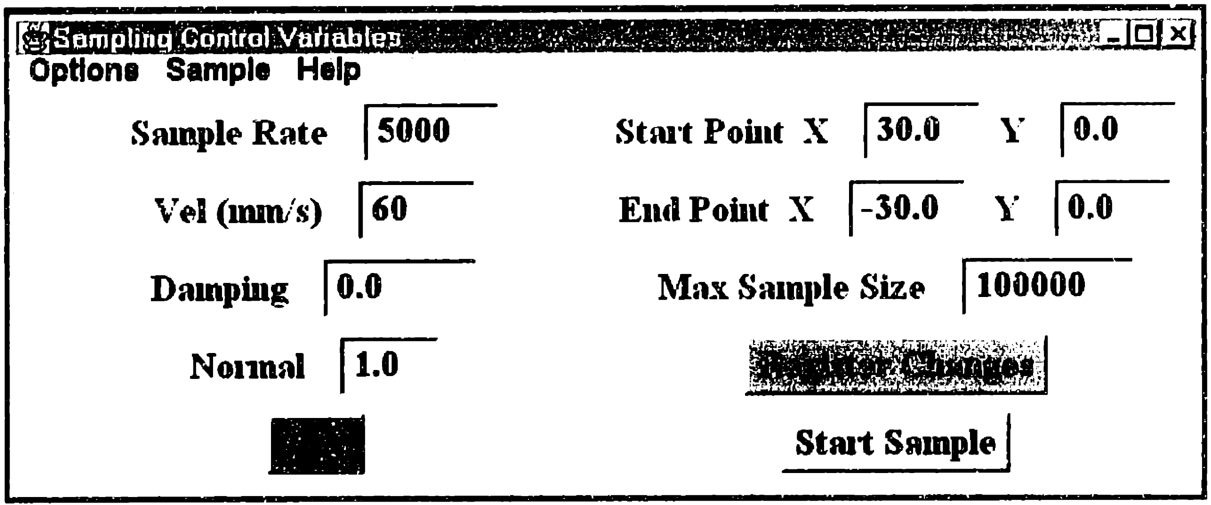
c:\palpate\bin

There should also be a shortcut called 'palpate' in the *c:\palpate* directory which when double clicked will run the program.

The program works with the PHANToM haptic interface device and contains a PHANToM initialization file in the *configs* directory called *phantom_init.ini* This initialization file assumes an A model PHANToM at address 0x300 using high resolution encoders. If the PHANToM you are using has different attributes you may delete the *phantom_init.ini* file and substitute your PHANToM initialization file but be sure to rename it *phantom_init.ini* and put it in the *c:\palpate\configs* directory.

A.3 Sampling Control Window

After taking these steps the software is ready to run. After double clicking the Palpate shortcut to start up the program a user interface window titled 'Sampling Control Variables' will appear as shown in figure A.1.



Sample Rate 5000

Vel (mm/s) 60

Damping 0.0

Normal 1.0

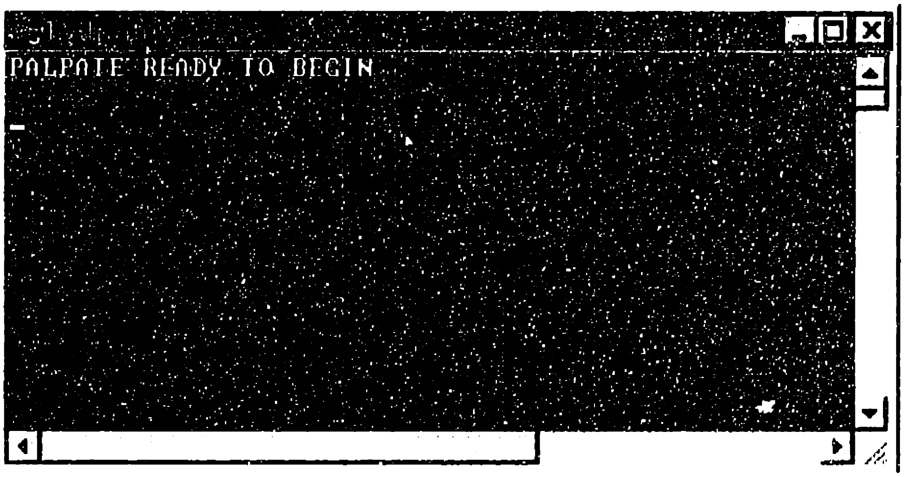
Start Point X 30.0 Y 0.0

End Point X -30.0 Y 0.0

Max Sample Size 100000

Figure A.1 Sample Control Variables Window

This window has user configurable variables for sampling textured surfaces. The values will be preset to some default values, which should work well for most purposes. Another console window or 'DOS window' will also appear providing user prompts and program state information, it is referred to as the status window. Figure A.2 shows how the window should appear at start-up.



```
PALPATE READY TO BEGIN
```

Figure A.2 Palpate Status Window

Use the console window to check for program errors and to check program progress.

A.3.1 Sampling Control Fields

The user configurable variables in the sample control window are sample rate, velocity, damping, normal, start point, end point, and maximum sample size.

Sample Rate is used to guide ballpark data acquisition rate when sampling a surface texture. The actual rate will be output to the console window after a sampling process. Since the program does not use the Sensable Devices Co. software scheduler exact timing cannot be provided but the advantage of higher update and sample rates is gained. We have found rates of 5000Hz to be more than adequate for sandpaper type textures. The units of this field are Hz.

The **velocity** field sets the stroke velocity or how fast the PHANToM arm moves during the surface stroking process. Typical safe values range from 10 to 200 mm/sec. We have found 60 to 120 mm/sec to work well. The units of this field are mm/sec.

The **damping** field allows the user to add more or less damping to the trajectory following algorithm of the surface stroking process. Typical stable values range from .001 to .009. This field is tricky in that system behavior will be very dependent on encoder resolution and the update rate of the phantom controller. It is suggested to use the default 0 value until the user becomes familiar with program operation. The units of this field are Kg/sec/1000 or grams/sec.

The **normal** field controls the constant force with which the surface stroking process presses down on the surface. Consult the PHANToM users guide to familiarize yourself with the limits of the PHANToM's force capabilities. The default value of one newton works adequately for most sandpaper type surfaces. The units of this field are in newtons (Kg.m/sec^2).

The **start point** fields give a starting point for the surface stroking process in the PHANToM's coordinate space. There are two text fields to enter x and y coordinate values to define the starting point of the trajectory to be followed during the sampling process. The units are in mm.

Similarly, the **end point** fields specify an end point for the surface stroking process. Again, the units are mm. The start and end points define the line along which the PHANToM will move at the velocity specified, pressing with the force specified, and sampling at roughly the rate specified.

The **max sample size** field sets the array size used when sampling surface data. Long stroke paths and slow stroke velocities require larger sample sizes. Use the default value unless a memory error occurs during execution. Memory errors will be reported in the console window.

The **register changes** button updates the program variables after making changes to the fields. Changes made in the text entry boxes are not implemented until the register changes button has been pressed.

The **start** button begins program execution. (HOLD PHANToM IN THE HOME POSITION WHEN PRESSING THIS BUTTON). Continue to hold the PHANToM after pressing the start button. Hold the PHANToM lightly enough to allow it to move to it's start position but firmly enough to keep the PHANToM from banging the probe tip down on the surface as it moves to the start point. After the PHANToM stops at the start position you should let go and allow the surface stroking process to progress unimpeded.

The **stop** button will kill execution of a surface stroking process and discard any data accumulated on the surface.

A.3.2 Sampling Menu Choices

The **options menu** under the sampling control variables window has 7 values; save config, save config as, default config, load config, simulation vars, system vars, and exit.

save config saves the current configuration to the current active configuration file. Unless the user has changed it using 'load config' or 'save config as', the current active configuration file is always *default.cfg*. Be careful not to modify *default.cfg* unless you are sure you want the current settings to be the default values. If you have accidentally modified the default values and would like to retrieve them they are:

Default Sampling Variables

sample rate = 5000

vel = 60

damping = 0

normal = 1

start point (x,y) = 30 , 0

end point (x,y) = -30, 0

max sample size 100000

Default Simulation Variables

Probe stiffness = 2.5

base height = -20

surface stiffness = 2.0

Default System Variable

port# = 4444

Save config as opens a dialog window to prompt the user to save the current configuration using a name entered by user. Any name and any suffix is acceptable but we recommend using 'cfg' as a suffix to make them easier to recognize.

Default config loads the *default.cfg* file and automatically registers the values with the PHANTOM controller software.

Load config opens a dialog window to allow the user to choose to load a custom configuration that they may have defined and saved.

Simulation panel opens a simulation control variables window used during surface texture simulation. See Section A.4 'Simulation Control Window' for more info.

System panel opens a system control variables window used to change certain computer system variables. See Section A.5 'System Control Window' for more info.

Exit terminates the palpate program.

The Sample Menu offers the user two choices: save sample and save sample as.

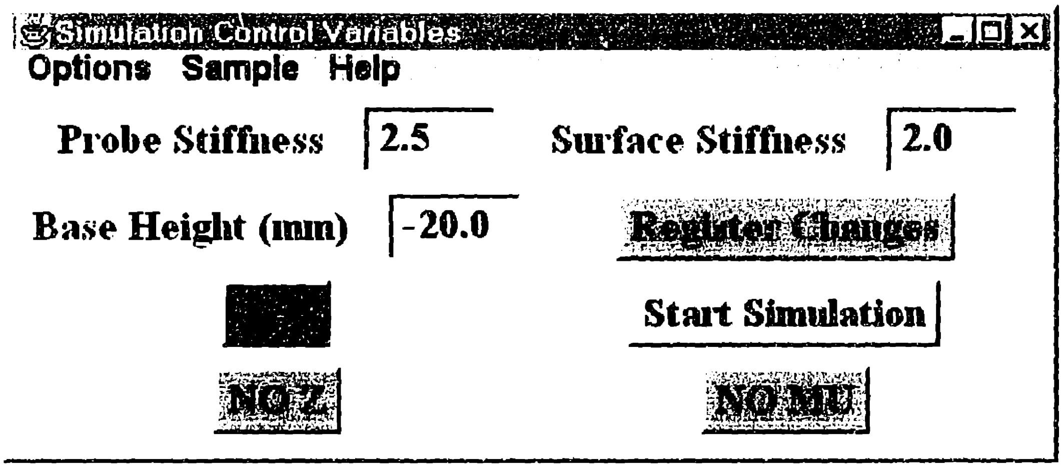
Save sample saves the current computed surface sample variables (mean and standard deviation of the static friction coefficients and surface height) to the current active sample file. If there is no current active sample file a dialog window appears prompting the user to name a file. There is no default sample file.

Save sample as opens a dialog window prompting the user to name a file to save the current sample variables in.

The **help** menu offers the user two choices: about and help. **About** displays a small window with version and developer info and **help** displays a window showing the text of a *readme.txt* file, which explains the basics of operating the *palplate* program.

A.4 Simulation Control Window

This window is used as a control panel for running texture simulations. From this window past samples may be loaded and simulated and certain simulation factors may be modified to examine their effects. The Simulation Control Vars window has 3 user modifiable fields, probe stiffness, base height, and surface stiffness.



Probe Stiffness 2.5
Surface Stiffness 2.0
Base Height (mm) -20.0

Figure A.3 Simulation Control Window

A.4.1 Simulation Control Variables

Probe stiffness changes the assumed stiffness of the probe arm. Since the underlying probe-surface interaction model assumes a stick-slip interaction probe stiffness defines the relationship between 'virtual arm' flex and lateral force exerted. See chapter two of this thesis for a more complete explanation of the underlying mechanical model informing the surface simulation.

Base height defines the height in PHANToM space that the simulated texture will be overlaid upon. It is important to use a **negative** value for the base height. Because the user holds the PHANToM in the home position, which will be defined to be the origin of PHANToM space (0,0,0) at startup, If a positive value is set here the PHANToM will be below the surface at startup and large forces will be generated. This can damage the PHANToM and should be avoided. The units of this field are in mm.

The **surface stiffness** field allows the user to change how much normal force is generated proportional to how far below the virtual surface the PHANToM end effector is.

The **register changes** button performs the same function as in the sampling control variables window.

The **NO MU** button toggles the friction element of the simulation on and off. Similarly, the **NO Z** button toggles the height deviation element of the simulation. These buttons allow the user to explore the relative importance of each component of the overall texture simulation to the users perception of coarseness.

The **start** button begins surface texture simulation. HOLD THE PHANTOM IN THE HOME POSITION WHEN PRESSING THIS BUTTON. Also, make sure that there is an active set of simulation variables. You can do this by loading an existing simulation file or running a simulation of the surface just previously sampled. There are no defined simulation variables at start up before a sample has been taken or a sample file has been loaded. After sampling a surface the sample variables computed will be used for any subsequent simulations until another sample is taken or a sample file is loaded.

The **stop** button terminates simulation execution.

A.4.2 Simulation Menu Choices

The **options** menu contains the same functions as that of the sampling control variables Options menu. The one difference being the substitution of the choice

'Sampling Panel' in the place of 'Simulation Panel'. The **sampling panel** opens a sampling control variables window.

The **sample** menu has one choice, **load sample**, which opens a dialog window allowing the user to select a saved sample file and use it to drive a texture simulation.

The **help** menu has the same functions as in the Sampling Control Vars window.

A.5 System Control Window

The system control variables window currently contains one field which chooses a port value used in socket based interprocess communications between the user interface processes and the PHANToM control device driver processes. It should not be necessary to change the default value of 4444 but if the program reports that socket errors have occurred then it is worth it to try to change the port to some other arbitrary number between 1000 and 5000.

A.6 Using the Palpate Program

The program comes with 5 sample textures. You may load these and run them from the Sample Control Vars window. Start the Palpate program by double clicking the Palpate shortcut. The Sample Control Vars window will appear. Select the Simulation Panel choice in the Options menu to open the Simulation Control Vars window. Choose load sample from the Sample menu and choose one of the included samples. Try '50grit' for starters. This will drive a simulation of 50 grit sandpaper.

After selecting a sample file hold the PHANToM in the home position and press the start button. A simulation of 50 grit sandpaper is now running. Find the surface plane of the simulated texture, it will be down at the base height in PHANToM space (i.e. if base height is specified to be -20 the surface will be 2 cm down, and will be a plane oriented perpendicularly to the z axis). Explore the surface.

Remember that the simulation is based on a compliance model of the surface and the probe so that excessive normal forces applied by the user will degrade the simulation besides overworking the PHANToM motors. A good rule of thumb is to apply normal forces similar to those used while sampling, in the neighborhood of how hard one presses while writing with a pencil.

After getting the general idea try the other sample files.

A.6.1 Sampling and Simulating a Texture

Before using the *palpate* program to sample a texture you must replace the thimble end effector of the PHANtoM with a steel or aluminum shaft with a diameter of 0.27 cm and a length of roughly 4cm. Sharpen the tip or at least round it off. Place a planar surface with a randomly distributed texture in front of the PHANTOM around 2 or 3 cm down from the tip when the PHANToM is in the home position. Be sure that the textured surface to be sampled is secure and will not slide. Check that the start and end points specified in the Sampling Control Vars window are entirely within the area of the textured surface. Check that the other variables are set up satisfactorily (use the default values when starting out). Hold the PHANToM in the home position and press the start button. DO NOT LET GO. The PHANToM will ramp up its forces fairly quickly in the direction of the start point specified. Gently hold the PHANToM while allowing it to move to the start position. Once it has stopped at the start position you may release it. After pausing for a moment the PHANToM will stroke the surface and then stop exerting forces at the trajectories end point. Refer to the console window for user prompts or error messages. You may save the sampled texture as described above or simply change to the simulation control variables window to run a simulation of the texture. It is not necessary to save the texture before running a simulation but remember that it will be lost if another sample is taken before saving the existing one.

Please Contact Donald Green dfg@ai.mit.edu with any questions.

Bibliography

1.) Billbeger,K. "En Jamforelse Mellan Haptisk Perception av Reella och Virtuella Texturer." ("A Comparision of Haptic Perception of Real and Virtual Textures.") Undergaduate Thesis. Department of Psychology, Upssala University, Upssala, Sweden. 1998.

2.) Basdogan, C., Ho, C., Srinivasan, M.A., " A Ray Based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments", Proceedings of the Winter Annual Meeting of ASME, Nov 15-21, 1997.

3.) Chen., C.H., "A study of Texture Classification Using Spectral Features." Proc. Sixth International Conference on Pattern Recognition. p.1074-1077. 1982

4.) Chen, J.; Taylor, R M. "Nanomanipulator Force Feedback Research." Proceedings of the First PHANToM Users Group Workshop, December 1996. MIT A.I. Technical Report No 1596.

5.) Fritz, J. P., Barner, K. E. "Stochastic Models for Haptic Texture" Proc. SPIE Int. Symp. on Intelligent Systems and Advanced Manufacturing - Telemanipulator and Telepresence Technologies III Conference, Boston, MA, Nov. 1996.

6.) Gramanopoulos, N. "Terrain Type Recognition Using ETRS-1 MSS Images" Rec. Symp. Significant Results Obtained from the Earth Res. Technol. Sat., NASA SP-327, pp. 1229-1241, March 1973.

7.) Haralick, R. M. et.al. "Textural Features for Image Classification" IEEE Transactions on Systems, Man, and Cybernetics, V. SMC-3, No. 6, 1973.

8.) Haralick, R. M. "Statistical and Structural approaches to Texture" Proceedings of the IEEE, V. 67, No. 5, May 1979.

9.) Jansson, G. "Haptic Perception of 3D Real and Virtual Objects." Proceedings of the Conference on Representation and Blindness, San Marino, May 22-23, 1998.

10.) Kass, M. and Witkin, A."Analyzing Oriented Patterns" Computer vision, Graphics, and Image Processing, Vol. 37, pp362-385, 1987.

11.) Lambe, T.W; Whitman, R.V. "Soil Mechanics" New York, John Wiley & Sons, Inc. 1969.

12.) Liu, C. and Evett, J. B. Soils and Foundations, 4th ed., Prentice-Hall, Inc., Upper Saddle River, N.J., 1998.

13.) Massie, T. H. "Design of a Three Degree of Freedom Force-Reflecting Haptic Interface." SB Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. May 1993.

14.) Massie, T. H. "Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction." SM Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology. February 1996.

15.) Massie, T. H.; Salisbury, J.K. "The PHANToM Haptic Interface: A Device for Probing Virtual Objects." ASME International Mechanical Engineering Exposition and Congress, Chicago, IL. November 1994.

16.) McKyes, E., Ali, O.S. "The Cutting of Soil by Narrow Blades" Journal of Terramechanics. V. 14, No. 2, pps. 43-58, 1977.

17.) Minsky, M. D. R. "Computational Haptics: The Sandpaper System of Synthesizing Texture for a Force Feedback Display." Ph.D. thesis, Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, 1995.

18.) Mitchell, J.K. "Fundamentals of Soil Behavior" New York, John Wiley & Sons, Inc, 1976.

19.) Moore, H.J., Clow, G.D., Hutton, R.E. "A Summary of Viking Sample-Trench Analyses for Angles of Internal Friction and Cohesions." Journal of Geophysical Research, Vol. 87, No. B12, pp. 10043-10050. Nov. 30, 1982.

20.) Pentland, A. "Fractal Based Description of Natural Scenes," IEEE Trans. Pattern Analysis and Machine Intelligence, V. 6, pp. 661-647. November 1984.

21.) Rao, A. R., A Taxonomy for Texture Description and Identification., Springer-Verlag., New York., 1990

22.) The Rover Team. "The Pathfinder Microrover" Journal of Geophysical Research, v. 102, No E2, pp. 3989-4001. Feb. 1997.

23.) Sirra, J.; Pai, D.K. "Haptic Texturing - A Stochastic Approach." Proceedings of the 1996 International Conference on Robotics and Automation. Minneapolis, April 1996.

24.) Strang, G. "Introduction to Applied Mathematics" Wellesley-Cambridge Press. pp. 263-330. 1986.

25.) Zilles, C.B., Salisbury, J.K. "A Constraint based God-Object Method for Haptic Display." Proceedings of the International Conference on Intelligent Robots and Systems, Augus 5-9, 1995.